

Ignacio Rojas · Olga Valenzuela ·  
Fernando Rojas · Luis Javier Herrera ·  
Francisco Ortuño (Eds.)

LNBI 13347

# Bioinformatics and Biomedical Engineering

9th International Work-Conference, IWBBIO 2022  
Maspalomas, Gran Canaria, Spain, June 27–30, 2022  
Proceedings, Part II

2 Part II

 Springer

MOREMEDIA



## Subseries of Lecture Notes in Computer Science

### Series Editors

Sorin Istrail

*Brown University, Providence, RI, USA*

Pavel Pevzner

*University of California, San Diego, CA, USA*

Michael Waterman

*University of Southern California, Los Angeles, CA, USA*

### Editorial Board Members

Søren Brunak

*Technical University of Denmark, Kongens Lyngby, Denmark*

Mikhail S. Gelfand

*IITP, Research and Training Center on Bioinformatics, Moscow, Russia*

Thomas Lengauer

*Max Planck Institute for Informatics, Saarbrücken, Germany*

Satoru Miyano

*University of Tokyo, Tokyo, Japan*

Eugene Myers

*Max Planck Institute of Molecular Cell Biology and Genetics, Dresden, Germany*

Marie-France Sagot

*Université Lyon 1, Villeurbanne, France*

David Sankoff

*University of Ottawa, Ottawa, Canada*

Ron Shamir

*Tel Aviv University, Ramat Aviv, Tel Aviv, Israel*

Terry Speed

*Walter and Eliza Hall Institute of Medical Research, Melbourne, VIC, Australia*

Martin Vingron

*Max Planck Institute for Molecular Genetics, Berlin, Germany*

W. Eric Wong

*University of Texas at Dallas, Richardson, TX, USA*



More information about this subseries at <https://link.springer.com/bookseries/5381>


Ignacio Rojas · Olga Valenzuela ·  
Fernando Rojas · Luis Javier Herrera ·  
Francisco Ortuño (Eds.)

# Bioinformatics and Biomedical Engineering


9th International Work-Conference, IWBBIO 2022  
Maspalomas, Gran Canaria, Spain, June 27–30, 2022  
Proceedings, Part II


*Editors*

Ignacio Rojas   
University of Granada  
Granada, Spain

Fernando Rojas   
ETSIIIT. CITIC-UGR  
University of Granada  
Granada, Spain

Francisco Ortuño  
University of Granada  
Granada, Spain

Olga Valenzuela   
Faculty of Sciences  
University of Granada  
Granada, Spain

Luis Javier Herrera   
ETSIIIT  
University of Granada  
Granada, Spain

ISSN 0302-9743

ISSN 1611-3349 (electronic)

Lecture Notes in Bioinformatics

ISBN 978-3-031-07801-9

ISBN 978-3-031-07802-6 (eBook)

<https://doi.org/10.1007/978-3-031-07802-6>

LNCS Sublibrary: SL8 – Bioinformatics

© Springer Nature Switzerland AG 2022

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG  
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

# Preface

We are proud to present the final set of accepted full papers for the 9th International Work-Conference on Bioinformatics and Biomedical Engineering (IWBBIO 2022) held in Gran Canaria, Spain, during June 27–30, 2022.

IWBBIO seeks to provide a discussion forum for scientists, engineers, educators, and students about the latest ideas and realizations in the foundations, theory, models, and applications for interdisciplinary and multidisciplinary research encompassing disciplines of computer science, mathematics, statistics, biology, bioinformatics, and biomedicine.

The aim of IWBBIO 2022 was to create a friendly environment that could lead to the establishment or strengthening of scientific collaborations and exchanges among attendees, and therefore IWBBIO 2022 solicited high-quality original research papers (including significant work in progress) on any aspect of bioinformatics, biomedicine, and biomedical engineering.

Submissions relating to new computational techniques and methods in machine learning; data mining; text analysis; pattern recognition; data integration; genomics and evolution; next generation sequencing data; protein and RNA structure; protein function and proteomics; medical informatics and translational bioinformatics; computational systems biology; modeling and simulation; and their application in the life science domain, biomedicine, and biomedical engineering were especially encouraged. The list of topics in the call for papers has also evolved, resulting in the following list for the present edition:

1. **Computational proteomics.** Analysis of protein-protein interactions. Protein structure modeling. Analysis of protein functionality. Quantitative proteomics and PTMs. Clinical proteomics. Protein annotation. Data mining in proteomics.
2. **Next generation sequencing and sequence analysis.** De novo sequencing, re-sequencing, and assembly. Expression estimation. Alternative splicing discovery. Pathway analysis. Chip-seq and RNA-Seq analysis. Metagenomics. SNPs prediction.
3. **High performance in bioinformatics.** Parallelization for biomedical analysis. Biomedical and biological databases. Data mining and biological text processing. Large scale biomedical data integration. Biological and medical ontologies. Novel architecture and technologies (GPU, P2P, Grid, etc) for bioinformatics.
4. **Biomedicine.** Biomedical computing. Personalized medicine. Nanomedicine. Medical education. Collaborative medicine. Biomedical signal analysis. Biomedicine in industry and society. Electrotherapy and radiotherapy.
5. **Biomedical engineering.** E-computer-assisted surgery. Therapeutic engineering. Interactive 3D modelling. Clinical engineering. Telemedicine. Biosensors and data acquisition. Intelligent instrumentation. Patient monitoring. Biomedical robotics. Bio-nanotechnology. Genetic engineering.
6. **Computational systems for modelling biological processes.** Inference of biological networks. Machine learning in bioinformatics. Classification for

biomedical data. Microarray data analysis. Simulation and visualization of biological systems. Molecular evolution and phylogenetic modeling.

7. **Healthcare and diseases.** Computational support for clinical decisions. Image visualization and signal analysis. Disease control and diagnosis. Genome-phenome analysis. Biomarker identification. Drug design. Computational immunology.
8. **E-health.** E-health technology and devices. E-health information processing. Telemedicine/E-health application and services. Medical image processing. Video techniques for medical images. Integration of classical medicine and E-health.
9. **COVID-19.** A special session analyzing different aspects, fields of application, and technologies that have been applied against COVID-19.

After a careful peer review and evaluation process (each submission was reviewed by at least 2, and on average 3.1, Program Committee members or additional reviewers), 75 papers were accepted, according to the recommendations of reviewers and the authors' preferences, to be included in the LNBI proceedings.

IWBBIO 2022 featured several Special Sessions, which are a very useful tool in order to complement the regular program with new and emerging topics of particular interest for the participating community. Special Sessions that emphasized multidisciplinary and transversal aspects, as well as cutting-edge topics were especially encouraged and welcomed, and in this edition of IWBBIO 2022 the following were received:

– **SS1. High-throughput Genomics: Bioinformatic Tools and Medical Applications.**

Genomics is concerned with the sequencing and analysis of an organism's genome. It is involved in the understanding of how every single gene can affect the entire genome. This goal is mainly afforded using the current, cost-effective, high-throughput sequencing technologies. These technologies produce a huge amount of data that usually require high-performance computing solutions and open new ways for the study of genomics, as well as transcriptomics, gene expression, and systems biology, among others. The continuous improvements and broader applications of sequencing technologies are producing a continuous new demand for improved high-throughput bioinformatics tools.

In this context, the generation, integration, and interpretation of genetic and genomic data is driving a new era of healthcare and patient management. Medical genomics (or genomic medicine) is an emerging discipline that involves the use of genomic information about a patient as part of the clinical care with diagnostic or therapeutic purposes to improve the health outcomes. Moreover, it can be considered a subset of precision medicine that has an impact in the fields of oncology, pharmacology, rare and undiagnosed diseases, and infectious diseases. The aim of this Special Session was to bring together researchers in medicine, genomics, and bioinformatics to translate medical genomics research into new diagnostic, therapeutic, and preventive medical approaches. Therefore, we invited authors to submit original research, new tools or pipelines, and update and review articles on relevant topics, such as (but not limited to):

- Tools for data pre-processing (quality control and filtering)
- Tools for sequence mapping
- Tools for the comparison of two read libraries without an external reference
- Tools for genomic variants (such as variant calling or variant annotation)
- Tools for functional annotation: identification of domains, orthologues, genetic markers, and controlled vocabulary (GO, KEGG, InterPro, etc.)
- Tools for gene expression studies and tools for Chip-Seq data
- Integrative workflows and pipelines

Organizers: M. Gonzalo Claros, Department of Molecular Biology and Biochemistry, University of Málaga, Spain; Javier Pérez Florido, Bioinformatics Research Area, Fundación Progreso y Salud, Seville, Spain; and Francisco M. Ortuño, Department of Computer Architecture and Technology, University of Granada, Spain.

– **SS2. Feature Selection, Extraction, and Data Mining in Bioinformatics: Approaches, Methods, and Adaptations.**

Various applications of bioinformatics, system biology, and biophysics measurement data mining require proper, accurate, and precise preprocessing or data transformation before the analysis itself. Here, the most important issues are covered by the feature selection and extraction techniques to translate the raw data into the inputs for the machine learning and multivariate statistic algorithms. Even if this is a complex task, it reduces the problem dimensionality, by removing redundant or irrelevant data, without affecting significantly the principal information. The methods and approaches are often conditioned by the physical properties of the measurement process, mathematically congruent description and parameterization, and biological aspects of specific tasks. With the increasing adoption of artificial intelligence methods to solve bioinformatics problems, it is necessary to understand the conditionality of such algorithms, to choose and use the correct approach and avoid misinterpretations, artefacts, and aliasing effects. The adoption often uses existing knowledge from different fields, and direct application might underestimate the required conditions and corrupt the analysis results. This Special Session provided a forum to discuss the multidisciplinary overlaps, development, implementation, and adoption of feature and selection methods for datasets with a biological origin in order to setup the pipeline from measurement design through signal processing to obtaining the results. The topic should cover theoretical questions, practical examples, and results verifications.

Organizer: Jan Urban, Laboratory of Signal and Image Processing, Institute of Complex Systems, South Bohemian Research Center of Aquaculture and Biodiversity of Hydrocenoses, Faculty of Fisheries and Protection of Waters, University of South Bohemia, Czech Republic.

– **SS3. Smart Healthcare Solutions for Handling COVID-19.**

Smart healthcare plays an important role towards providing robust solutions, especially for COVID-19 related problems, both locally and globally. Collection and interpretation of data worldwide and systematic research helps in identifying the potential solutions as well as predicting the future issues. This Special Session was

organized to emphasize the potential problems and the related solutions, focusing on the following topics:

- Smart wearable healthcare
- Microbiological analysis
- Minimal invasive sensors
- Biomedical waste management
- Drug-induced therapy
- Early prediction and diagnosis
- Biostatistical driven solution
- Explainable AI and deep learning driven solutions

Organizer: N. Sriraam, Department of Medical Electronics, M.S. Ramaiah Institute of Technology, India.

– **SS4. Computational Systems for Modeling of Medical Micro Sensors.**

Medical sensors are micro devices containing several parts mainly including micro-tubes, micro-valves, biological/body fluids (blood, plasma, saliva, etc.), and chemical materials (reagents and other materials). Microfluidics is an interdisciplinary field that involves the science and technology of fluid flow through systems with micro scales. Computational systems and engineering simulation are essential from the start to the end of the medical sensor design and development process. The main advantages of computational systems (AI, CFD, etc.) in medical sensors design and development are as follows:

- Improvement and optimization of design
- Acceleration in medical device innovation
- Reduction of cost and failure risk
- Reduction of production times and regulatory approval processes

The main objectives of this Special Session were as follows:

- To determine the role of computational systems (AI, CFD, etc.) in medical sensors design and development
- To determine the role of simulation in optimizing the analysis process and design of medical micro-sensors
- To discuss the use of computational fluid dynamics (CFD) in analyzing medical micro sensors
- To discuss the use of computational systems to combine engineering, biology, chemical, and other criteria

Organizers: Patrizia Piro and Behrouz Pirouz, Department of Civil Engineering, University of Calabria, Italy.

It is important to note that for the sake of consistency and readability the accepted papers are organized into 15 chapters over two volumes, essentially following the topics

list included in the call for papers. The first volume (LNBI 13346), entitled “Bioinformatics and Biomedical Engineering. Part I” is divided into eight main parts and includes the contributions on

1. Biomedical Computing
2. Biomedical Engineering
3. Biomedical Signal Analysis
4. Biomedicine. New Advances and Applications
5. Biosensors and Data Acquisition
6. Image Visualization and Signal Analysis in Biomedical Applications
7. Computational Support for Clinical Decisions
8. COVID-19. Bioinformatics and Biomedicine

The second volume (LNBI 13347), entitled “Bioinformatics and Biomedical Engineering. Part II” is divided into seven main parts and includes the contributions on:

1. Chip-seq and RNA-Seq Analysis
2. Bioinformatics and Biomarker Identification
3. Computational Proteomics
4. Computational Systems for Modelling Biological Processes
5. Feature Selection, Extraction, and Data Mining in Bioinformatics: Approaches, Methods, and Adaptations
6. Machine Learning in Bioinformatics
7. Next Generation Sequencing and Sequence Analysis

This 9th edition of IWBBIO was organized by the University of Granada. We wish to thank our main sponsor as well as the Department of Computer Architecture and Computer Technology at the University of Granada (CITIC-UGR) and International Society for Computational Biology (ISCB) for their support and grants. We also wish to thank the editors in charge of different international journals for their interest in publishing special issues of a selection of the best papers of IWBBIO 2022. In this edition of IWBBIO there were two awards (best contribution award and best contribution from student participant) sponsored by the Editorial Office of Genes, a MDPI journal.

We would also like to express our gratitude to the members of the different committees for their support, collaboration, and good work. We especially thank the Program Committee, the reviewers, and the Special Session organizers. We also want to express our gratitude to the EasyChair platform. Finally, we wish to thank Springer for their continuous support and cooperation.

April 2022

Ignacio Rojas  
 Olga Valenzuela  
 Fernando Rojas  
 Luis Javier Herrera  
 Francisco Ortuño



# Organization

## Conference Chairs

Ignacio Rojas	University of Granada, Spain
Olga Valenzuela	University of Granada, Spain
Fernando Rojas	University of Granada, Spain
Luis Javier Herrera	University of Granada, Spain
Francisco Ortuño	University of Granada, Spain

## Steering Committee

Miguel A. Andrade	University of Mainz, Germany
Hesham H. Ali	University of Nebraska, USA
Oresti Baños	University of Granada, Spain
Alfredo Benso	Politecnico di Torino, Italy
Larbi Boubchir	LIASD, University of Paris 8, France
Giorgio Buttazzo	Superior School Sant'Anna, Italy
Gabriel Caffarena	University San Pablo CEU, Spain
Mario Cannataro	Magna Graecia University of Catanzaro, Italy
Jose María Carazo	Spanish National Center for Biotechnology (CNB), Spain
Jose M. Cecilia	Universidad Católica San Antonio de Murcia (UCAM), Spain
M. Gonzalo Claros	University of Malaga, Spain
Joaquin Dopazo	Fundacion Progreso y Salud, Spain
Werner Dubitzky	University of Ulster, UK
Afshin Fassihi	Universidad Católica San Antonio de Murcia (UCAM), Spain
Jean-Fred Fontaine	University of Mainz, Germany
Humberto Gonzalez	University of Basque Country (UPV/EHU), Spain
Concettina Guerra	Georgia Tech, USA
Roderic Guigo	Pompeu Fabra University, Spain
Andy Jenkinson	Karolinska Institute, Sweden
Craig E. Kapfer	Reutlingen University, Germany
Narsis Aftab Kiani	European Bioinformatics Institute (EBI), UK
Natividad Martinez	Reutlingen University, Germany
Marco Masseroli	Politechnical University of Milan, Italy
Federico Moran	Complutense University of Madrid, Spain

Cristian R. Munteanu	University of A Coruña, Spain
Jorge A. Naranjo	NYU Abu Dhabi, Abu Dhabi
Michael Ng	Hong Kong Baptist University, China
Jose L. Oliver	University of Granada, Spain
Juan Antonio Ortega	University of Seville, Spain
Fernando Rojas	University of Granada, Spain
Alejandro Pazos	University of A Coruña, Spain
Javier Perez Florido	Genomics and Bioinformatics Platform of Andalusia, Spain
Violeta I. Pérez Nueno	Inria Nancy-Grand Est and Loria, France
Horacio Pérez-Sánchez	Universidad Católica San Antonio de Murcia (UCAM), Spain
Alberto Policriti	Università di Udine, Italy
Omer F. Rana	Cardiff University, UK
M. Francesca Romano	Superior School Sant' Anna, Italy
Yvan Saeys	Ghent University, Belgium
Vicky Schneider	The Genome Analysis Centre (TGAC), UK
Ralf Seepold	HTWG Konstanz, Germany
Mohammad Soruri	University of Birjand, Iran
Yoshiyuki Suzuki	Tokyo Metropolitan Institute of Medical Science, Japan
Oswaldo Trelles	University of Malaga, Spain
Shusaku Tsumoto	Shimane University, Japan
Renato Umeton	Dana-Farber Cancer Institute and Massachusetts Institute of Technology, USA
Jan Urban	University of South Bohemia, Czech Republic
Alfredo Vellido	Polytechnic University of Catalonia, Spain
Wolfgang Wurst	GSF National Research Center of Environment and Health, Germany

## **Program Committee and Additional Reviewers**

Magda Abdellattif	Taif University, Saudi Arabia
Fares Al-Shargie	American University of Sharjah, United Arab Emirates
Jesus Alcalá-Fdez	University of Granada, Spain
Hesham Ali	University of Nebraska Omaha, USA
Georgios Anagnostopoulos	Florida Institute of Technology, USA
Patrizio Arrigo	SCITEC, Italy
Gajendra Kumar Azad	Patna University, India
Hazem Bahig	Ain Shams University, Egypt
Ugo Bastolla	Centro de Biología Molecular Severo Ochoa, Spain

Payam Behzadi	Islamic Azad University, Iran
Alfredo Benso	Politecnico di Torino, Italy
Anna Bernasconi	Politecnico di Milano, Italy
Mahua Bhattacharya	Indian Institute of Information Technology and Management, Gwalior, India
Paola Bonizzoni	Università di Milano-Bicocca, Italy
Larbi Boubchir	University of Paris 8, France
Hacene Boukari	Delaware State University, USA
Gabriel Caffarena	Universidad CEU San Pablo, Spain
Mario Cannataro	Magna Graecia University of Catanzaro, Italy
Jose Maria Carazo	National Center for Biotechnology (CNB-CSIC), Spain
Rita Casadio	University of Bologna, Italy
Daniel Castillo-Secilla	University of Granada, Spain
Claudia Cava	IBFM-CNR, Italy
Francisco Cavas-Martínez	Technical University of Cartagena, Spain
Chinmay Chakraborty	Birla Institute of Technology, India
Ting-Fung Chan	The Chinese University of Hong Kong, Hong Kong
Satyendra Chandra Tripathi	AIIMS Nagpur, India
Kun-Mao Chao	National Taiwan University, Taiwan
Bolin Chen	Northwestern Polytechnical University, China
Brian Chen	Lehigh University, USA
Chuming Chen	University of Delaware, USA
Jeonghyeon Choi	Georgia Regents University, USA
Javier Cifuentes Faura	University of Murcia, Spain
M. Gonzalo Claros	Universidad de Málaga, Spain
Zhu Daming	Shandong University, China
Bhaskar Dasgupta	University of Illinois at Chicago, USA
Alexandre G. De Brevern	INSERM UMR-S, Université Paris Cité, France
Javier De Las Rivas	Cancer Research Center (CiC-IBMCC, CSIC/USAL/IBSAL), Spain
Ricardo de Matos Simoes	Harvard University, USA
Marie-Dominique Devignes	Loria, France
Paolo Di Giamberardino	Sapienza University of Rome, Italy
Maria Natalia Dias Soeiro Cordeiro	University of Porto, Portugal
Marko Djordjevic	University of Belgrade, Serbia
Joaquin Dopazo	Fundacion Progreso y Salud, Spain
Mohammed Elmogy	Mansoura University, Egypt
Gionata Fragomeni	Magna Graecia University of Catanzaro, Italy
Hassan Ghazal	Mohammed I University, Morocco

Razvan Ghinea	University of Granada, Spain
Christophe Guyeux	University of Franche-Comté, France
Pietro Hiram Guzzi	Magna Graecia University of Catanzaro, Italy
Michael Hackenberg	University of Granada, Spain
Nurit Haspel	University of Massachusetts Boston, USA
Morihiro Hayashida	National Institute of Technology, Matsue College, Japan
Luis Herrera	University of Granada, Spain
Ralf Hofstaedt	Bielefeld University, Germany
Jingshan Huang	University of South Alabama, USA
Cosimo Ieracitano	University Mediterranea of Reggio Calabria, Italy
Chang-Hwan Im	Hanyang University, South Korea
Hamed Khodadadi	Islamic Azad University, Iran
Narsis Kiani	Karolinska Institute, Sweden
Tomas Koutny	University of West Bohemia, Czech Republic
Konstantin Krutovsky	Georg-August-University of Göttingen, Germany
Chen Li	Monash University, Australia
Shuai Cheng Li	City University of Hong Kong, Hong Kong
Li Liao	University of Delaware, USA
Hongfei Lin	Dalian University of Technology, China
Javier Martin	IPBLN-CSIC, Spain
Francisco Martínez-Álvarez	Universidad Pablo de Olavide, Spain
Roderick Melnik	Wilfrid Laurier University, Canada
Francesco Carlo Morabito	University of Reggio Calabria, Italy
Enrique Muro	Johannes Gutenberg University, Germany
Kenta Nakai	University of Tokyo, Japan
Isabel Nepomuceno	University of Seville, Spain
Dang Ngoc Hoang Thanh	University of Economics Ho Chi Minh City, Vietnam
José Luis Oliveira	University of Aveiro, Portugal
Yuriy Orlov	Institute of Cytology and Genetics, Russia
Juan Antonio Ortega	University of Seville, Spain
Andres Ortiz	University of Malaga, Spain
Francisco Manuel Ortuño	University of Granada, Spain
Motonori Ota	Nagoya University, Japan
Mehmet Akif Ozdemir	Izmir Katip Celebi University, Turkey
Joel P. Arrais	University of Coimbra, Portugal
Paolo Paradisi	ISTI-CNR, Italy
Taesung Park	Seoul National University, South Korea
Antonio Pinti	I3MTO Orléans, France
Yuri Pirola	Università degli Studi di Milano-Bicocca, Italy
Joanna Polanska	The Silesian University of Technology, Poland

Alberto Policriti	University of Udine, Italy
Hector Pomares	University of Granada, Spain
María M. Pérez	University of Granada, Spain
Hossein Rabbani	Isfahan University of Medical Sciences, Iran
Amgad Rabie	Ain Shams University, Egypt
Julietta Rau	Istituto di Struttura della Materia, Italy
Khalid Raza	Jamia Millia Islamia, India
Jairo Rocha	University of the Balearic Islands, Spain
Maria Rodriguez Martinez	IBM Zurich Research Laboratory, Switzerland
Fernando Rojas	University of Granada, Spain
Ignacio Rojas	University of Granada, Spain
Gregorio Rubio	Universitat Politècnica de València, Spain
Irena Rusu	LINA, University of Nantes, France
Michael Sadovsky	Institute of Computational Modelling of SB RAS, Russia
Beata Sarecka-Hujar	Medical University of Silesia in Katowice, Poland
Jean-Marc Schwartz	University of Manchester, UK
Russell Schwartz	Carnegie Mellon University, USA
Preeti Singh	UIET, India
Surinder Singh	Panjab University, India
Sónia Sobral	Universidade Portucalense, Portugal
Jiangning Song	Monash University, Australia
Joe Song	New Mexico State University, USA
Natarajan Sriraam	M.S. Ramaiah Institute of Technology, India
Jiangtao Sun	Beihang University, China
Wing-Kin Sung	National University of Singapore, Singapore
Prashanth Suravajhala	Amrita University Kerala, India
Martin Swain	Aberystwyth University, UK
Sing-Hoi Sze	Texas A&M University, USA
Alessandro Tonacci	IFC-CNR, Italy
Carolina Torres	University of Granada, Spain
Marcos Roberto Tovani Palone	University of São Paulo, Brazil
Shusaku Tsumoto	Shimane University, Japan
Renato Umeton	Massachusetts Institute of Technology, USA
Jan Urban	Institute of Complex Systems, FFPW USB, Czech Republic
Olga Valenzuela	University of Granada, Spain
Alfredo Vellido	Universitat Politècnica de Catalunya, Spain
Jianxin Wang	Central South University, China
Jiayin Wang	Xi'an Jiaotong University, China
Junbai Wang	Radium Hospital, Norway
Lusheng Wang	City University of Hong Kong, Hong Kong

Ka-Chun Wong

Phil Yang

Jin Zhang

Yudong Zhang

Zhongming Zhao

Huiru Zheng

Shanfeng Zhu

City University of Hong Kong, Hong Kong

George Mason University, USA

Washington University in St. Louis, USA

University of Leicester, UK

University of Texas Health Science Center at  
Houston, USA

University of Ulster, UK

Fudan University, China

## Contents – Part II

### Chip-Seq and RNA-Seq Analysis

Integrative Analysis of Ovarian Serious Adenocarcinoma to Understand Disease Network Biology .....	3
<i>Sahar Qazi and Khalid Raza</i>	
GAGAM: A Genomic Annotation-Based Enrichment of scATAC-seq Data for Gene Activity Matrix .....	18
<i>Lorenzo Martini, Roberta Bardini, Alessandro Savino, and Stefano Di Carlo</i>	
Finding Significantly Enriched Cells in Single-Cell RNA Sequencing by Single-Sample Approaches .....	33
<i>Anna Mrukwa, Michal Marczyk, and Joanna Zyla</i>	
Comparison of Stranded and Non-stranded RNA-Seq in Predicting Small RNAs in a Non-model Bacterium .....	45
<i>Karel Sedlar and Ralf Zimmer</i>	
Comparative Study of Synthetic Bulk RNA-Seq Generators .....	57
<i>Felitsiya Shakola, Dean Palejev, and Ivan Ivanov</i>	
Investigating Sources of Zeros in 10× Single-Cell RNAseq Data .....	71
<i>Hanna Slowik, Joanna Zyla, and Michal Marczyk</i>	

### Bioinformatics and Biomarker Identification

Exhaled Breath Condensate Study for Biomarkers Discovery .....	83
<i>S. Patsiris, T. Exarchos, and P. Vlamos</i>	
Statistical Learning Analysis of Thyroid Cancer Microarray Data .....	90
<i>Iván Petrini, Rocío L. Cecchini, Marilina Mascaró, Ignacio Ponzoni, and Jessica A. Carballido</i>	
Migrating CUDA to oneAPI: A Smith-Waterman Case Study .....	103
<i>Manuel Costanzo, Enzo Rucci, Carlos García-Sánchez, Marcelo Naiouf, and Manuel Prieto-Matías</i>	

**Computational Proteomics**

Fuzzy-Inference System for Isotopic Envelope Identification in Mass Spectrometry Imaging Data ..... 119  
*Anna Glodek*

Receptor Tyrosine Kinase KIT: A New Look for an Old Receptor ..... 133  
*Julie Ledoux and Luba Tchertanov*

Human Vitamin K Epoxide Reductase as a Target of Its Redox Protein ..... 138  
*Julie Ledoux, Maxim Stolyarchuk, and Luba Tchertanov*

A Distance Geometry Procedure Using the Levenberg-Marquardt Algorithm and with Applications in Biology but Not only ..... 142  
*Douglas S. Gonçalves and Antonio Mucherino*

A Semi-supervised Graph Deep Neural Network for Automatic Protein Function Annotation ..... 153  
*Akrem Sellami, Bishnu Sarker, Salvatore Tabbone, Marie-Dominique Devignes, and Sabeur Aridhi*

**Computational Systems for Modelling Biological Processes**

Strong Prevalence of the Function over Taxonomy in Human rRNA Genes ..... 169  
*Yana Nedorez and Michael Sadovsky*

A Methodology for Co-simulation-Based Optimization of Biofabrication Protocols ..... 179  
*Leonardo Giannantoni, Roberta Bardini, and Stefano Di Carlo*

A 3D Multicellular Simulation Layer for the Synthetic Biology CAD Infobotics Workbench Suite ..... 193  
*Richard Oliver Matzko, Laurentiu Mierla, and Savas Konur*

Integrating *in-vivo* Data in CFD Simulations and in *in-vitro* Experiments of the Hemodynamic in Healthy and Pathologic Thoracic Aorta ..... 208  
*Alessandro Mariotti, Emanuele Gasparotti, Emanuele Vignali, Pietro Marchese, Simona Celi, and Maria Vittoria Salvetti*

Sensitivity Analysis of Adhesion in Computational Model of Elastic Doublet ..... 220  
*Alžbeta Bohiniková, Iveta Jančigová, Ivan Cimrák, and James J. Feng*



Increasing the Accuracy of Optipharm’s Virtual Screening Predictions by Implementing Molecular Flexibility .....	234
<i>Savíns Puertas-Martín, Juana L. Redondo, Ester M. Garzón, Horacio Pérez-Sánchez, and Pilar M. Ortigosa</i>	

## **Feature Selection, Extraction, and Data Mining in Bioinformatics: Approaches, Methods and Adaptations**

Comparisons of Knowledge Graphs and Entity Extraction in Breast Cancer Subtyping Biomedical Text Analysis .....	249
<i>Jean Davidson, Grif Hawblitzel, McClain Kressman, Andrew Doud, Harsha Lakshman Kumar, Ella Thomas, Paul Kim, Ava Jakusovszky, and Paul Anderson</i>	

Towards XAI: Interpretable Shallow Neural Network Used to Model HCP’s fMRI Motor Paradigm Data .....	260
<i>José Diogo Marques dos Santos and José Paulo Marques dos Santos</i>	

A Deep Learning-Based Method for Uncovering GPCR Ligand-Induced Conformational States Using Interpretability Techniques .....	275
<i>Mario A. Gutiérrez-Mondragón, Caroline König, and Alfredo Vellido</i>	

Data Transformation for Clustering Utilization for Feature Detection in Mass Spectrometry .....	288
<i>Vojtech Barton and Helena Skutkova</i>	

Spolmap: An Enriched Visualization of CRISPR Diversity .....	300
<i>Christophe Guyeux, Guislaine Refrégier, and Christophe Sola</i>	

How to Compare Various Clustering Outcomes? Metrics to Investigate Breast Cancer Patient Subpopulations Based on Proteomic Profiles .....	309
<i>Joanna Tobiasz and Joanna Polanska</i>	

Sperm-cell Detection Using YOLOv5 Architecture .....	319
<i>Michal Dobrovolny, Jakub Benes, Ondrej Krejcar, and Ali Selamat</i>	

## **Machine Learning in Bioinformatics**

Comparative Analysis of Supervised Cell Type Detection in Single-Cell RNA-seq Data .....	333
<i>Akram Vasighizaker, Sheena Hora, Yash Trivedi, and Luis Rueda</i>	

<b>PathWeigh – Quantifying the Behavior of Biochemical Pathway Cascades</b> .....	346
<i>Dani Livne and Sol Efroni</i>	
<b>Translational Challenges of Biomedical Machine Learning Solutions in Clinical and Laboratory Settings</b> .....	353
<i>Carlos Vega, Miroslav Kratochvil, Venkata Satagopam, and Reinhard Schneider</i>	
<b>Human Multi-omics Data Pre-processing for Predictive Purposes Using Machine Learning: A Case Study in Childhood Obesity</b> .....	359
<i>Álvaro Torres-Martos, Augusto Anguita-Ruiz, Mireia Bustos-Aibar, Sofia Cámara-Sánchez, Rafael Alcalá, Concepción M. Aguilera, and Jesús Alcalá-Fdez</i>	
<b>Feature Density as an Uncertainty Estimator Method in the Binary Classification Mammography Images Task for a Supervised Deep Learning Model</b> .....	375
<i>Ricardo Javier Fuentes-Fino, Saúl Calderón-Ramírez, Enrique Domínguez, Ezequiel López-Rubio, Marco A. Hernandez-Vasquez, and Miguel A. Molina-Cabello</i>	
<b>Iterative Clustering for Differential Gene Expression Analysis</b> .....	389
<i>Olga Georgieva</i>	
<b>Comparison of Batch Effect Removal Methods for High Dimensional Mass Cytometry Data</b> .....	399
<i>Aleksandra Suwalska, Nelita du Plessis-Burger, Gian van der Spuy, and Joanna Polanska</i>	
<b>Next Generation Sequencing and Sequence Analysis</b>	
<b>Evaluating Performance of Regression and Classification Models Using Known Lung Carcinomas Prognostic Markers</b> .....	413
<i>Shrikant Pawar, Karuna Mittal, and Chandrajit Lahiri</i>	
<b>Approximate Pattern Matching Using Search Schemes and In-Text Verification</b> .....	419
<i>Luca Renders, Lore Depuydt, and Jan Fostier</i>	
<b>KFinger: Capturing Overlaps Between Long Reads by Using Lyndon Fingerprints</b> .....	436
<i>Paola Bonizzoni, Alessia Petescia, Yuri Pirola, Raffaella Rizzi, Rocco Zaccagnino, and Rosalba Zizza</i>	

**Can We Detect T Cell Receptors from Long-Read RNA-Seq Data? . . . . . 450**  
*Justyna Mika, Serge M. Candéias, Christophe Badie,  
and Joanna Polanska*

**Author Index . . . . . 465**

# Contents – Part I

## Biomedical Computing

Calculation of DNA Strand Breaks by Types of Electron Interaction with Monte Carlo Simulation .....	3
--	---

*Youssef Lamghari, Huizhong Lu, and M'hamed Bentourkia*

Linear Predictive Modeling for Immune Metabolites Related to Other Metabolites .....	16
---	----

*Jana Schwarzerova, Iro Pierides, Karel Sedlar, and Wolfram Weckwerth*

Modelling of Arbitrary Shaped Channels and Obstacles by Distance Function .....	28
--	----

*Kristína Kovalčíková Ďuračková, Alžbeta Bugáňová, and Ivan Cimrák*

Gene Expression Profiles of Visceral and Subcutaneous Adipose Tissues in Children with Overweight or Obesity: The KIDADIPOSEQ Project .....	42
--	----

*Mireia Bustos-Aibar, Augusto Anguita-Ruiz, Álvaro Torres-Martos,  
Jesús Alcalá-Fdez, Francisco Javier Ruiz-Ojeda,  
Marjorie Reyes-Farias, Andrea Soria-Gondek, Laura Herrero,  
David Sánchez-Infantes, and Concepción María Aguilera*

The Role of Astrocytes in Alzheimer's Disease Progression .....	47
---	----

*Swadesh Pal and Roderick Melnik*

Effects of Random Inputs and Short-Term Synaptic Plasticity in a LIF Conductance Model for Working Memory Applications .....	59
---	----

*Thi Kim Thoa Thieu and Roderick Melnik*

## Biomedical Engineering

Thermal Effects of Manual Therapy in Low Back Pain: A Pilot Study .....	75
---	----

*Andrea Rosales-Hernandez, Daniela Viguera-Becerril,  
Arelly G. Morales-Hernandez, Sandra M. Chavez-Monjaras,  
Luis A. Morales-Hernandez, and Irving A. Cruz-Albarran*

Bone Health Parameters in Young Adult Female Handball Players .....	90
---	----

*Elie Maliha, Anthony Khawaja, Hechmi Toumi, Rachid Jennane,  
Antonio Pinti, and Rawad El Hage*

Adaptative Modelling of the Corneal Architecture in a Free-of-Stress State  
in Incipient Keratoconus ..... 108  
*Francisco Cavas, Carmelo Gómez, José S. Velázquez, David Piñero,  
Francisco L. Sáez-Gutiérrez, and Jorge Alió*

Design of an Analysis Method for the Human Cornea’s Bilateral  
Symmetry. A Case-Study in Healthy Patients ..... 119  
*Francisco Cavas, José S. Velázquez, Carmelo Gómez, Jorge Mira,  
Francisco L. Sáez-Gutiérrez, and Jorge Alió*

**Biomedical Signal Analysis**

Automated TTC Image-Based Analysis of Mouse Brain Lesions ..... 135  
*Gerasimos Damigos, Nefeli Zerva, Angelos Pavlopoulos,  
Konstantina Chatzikyriakou, Argyro Koumenti, Konstantinos Moustakas,  
Constantinos Pantos, Iordanis Mourouzis, Athanasios Loubopoulos,  
and Evangelia I. Zacharaki*

PET-Neuroimaging and Neuropsychological Study for Early Cognitive  
Impairment in Parkinson’s Disease ..... 143  
*Sergey Lytaev*

Architecture and Calibration of a Multi-channel Electrical Impedance  
Myograph ..... 154  
*Edson Rodrigues, Erick Dario León Bueno de Camargo,  
and Olavo Luppi Silva*

**Biomedicine. New Advances and Applications**

Advanced Incremental Attribute Learning Clustering Algorithm  
for Medical and Healthcare Applications ..... 171  
*Siwar Gorra, Fahmi Ben Rejab, and Kaouther Nouira*

Assessment of Inflammation in Non-calcified Artery Plaques with Dynamic  
18F-FDG-PET/CT: CT Alone, Does-It Detect the Vulnerable Plaque? ..... 184  
*Mamdouh S. Al-enezi, Abdelouahed Khalil, Tamas Fulop, Éric Turcotte,  
and M’hamed Bentourkia*

Comparative Analysis of the Spatial Structure Chloroplasts  
and Cyanobacteria Photosynthetic Systems I and II Genes ..... 197  
*Maria Senashova and Michael Sadovsky*

Unsupervised Classification of Some Bacteria with 16S RNA Genes ..... 205  
*Agnia Teterleva, Vladislav Abramov, Andrey Morgun, Irina Larionova,  
and Michael Sadovsky*

Modern Approaches to Cancer Treatment ..... 216  
*Snezhana M. Bakalova, Milena Georgieva, and Jose Kaneti*

A Service for Flexible Management and Analysis of Heterogeneous  
 Clinical Data ..... 227  
*Sandro Hurtado, José García-Nieto, and Ismael Navas-Delgado*

**Biosensors and Data Acquisition**

Reconfigurable Arduino Shield for Biosignal Acquisition ..... 241  
*Leozítor Floro de Souza, Fábio Iaione, and Shih Ting Ju*

Smart Watch for Smart Health Monitoring: A Literature Review ..... 256  
*Avnish Singh Jat and Tor-Morten Grønli*

Data Quality Enhancement for Machine Learning on Wearable ECGs ..... 269  
*Balázs Molnár, László Micsinyei, Gábor Perlaki, Gergely Orsi,  
 László Hejmel, Tamás Dóczy, József Janszky, Norbert Laky, and Ákos Tényi*

**Image Visualization and Signal Analysis in Biomedical Applications**

Measurable Difference Between Malignant and Benign Tumor  
 of the Thyroid Gland Recognizable Using Echogenicity Index  
 in Ultrasound B-MODE Imaging: An Experimental Blind Study ..... 283  
*Jiri Blahuta, Tomas Soukup, Jan Lavrincik, Lukas Pavlik,  
 and Zuzana Repaska*

Initial Prototype of Low-Cost Stool Monitoring System for Early  
 Detection of Diseases ..... 297  
*José Luis López-Ruiz, David Díaz-Jiménez, Alicia Montoro-Lendínez,  
 and Macarena Espinilla*

Cerebral Activation in Subjects with Developmental Coordination  
 Disorder: A Pilot Study with PET Imaging ..... 309  
*Marie Farmer, Bernard Echenne, and M’hamed Bentourkia*

On the Use of Explainable Artificial Intelligence for the Differential  
 Diagnosis of Pigmented Skin Lesions ..... 319  
*Sandro Hurtado, Hossein Nematzadeh, José García-Nieto,  
 Miguel-Ángel Berciano-Guerrero, and Ismael Navas-Delgado*

Estimating Frontal Body Landmarks from Thermal Sensors Using  
 Residual Neural Networks ..... 330  
*Aurora Polo-Rodríguez, Marcos Lupión, Pilar M. Ortigosa,  
 and Javier Medina-Quero*

<b>NMF for Quality Control of Multi-modal Retinal Images for Diagnosis of Diabetes Mellitus and Diabetic Retinopathy</b> .....	343
<i>Anass Benali, Laura Carrera, Ann Christin, Ruben Martín, Anibal Alé, Marina Barraso, Carolina Bernal, Sara Marín, Silvia Feu, Josep Rosinés, Teresa Hernandez, Irene Vilá, Cristian Oliva, Irene Vinagre, Emilio Ortega, Marga Gimenez, Enric Esmatjes, Javier Zarranz-Ventura, Enrique Romero, and Alfredo Vellido</i>	
<b>Radiomic-Based Lung Nodule Classification in Low-Dose Computed Tomography</b> .....	357
<i>Wojciech Prazuch, Malgorzata Jelitto-Gorska, Agata Durawa, Katarzyna Dziadziuszko, and Joanna Polanska</i>	
<b>Segmentation of Brain MR Images Using Quantum Inspired Firefly Algorithm with Mutation</b> .....	364
<i>Alokeparna Choudhury, Sourav Samanta, Sanjoy Pratihar, and Oishila Bandyopadhyay</i>	
<b>Computational Support for Clinical Decisions</b>	
<b>Single-Channel EEG Detection of REM Sleep Behaviour Disorder: The Influence of REM and Slow Wave Sleep</b> .....	381
<i>Irene Rechichi, Federica Amato, Alessandro Cicolin, and Gabriella Olmo</i>	
<b>A Deep Learning Framework for the Prediction of Conversion to Alzheimer Disease</b> .....	395
<i>Sofia Ostellino, Alfredo Benso, and Gianfranco Politano</i>	
<b>Gene Expression Tools from a Technical Perspective: Current Approaches and Alternative Solutions for the KnowSeq Suite</b> .....	404
<i>Daniel Castillo-Secilla, Daniel Redondo-Sánchez, Luis Javier Herrera, Ignacio Rojas, and Alberto Guillén</i>	
<b>COVID-19. Bioinformatics and Biomedicine</b>	
<b>Optimal Chair Location Through a Maximum Diversity Problem Genetic Algorithm Optimization</b> .....	417
<i>Rubén Ferrero-Guillén, Javier Díez-González, Paula Verde, Alberto Martínez-Gutiérrez, José-Manuel Alija-Pérez, and Rubén Álvarez</i>	
<b>Collecting SARS-CoV-2 Encoded miRNAs via Text Mining</b> .....	429
<i>Alexandra Schubö, Armin Hadziahmetovic, Markus Joppich, and Ralf Zimmer</i>	

COVID-19 Severity Classification Using a Hierarchical Classification Deep Learning Model .....	442
<i>Sergio Ortiz, Juan Carlos Morales, Fernando Rojas, Olga Valenzuela, Luis Javier Herrera, and Ignacio Rojas</i>	
The Role of Information Sources, Trust in Information Sources, and COVID-19 Conspiracy Theory in the Compliance with COVID-19 Related Measures .....	453
<i>Ana Jovančević, Izabel Cvetković, and Nebojša Milićević</i>	
<b>Author Index</b> .....	459



# **Chip-Seq and RNA-Seq Analysis**



# Integrative Analysis of Ovarian Serious Adenocarcinoma to Understand Disease Network Biology

Sahar Qazi  and Khalid Raza  

Department of Computer Science, Jamia Millia Islamia, New Delhi 110025, India  
{rs.sahar1900560, kraza}@jmi.ac.in

**Abstract.** Ovarian cancer (OC) is the third leading gynecological malignancy in females that is silent and leads to significant deaths annually. As per GLOBOCAN 2020 statistics, Asia recorded a total of 100,854 ovarian cancer incidence cases with China and India leading the cases with 34.2% and 24.8% respectively. This paper aims to identify the genes that regulate ovarian cancer network biology by integrating high-grade serous ovarian adenocarcinoma data from TCGA and GEO databases. The data has been used to detect differentially expressed genes (DEGs), and further to assess the potential of the seed genes for disease-gene associations (DGA), principal component analysis (PCA), and Kaplan Meier (KM) survival estimations to give insights about these genes function in ovarian cancer pathway. We conclude that genes – *CLDN3*, *CLDN4*, *NFKB1*, *GSN*, *MUC16*, *NANOG*, *FKBP10*, and *CD274* are highly significant and influential in dominating ovarian serous adenocarcinoma in females and must be further deployed to construct a specific ovarian cancer network depicting functional attributes of each of these genes.

**Keywords:** Differentially expressed genes (DEGs) · Disease-gene associations (DGA) · Kaplan Meier analysis · Ovarian serous adenocarcinoma · Principal component analysis (PCA)

## 1 Introduction

Ovarian cancer ranks the third position after breast and cervical cancers that predominate lethality in females every year [1]. According to GLOBOCAN 2020 statistics, Asia recorded a total of 100,854 ovarian cancer incidences with China and India leading the cases with 34.2% and 24.8% respectively [2]. This increment in incidence cases directly points to the loopholes that exist in the medical infrastructure and lack of awareness in the general public. There are two factors that have been studied to cause this gynecological malignancy – *a) Intrinsic factors* – the genetic makeup of an individual, age, family history, epigenetic alterations, etc.; *b) Extrinsic factors* – environment and lifestyle of an individual [1, 3]. Furthermore, because of the lack of specific prognostic indicators for the disease, ovarian cancer remains undetected for a majority of the patients in its initial proliferation in the body as its symptoms coincide with various other medical problems

[4]. Various studies have revealed the median age of detection of ovarian cancer in females is ~ 60 years [5].

With translational bioinformatics, epigenomics, epi-informatics, molecular biology, systems biology, researchers have dexterously worked to decode the stealth behavior of ovarian cancer. Recent studies have revealed that many genes get hypomethylated (over-expressed) in case of ovarian cancer such as – members of POTE family (POTEC, POTEE, POTEF), BRCA1, NOTCH, SAT2, BRCA2, FOXM1, CCNE1, CLDN4, BORIS, IGF2, SNCG, MAPK, MAL, WFDC2, FOLR1, COL18A1, FLJ12988, CLASP1, TRAIL, etc. [6–10]. However, all these genes are also found to cause other malignancies such as – colorectal cancer, breast cancer, cervical cancer, non-small cell lung cancer, etc. [11–13]. Even the KEGG pathways [14] don't report specific pathways for ovarian serous adenocarcinoma. Therefore, there is an urgent need to identify significant genes playing a role in ovarian cancer network biology.

This paper aims to identify the genes that regulate ovarian cancer network biology by integrating high-grade serous ovarian adenocarcinoma data from TCGA and GEO databases. The data has been used to check for differentially expressed genes (DEGs), and further to assess the potential of the seed genes for disease-gene associations (DGA), principal component analysis (PCA), and Kaplan Meier (KM) survival estimation to give insights about these genes function in ovarian cancer pathway.

## 2 Material and Methods

### 2.1 Data Retrieval

- a) *The Cancer Genome Atlas Program (TCGA)*: The TCGA [15] was first explored to retrieve gene expression data for serous ovarian cancer. MeSH terms such as – {“gene expression of ovarian cancer”, “gene expression of ovarian carcinoma”, “gene expression of ovarian adenocarcinoma”} were used to search for relevant datasets for this study.
- b) *Gene Expression Omnibus (GEO)*: Gene Expression Omnibus (GEO) of the National Centre for Biotechnology Information (NCBI) [16] was also used to retrieve gene expression datasets. Again, MeSH terms such as – {“gene expression of ovarian cancer”, “gene expression of ovarian carcinoma”, “gene expression of ovarian adenocarcinoma”} were used to search for relevant datasets.

A total of 25,229 datasets were retrieved from both TCGA and GEO, however, after selective streamlining procedure, only 2567 datasets were obtained for MeSH term = “gene expression of ovarian adenocarcinoma”. After final sorting, from GEO we selected expression profiles: GSE185008, GSE157153, GSE111776, GSE181955, GSE168930, GSE154762, GSE185008, GSE151335, GSE171033, GSE171032, GSE166539, GSE162626, GSE142310, GSE114332, GSE115481, GSE118828, GSE99217, GSE90125, GSE108084, GSE84539, while from TCGA we selected TCGA-61–2113, TCGA-20–0991, TCGA-24–1426, TCGA-09–2051, TCGA-61–1998, TCGA-23–2078, TCGA-24–1431, TCGA-24–1845, TCGA-29–1763, TCGA-23–1116, TCGA-25–2042, TCGA-61–2110, TCGA-13–1492, TCGA-29–1770, TCGA-13–0920, TCGA-61–2003, TCGA-24–2280, TCGA-24–2293, TCGA-23–2084, TCGA-13–1477.

## 2.2 Differentially Expressed Genes (DEGs) Identification

Gene expression profiling was further done on selected datasets that had methylations status of serous ovarian adenocarcinoma only. The datasets were pre-processed and normalized to reduce redundancy and erroring. Fold-change (FC) statistics and p-values were considered to identify significant DEGs. FC-method is a statistical measure that showcases the altered expression of genes over two conditions – cancerous samples and normal ones. In this case, we deploy for a log<sub>2</sub>-foldchange, wherein all values greater than 0.5849 were considered as up-regulated whereas all values less than -0.5849 (or FC = 0.666) were be down-regulated genes [17]. LogFC function was used to check the expression levels in RStudio [18, 19]. To control the false discovery rate (FDR) in the analysis, *Benjamini & Hochberg* algorithm [23] was deployed with a significance cut-off set defined as 0.05.

## 2.3 Disease-Gene Association analysis

These DEGs were further submitted in DisGeNET [20] to understand the basic disease-gene association (DGA) of each DEG identified in the study.

## 2.4 Principal Component Analysis and Kaplan-Meiers Survival Estimation

These seed genes were submitted for a principal component analysis (PCA) and Kaplan-Meiers (KM) survival estimation. PCA analysis was executed using an online webserver named – Principal Components Analysis Online [21] and UALCAN [25] while KM plots were plotted using the KMplotter for ovarian cancer [22] that uses data from GEO, TCGA, and EGA. Figure 1 gives a graphical overview of the entire analysis performed in the study.

## 2.5 Gene Set Enrichment Analysis (GSEA)

Gene set enrichment analysis includes – gene ontology (GO) analysis, pathway enrichment, disease-drug associations etc. that eventually allow to assign functional features based on biological processes, molecular function and cellular localization aspects to a set of genes. Based on the genetic expression of genes, GO enrichment displays whether the genes are overrepresented or under-represented based on several annotations [17]. We deployed g:Profiler for the execution of GO enrichment analysis [34] and Enrichr [35].

## 2.6 Construction of Gene Regulatory Network and Analysis

GeneMania [36] plugin in Cytoscape was deployed to analyse the seed genes and also to construct a gene regulatory network (GRN) using them. Gene regulatory network (GRN) construction is an essential step towards screening the significance of each of the seed gene in myriad biological processes, topological analysis, network module identification further provides an insight towards specific drug targets. After the interaction network is retrieved, we employed OmicsNet visualization webtool [37] for network visualization

and topological analysis. Consensus Pathway Analysis [38] was used to predict highly significant pathways and processes from KEGG [39] and GO [40] based on significant p-values.

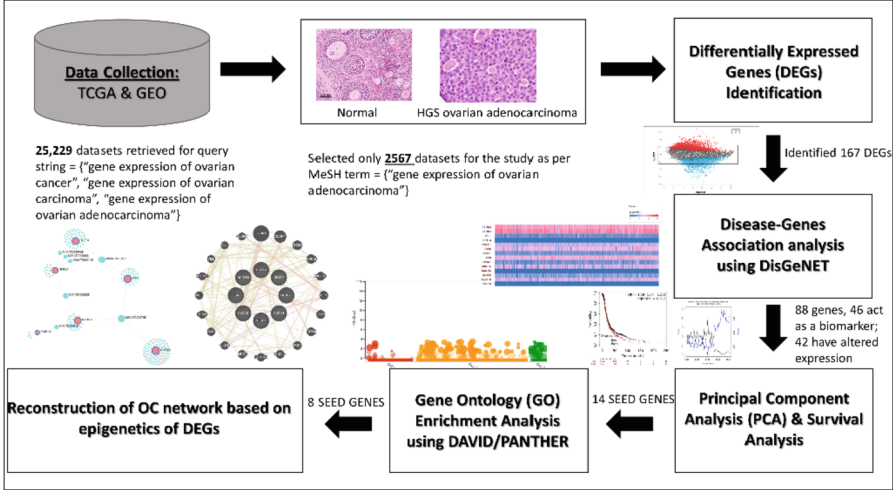
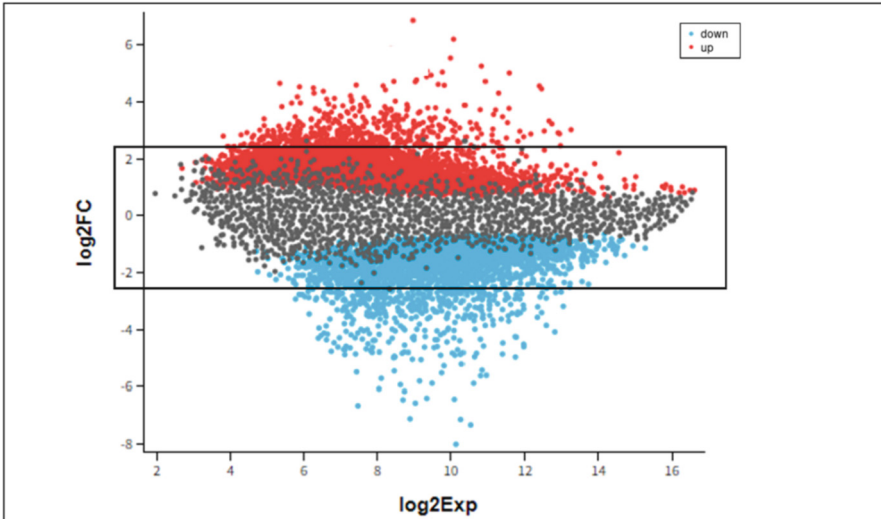


Fig. 1. Graphical overview of the study.

## 3 Results

### 3.1 Significant Differentially Expressed Genes (DEGs)

By using the FC method as a crucial parameter in order to select the differentially expressed genes (DEGs), we observed that thousands of genes (~2200 approx) were differentially expressed having  $\log_2FC$  scores ranging from  $-2.0$  to  $+2.0$  with  $p\text{-value} \leq 0.05$ . Most of the genes are up-regulated (shown in red) referring to their over-expression in serous ovarian adenocarcinoma when compared to down-regulated genes (shown in blue) (refer Fig. 2). A total of 167 significant differentially expressed genes that had a good  $\log_2FC$  and with smaller p-values were selected for further analyses. Figure 2 displays the scatter-plot of  $\log_2FC$  of all the genes.



**Fig. 2.** Scatter-plot of  $\log_2$  fold change (FC) values of all genes. Here, the red-colored dots represent up-regulated (over-expressed) genes while dots in blue represent down-regulated (under-expressed) genes in serous ovarian adenocarcinoma. The grey dots represent those genes that maintain a normal level of expression (Color figure online).

### 3.2 Disease-gene Associations (DGA) Study

Disease-gene associations enable the identification of potential genes that can act as prognostic indicators for a disease of interest. To get closer to the best biomarker candidates, we mapped the selected 167 DEGs for a disease-gene association using DisGeNET with query string = ‘(“ovarian serous adenocarcinoma”) AND biomarker AND altered expression’. After the execution of the search string, we retrieved a list of 144 significant genes that may have a potential role in ovarian cancer. The list of identified 144 genes with their statistical scores such as disease specificity index (DSI), disease pleiotropy index (DPI), disease-gene association score, and number of SNPs reported for the genes for the disease.

Out of these 144 genes, we could categorize the genes based on – biomarkers, altered expression, and genetic variations. To further narrow down, we focused only on those genes that were classified as biomarkers and had altered genetic expression. Therefore, only 88 genes popped out – 46 genes had evidence as biomarkers while 42 genes were reported to have altered expression. While screening these 88 genes, we noted the list shared common genes too. Therefore, from here we selected only 15 common seed genes that will further be studied for PCA and KM survival analyses. Table 1 displays the 15 seed genes and their association in ovarian serous adenocarcinoma identified after disease-gene association analysis.

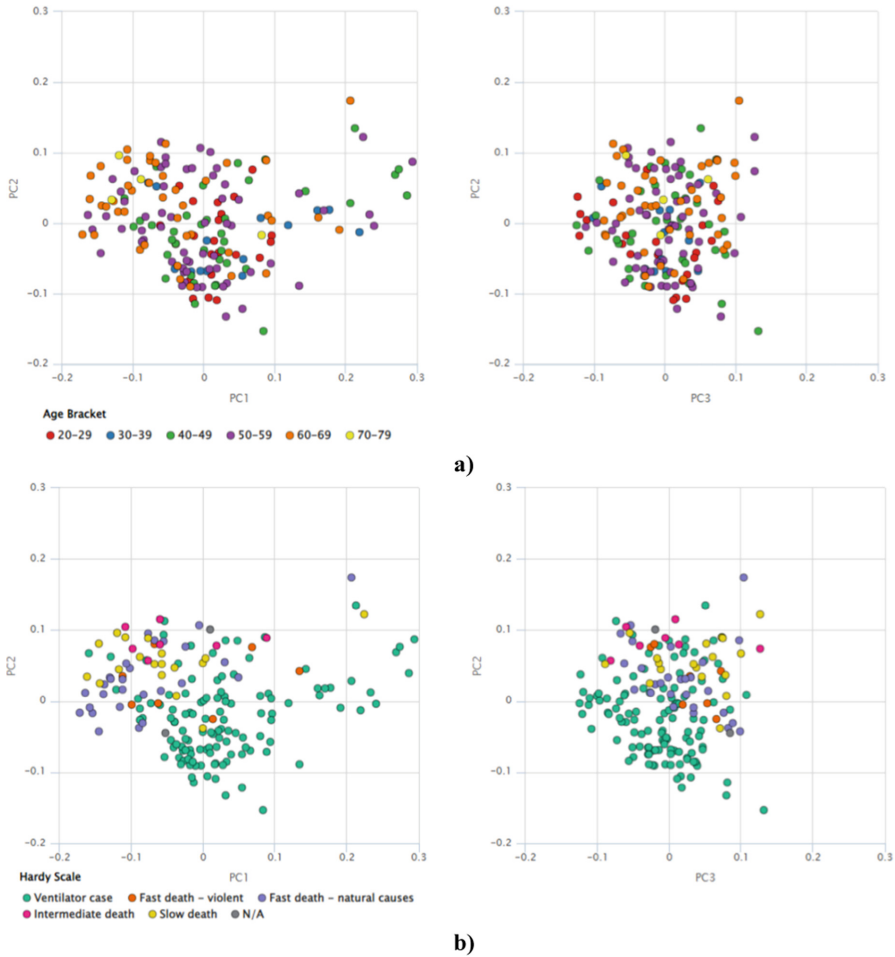
**Table 1.** 15 seed genes identified from disease-gene association.

S.No.	Seed gene	Association	Score
1	CLDN3	Biomarker	0.02
2	CLDN4	Biomarker	0.11
3	RSF1	Altered Expression	0.02
4	SCN1A	Biomarker; Altered Expression	0.3
5	NACC1	Biomarker, Altered Expression	0.01
6	PAX8	Biomarker	0.1
7	NFKB1	Biomarker	0.01
8	GSN	Biomarker, Altered Expression	0.1
9	EPHA1	Biomarker	0.02
10	MUC16	Biomarker	0.3
11	TP53	Biomarker	0.1
12	NANOG	Altered Expression	0.01
13	AURKB	Altered Expression	0.3
14	FKBP10	Altered Expression	0.01
15	CD274	Altered Expression	0.02

### 3.3 PCA and Kaplan-Meiers (KM) Survival Estimation

Out of 88 DEGs, we selected only 15 seed genes for further analysis. With these 15 seed genes in hand, we examined these for a principal component analysis (PCA) calculation and Kaplan-Meiers (KM) survival estimations. PCA is done to reduce the dimensionality of huge datasets thereby increasing the interpretability with minimal information loss. For this study, we plotted PCA graphs based on two parameters: a) age of the female, b) *Hardy-Weinberg's* equilibrium. Figure 3 depicts the principal component analysis (PCA) based on the above-mentioned parameters.

We observed that when age was kept as the main criteria for assessment, these seed genes mainly target females of age bracket 50–69 [1, 5]. The expression of these seed genes could be in older females because ovarian cancer gets detected in the advanced stages of the disease. It is mainly due to the lack of specific biomarkers for oncological malignancy. When the Hardy scale was placed as the main criteria for PCA analysis, we found that majority of the females that get diagnosed with ovarian cancer are screened late, therefore, admitted late for treatment [24].



**Fig. 3.** Representing principal component analysis. **a)** An incidence of ovarian serous adenocarcinoma based on the age bracket of a female. **b)** Severity of patients based on *Hardy-Weinberg's* equilibrium.

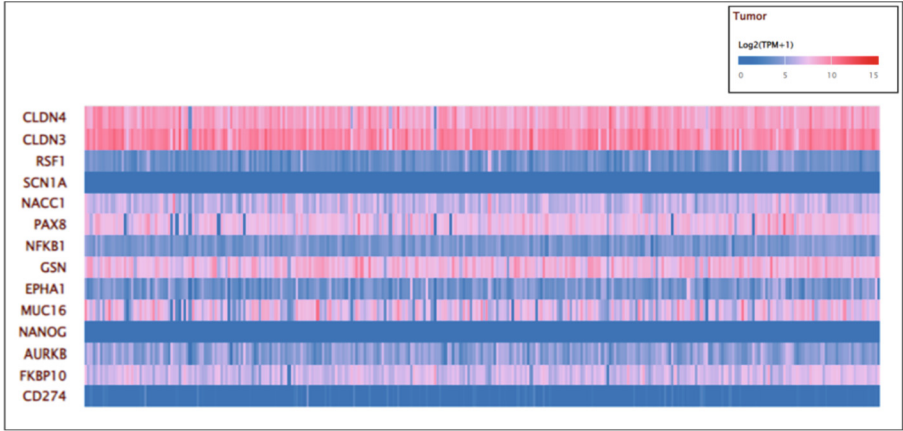
To check for genes that have a greater expression, we plotted an expression pattern heatmap aligning the expressions of each seed gene. The overall expression pattern of the seed genes is shown in Fig. 4. *CLDN3*, *CLDN4*, *PAX8*, *NAC1*, *GSN*, *MUC16*, and *FKBP10* showcased a greater over-expression when compared to the rest. This suggests that these genes are more likely to cause tumors that lead to the severity in a majority of ovarian cancer cases. The KM plots which are known as the *Kaplan-Meier curve* represent the probability of survival of a patient at a specific time interval. Table 2 represents the median survival estimates for each of the seed genes. The KM survival curves suggest that seed genes – *CLDN3*, *CLDN4*, *NFKB1*, *GSN*, *MUC16*, *NANOG*, *FKBP10*, and *CD274* have better survival medians in both low and high expression when



compared to the rest of the seed genes. Figure 5 represents the KM survival curves for these best 8 seed genes.

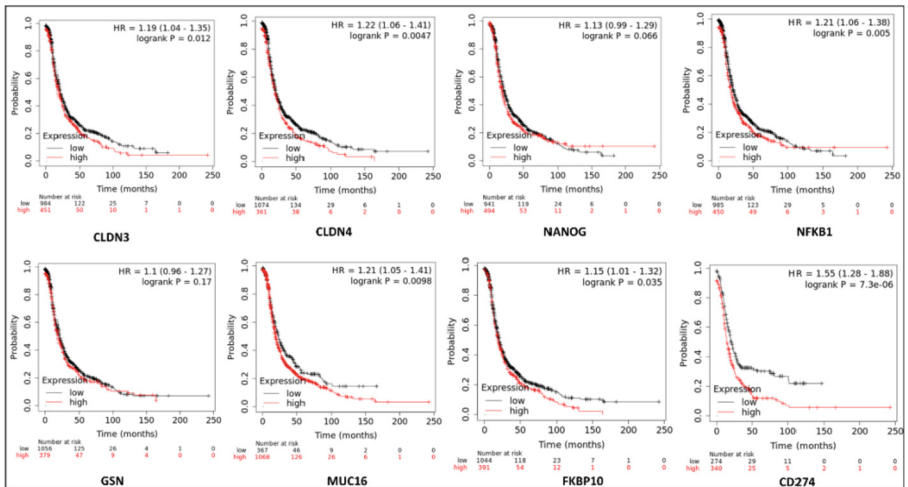
**Table 2.** Median survival estimates for all 15 seed genes.

S.No.	Seed gene	Description	Low expression cohort (months)	High expression cohort (months)
1	CLDN3	Claudin 3	20.63	18.83
2	CLDN4	Claudin 4	20.2	19.0
3	RSF1	Remodeling and spacing factor 1	18.0	15.0
4	SCN1A	Sodium voltage-gated channel alpha subunit 1	18.0	21.29
5	NACCI	Nucleus Accumbens Associated 1	18.27	16.0
6	PAX8	Paired box gene 8	18.79	22.5
7	NFKB1	Nuclear factor kappa B subunit 1	21.13	17.9
8	GSN	Gelsolin	20.53	18.43
9	EPHA1	Ephyrin type A receptor 1	18.23	22.02
10	MUC16	Mucin 16	23.24	19.0
11	TP53	Tumor protein P53	17.43	21.29
12	NANOG	Nanog Homeobox	21.13	18.2
13	AURKB	Aurora Kinase B	18.93	20.63
14	FKBP10	FKBP Prolyl Isomerase 10	20.47	19.0
15	CD274	CD274 molecule	20.0	14.37



**Fig. 4.** Heatmap showing expression patterns of seed genes in ovarian serous adenocarcinoma

Table 3 showcases the p-values and hazard ratio (HR) of the best 8 genes. These 8 genes were selected based on their P-values, HR scores, and KM survival curves and expression values.



**Fig. 5.** Kaplan Meier survival estimation for best 8 seed genes. a) KM survival curves. Here red color represents high expression while black represents the low expression of the genes (Color figure online).

**Table 3.** Hazard ratios and P-values of the 8 genes identified after KM analysis.

S.No.	Gene	Hazard Ratio (HR) value	p-value
1	CLDN3	1.19	0.012
2	CLDN4	1.22	0.0047
3	NFKB1	1.21	0.005
4	GSN	1.1	0.17
5	MUC16	1.21	0.0098
6	NANOG	1.13	0.066
7	FKBP10	1.15	0.035
8	CD274	1.55	7.3e-06

### 3.4 Gene Set Enrichment Analysis of the 8 Seed Genes

Gene ontology (GO) enrichment of the 8 seed genes suggest that they are involved in various biological processes, have different molecular functions and are localized in the major membrane systems. Table 4 depicts the GO enrichment of the seed genes based on engagement in biological processes, their molecular functions, cellular localization and pathways the seed genes are actively engaged in. As per our analysis, these seed genes are localized mainly in the membrane system – plasma membrane, endomembrane system, and organelle lumen. As far as the molecular functions of these 8 seed genes are concerned, they are all heavily deployed in binding and regulation of different processes. It was also known that these 8 seed genes are found crucial for various biological pathways too – mostly in gynecological cancers and signalling pathways.

### 3.5 Gene Regulatory Network Construction, Visualization and Topological Analysis

GeneMania plugin was used to construct gene regulatory network (GRN) with the 8 seed genes namely – CLDN3, CLDN4, NFKB1, GSN, MUC16, NANOG, FKBP10, and CD274 in cytoscape. We found that there were 20 direct interacting partners with the 8 seed genes where each association depicted a greater number of co-expression (96.12%), physical interactions (2.25%), sharing of protein domains was observed to be 1.55% with co-localization of 0.08%. We uploaded the retrieved network on OmicsNet webserver and selected to screen for possible genes, proteins, and microRNAs present in the subnetworks. To simplify the network, we used label propagation algorithm (LPA) [41] to visualize the GRN. After applying LPA to our network, we found only NFKB1, CD274, GSN, NANOG, FKBP10, CLDN4 were significant in the subnetwork that was obtained. These genes had a better degree along with higher betweenness between nodes (Table 5). We found that these 6 genes were interacting with various proteins, genes and microRNAs. These hub genes were further deployed to check for their roles in pathways from KEGG and GO using Consensus Pathway Analysis webserver. We found that these genes are known to be playing crucial roles in commencing different cancers, triggering

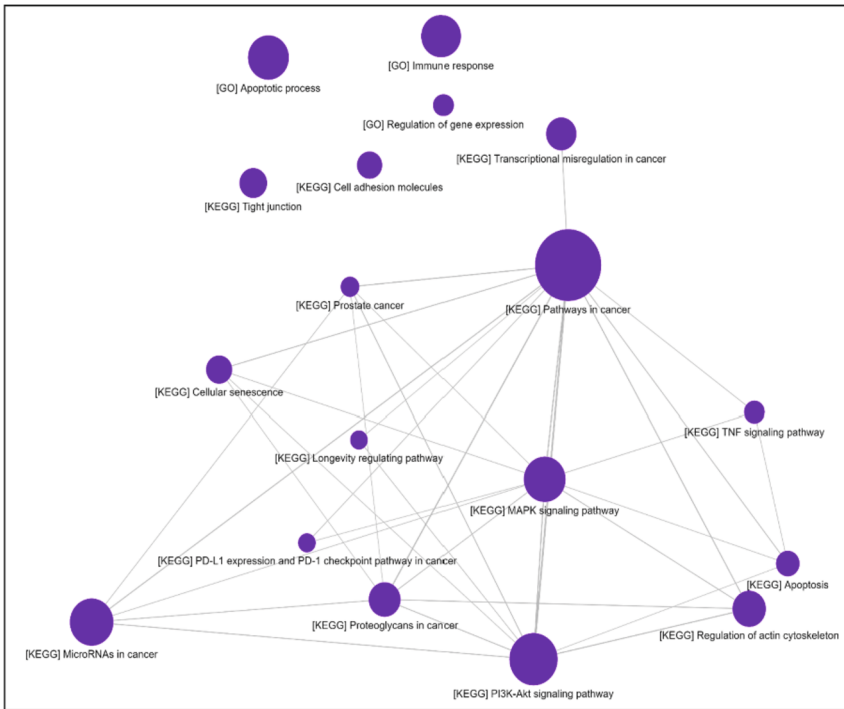
**Table 4.** Gene ontology enrichment of the 8 seed genes

GENE ONTOLOGY (GO)				
<i>Seed Gene</i>	<i>Biological Processes</i>	<i>Molecular Functions</i>	<i>Cellular Localization</i>	<i>Pathway Collection</i>
<b><i>CLDN3</i></b>	Response to stimulus, Wound healing, Cell adhesion, Macromolecular metabolic process, biological adhesion, Positive regulation of biological process, Positive regulation of metabolic process, response to cytokine, regulation of peptidase activity, Tight junction assembly, Positive regulation of cell motility	Binding, Protein Binding, DrugBinding, Chloride channel activity, cis-trans isomerase activity, DNA binding transcription repressor activity, Actinin binding	Membrane enclosed lumen, Apicolateral plasma membrane, Organelle lumen, Lateral plasma membrane, Tight junction, Anchoring junction, Endomembrane system, Bicellular tight junction, Cellular component	Proteins with altered expression in endometrial cancer, proteins involved in endometriosis, Coagulation, Apoptosis, Inflammatory response, TNF-Alpha signalling via NF-kB, Cancer immunotherapy by PD-1 blockade WP4585, Interactions between immune cells and microRNAs in tumor microenvironment WP4559
<b><i>CLDN4</i></b>				
<b><i>NFKB1</i></b>				
<b><i>GSN</i></b>				
<b><i>MUC16</i></b>				
<b><i>NANOG</i></b>				
<b><i>FKBP10</i></b>				
<b><i>CD274</i></b>				

signalling pathways such as – MAPK pathway, TNF pathway, PI3K-Akt pathway etc. They also known for causing apoptosis, regulation of actin cytoskeleton, cell adhesion, tight junction, immune responses etc. Figure 6 depicts the highly confident pathways and biological processes that are crucially played by the six hub genes CD274, NFKB1, NANOG, FKBP10, GSN, CLDN4. The greater the circle node, the greater the confidence score.

**Table 5.** Subnetwork topology

Gene Name	Degree	Betweenness	p-value
<i>CLDN4</i>	88	14703	3.36e-43
<i>GSN</i>	31	10264.75	2.28e-08
<i>CD274</i>	47	8770.75	8.89e-22
<i>NANOG</i>	28	7695.75	2.32e-13
<i>NFKB1</i>	17	4012.75	3.01e-09
<i>FKBP10</i>	5	842	8.94e-11



**Fig. 6.** Significant pathways where 6 hub genes – CD274, NFKB1, NANOG, FKBP10, GSN, CLDN4 are involved.

## 4 Discussion

Our analysis pipeline screened 15 common seed genes and found that these seed genes mainly target females of age bracket 50–69. The expression of these seed genes could be in older females because ovarian cancer gets detected in the advanced stages of the disease. It is mainly due to the lack of specific biomarkers for oncological malignancy [1, 5]. Furthermore, gene expression of most of seed genes tend to be on a milder

side referring to the fact that these seed genes result in poor RNA quality [26]. Genes namely – *CLDN3*, *CLDN4*, *PAX8*, *NAC1*, *GSN*, *MUC16* and *FKBP10* showcased a greater over-expression when compared to the rest. This suggests that these genes are more likely to cause tumors that lead to severity in majority of the ovarian cancer cases. The KM survival curves suggest that seed genes – *CLDN3*, *CLDN4*, *NFKB1*, *GSN*, *MUC16*, *NANOG*, *FKBP10* and *CD274* have better survival medians in both low and high expression when compared to the rest of the seed genes. These 8 genes were selected based on their P-values, HR scores, and KM survival curves and expression values.

We suggest these 8 genes – *CLDN3*, *CLDN4*, *NFKB1*, *GSN*, *MUC16*, *NANOG*, *FKBP10* and *CD274*, are highly significant and influential in dominating ovarian serous adenocarcinoma in females. After gene set enrichment analysis we found that the 8 seed genes are localized mainly in the membrane system – plasma membrane, endomembrane system, and organelle lumen. These seed genes are heavily deployed in binding and regulation of different processes (Table 4). These 8 seed genes are known to be crucial players in various disease and signalling processes. Some of these are cell adhesion, MAPK signalling pathway, transcriptional misregulation in cancer, apoptosis, etc. During network reconstruction analysis, we found that only six genes – *CLDN4*, *GSN*, *CD274*, *NANOG*, *FKBP10* and *NFKB1* showed a strong sub-network that were further associating with smaller proteins and microRNAs. We found that these genes are known to be playing crucial roles in commencing different cancers, triggering signalling pathways such as – MAPK pathway, TNF pathway, PI3K-Akt pathway etc. They also known for causing apoptosis, regulation of actin cytoskeleton, cell adhesion, tight junction, immune responses etc. All these pathways have been depicted based on greater P-value scores ( $> = 0.5$ ) and have been reported in KEGG and GO databases.

*MUC16* is simply CA-125 that is one of the recognized biomarkers for screening ovarian cancer. Studies reveal that *MUC16* regulates the innate immune response against ovarian cancer cells by directly stopping the Natural Killer (NK) cells to function [27]. *CLDN3* and *CLDN4* have been found to over-expressed in ovarian cancers [28]. *NFKB1* on the other hand has been proved to build an immune-evasive environment in ovarian cancer [29]. *GSN* has been discerned to be fruitful target for chemoresistant ovarian cancer, thus can be deployed as a screening marker for the disease in its initial stages [30]. *NANOG* has been found to monitor the stemness of cells (for instance -Cancer Stem Cells (CSCs) and thus can be used as a new target for screening ovarian cancer [31]. *FKBP10* has been proved to be under-expressed in high grade serous ovarian cancer [32]. *CD274* has been known as a tumor cell-intrinsic molecule that pushes MTORC1 signaling in mouse melanoma and mouse and human ovarian cancer preventing autophagy. Thus, *CD274* can be an indicator for autophagy in ovarian cancer [33].

## 5 Conclusion

We conclude that genes – *CLDN3*, *CLDN4*, *NFKB1*, *GSN*, *MUC16*, *NANOG*, *FKBP10* and *CD274* are highly significant and influential in dominating ovarian serous adenocarcinoma in females, and thus, can be looked as biomarkers for ovarian cancer initial screening examinations in patients.

**Acknowledgments.** SQ is supported by the DST-INSPIRE fellowship provided by the Department of Science & Technology, Govt. of India.

## References





1. Qazi, S., Sharma, A., Raza, K.: The role of epigenetic changes in ovarian cancer: a review. *Indian J. Gynecol. Oncol.* **19**(2) (2021)
2. Sung, H., et al.: Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: A Cancer Journal for Clinicians* **71**(3), 209–249 (2021)
3. Howlader, N., et al.: *Seer cancer statistics review: 1975 to 2014*. National Cancer Institute, Bethesda (2017)
4. Romero, I., Bast, R.C., Jr.: Minireview: human ovarian cancer: biology, current management, and paths to personalizing therapy. *Endocrinology* **153**(4), 1593–1602 (2012)
5. Jemal, A., et al.: Global cancer statistics. *CA Cancer J Clin.* **61**(2), 69–90 (2011)
6. Qazi, S., Raza, K.: In silico approach to understand epigenetics of POTEE in ovarian cancer. *J. Integr. Bioinform.* **18**(4) (2021)
7. Zhang, Y., Qazi, S., Raza, K.: Differential expression analysis in ovarian cancer: a functional genomics and systems biology approach. *Saudi J. Biologi. Sci.* **28**(7), 4069–4081 (2021)
8. Widschwendter, M., et al.: DNA hypomethylation and ovarian cancer biology. *Can. Res.* **64**(13), 4472–4480 (2004)
9. Zhang, W., et al.: Global DNA hypomethylation in epithelial ovarian cancer: passive demethylation and association with genomic instability. *Cancers* **12**(3), 764 (2020)
10. Singh, A., Gupta, S., Sachan, M.: Epigenetic biomarkers in the management of ovarian cancer: current perspectives. *Frontiers in Cell and Developmental Biology* **7** (2019)
11. Shen, Z., et al.: POTEE drives colorectal cancer development via regulating SPHK1/p65 signaling. *Cell Death & Disease* **10**(11), (2019)
12. Cine, N., et al.: Identification of ApoA1, HPX and POTEE genes by omic analysis in breast cancer. *Oncol. Rep.* **32**(3), 1078–1086 (2014)
13. Wang, Q., et al.: Serum levels of the cancer-testis antigen POTEE and its clinical significance in non-small-cell lung cancer. Coleman WB, ed. *PLOS ONE.* **10**(4), e0122792 (2015)
14. Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y., Morishima, K.: KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res.* **45**(D1), D353–D361 (2016)
15. The Cancer Genome Atlas program (TCGA): <https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga>. Accessed on 28 December 2021
16. Gene Expression Omnibus (GEO): <https://www.ncbi.nlm.nih.gov/geo/>. Accessed on 28 December 2021
17. Raza, K.: Clustering analysis of cancerous microarray data. *J. Chem. Pharm. Res.* **6**(9), 488–493 (2014)
18. RStudio: <https://www.rstudio.com/products/rstudio/download/>. Accessed on 15 December 2021
19. logFC: Calculate log-fold changes from hurdle model components. <https://www.rdocumentation.org/packages/PRIST/versions/0.925/topics/logFC>. Accessed on 15 December 2021
20. Piñero, J., et al.: DisGeNET: a comprehensive platform integrating information on human disease-associated genes and variants. *Nucleic Acids Res.* **45**(D1), D833–D839 (2016)
21. Principal components analysis online: <https://labriata.github.io/jsinscience/pca/index.html>. Accessed 3 January 2022
22. Kaplan-Meier Plotter (Ovarian cancer): <https://kmplot.com/analysis/> Accessed on 3 January 2022

23. Benjamini, Y., Hochberg, Y.: Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. Roy. Stat. Soc. Ser. B (Methodol.)* **57**(1), 289–300 (1995)
24. Tetsche, M.S., Dethlefsen, C., Pedersen, L., Sorensen, H.T., Norgaard, M.: The impact of comorbidity and stage on ovarian cancer mortality: a nationwide Danish cohort study. *BMC Cancer* **8**(1), (2008)
25. Crowe, C.: UALCAN: An integrated data-mining platform to facilitate the comprehensive analysis of cancer transcriptome - School of Medicine - Pathology | UAB. Published, Uab.edu (2017)
26. Bhuva, D.D., Cursons, J., Davis, M.J.: Stable gene expression for normalisation and single-sample scoring. *Nucleic Acids Res.* **48**(19), e113–e113 (2020)
27. Aithal, A., et al.: MUC16 as a novel target for cancer therapy. *Expert Opin. Ther. Targets* **22**(8), 675–686 (2018)
28. Honda, H., Pazin, M.J., D'Souza, T., Ji, H., Morin, P.J.: Regulation of the CLDN3 gene in ovarian cancer cells. *Cancer Biol. Ther.* **6**(11), 1733–1742 (2006)
29. Harrington, B.S., Annunziata, C.M.: NF- $\kappa$ B Signaling in Ovarian Cancer. *Cancers* **11**(8), 1182 (2019)
30. Asare-Werehene, M.: The role of plasma gelsolin in epithelial ovarian cancer chemoresistance. *Uottawaca* (2020)
31. Mahalaxmi, I., Devi, S.M., Kaavya, J., Arul, N., Balachandar, V., Santhy, K.S.: New insight into NANOG: a novel therapeutic target for ovarian cancer (OC). *Eur. J. Pharmacol.* **852**, 51–57 (2019)
32. Quin, M.C.J., et al.: FKBP10/FKBP65 expression in high-grade ovarian serous carcinoma and its association with patient outcome. *Int. J. Oncol.* **42**(3), 912–920 (2013)
33. Clark, C.A., Gupta, H.B., Curiel, T.J.: Tumor cell-intrinsic CD274/PD-L1: a novel metabolic balancing act with clinical potential. *Autophagy* **13**(5), 987–988 (2017)
34. Raudvere, U., et al.: G: Profiler: a web server for functional enrichment analysis and conversions of gene lists (2019 Update). *Nucleic Acids Research* **47**(W1), W191–W198 (2019)
35. Chen, E.Y., et al.: Enrichr: interactive and collaborative HTML5 gene list enrichment analysis tool. *BMC Bioinformatics* **14**(1), 128 (2013)
36. Franz, M., et al.: GeneMANIA update 2018. *Nucleic acids research* **46**(W1), W60–W64 (2018). <https://doi.org/10.1093/nar/gky311>
37. Zhou, G., Jianguo, X.: OmicsNet: a web-based tool for creation and visual analysis of biological networks in 3D space. *Nucleic Acids Research* **46**(W1), W514–W522 (2018)
38. Nguyen, H., et al.: CPA: a web-based platform for consensus pathway analysis and interactive visualization. *Nucleic Acids Research* **49**(W1), W114–24 (2021)
39. Kanehisa, M.: KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.* **28**(1), 27–30 (2000)
40. Ashburner, M., et al.: Gene ontology: tool for the unification of biology. *Nat. Genet.* **25**(1), 25–29 (2000)
41. Malhotra, D., Anuradha, C.: A modified label propagation algorithm for community detection in attributed networks. *Int. J. Info. Manage. Data Insights* **1**(2), 100030 (2021)





# GAGAM: A Genomic Annotation-Based Enrichment of scATAC-seq Data for Gene Activity Matrix

Lorenzo Martini<sup>(✉)</sup> , Roberta Bardini<sup>(✉)</sup> , Alessandro Savino<sup>(✉)</sup> ,  
and Stefano Di Carlo<sup>(✉)</sup> 

Control and Computer Engineering Department, Politecnico di Torino,  
10129 Turin, Italy

{lorenzo.martini,roberta.bardini,alessandro.savino,  
stefano.dicarlo}@polito.it

<https://www.smilies.polito.it>

**Abstract.** Single-cell Assay for Transposase Accessible Chromatin using sequencing (scATAC-seq) is rapidly becoming a powerful technology to assess the epigenetic landscape of thousands of cells. However, the current great sparsity of the resulting data poses significant challenges to their interpretability and informativeness. Different computational methods are available, proposing ways to generate significant features from accessibility data and process them to obtain meaningful results. In particular, the most common way to interpret the raw scATAC-seq data is through peak-calling, generating the peaks as features. Nevertheless, this method is dataset-dependent because the peaks are related to the given dataset and can not be directly compared between different experiments. For this reason, this study wants to improve on the concept of the Gene Activity Matrix (GAM), which links the accessibility data to the genes, by proposing a Genomic-Annotated Gene Activity Matrix (GAGAM), which aims to label the peaks and link them to the genes through functional annotation of the whole genome. Using genes as features solves the problem of the feature dataset dependency allowing for the link of gene accessibility and expression. The latter is crucial for gene regulation understanding and fundamental for the increasing impact of multi-omics data. Results confirm that our method performs better than the previous GAMs.

**Keywords:** Epigenomic single-cell data · Gene Activity Matrix · Bioinformatics

## 1 Introduction

Recent advances in New Generation Sequencing (NGS) technologies paved the way for single-cell multi-omics data analysis, which captures different facets of cells' regulative state, including the epigenome, the genome, the transcriptome,

and the proteome [19]. Multi-omics approaches increase resolution and sensitivity in the characterization of cellular states, the identification of known or new cellular phenotypes, and the understanding of cell dynamics [13]. This characteristic supports a quantitative and comprehensive approach to study cellular heterogeneity [5].

In particular, the combination of transcriptomic and epigenomic data provides integrated information on the functional activation of genes and the structural organization of chromatin. There are different experimental approaches to generate epigenomic data. These include accessibility measurements, which indicate whether chromatin is open or closed at genomic locations, exposing other genomic regions for transcriptional and regulatory processes [15]. These data have a very different organization than transcriptomic data indicating the expression level of genes.

Analyzing data from multiple omics does not directly imply to gain richer information on the cellular system, nor to gain a systemic understanding of regulative modalities generating the data. To achieve that, a multi-omics analysis must combine data-driven and model-driven approaches by considering not only the multiple modalities but also their interrelations in the cellular system [29]. To consider them together, it is necessary to correlate the expression level of genes (i.e., transcriptomic analysis) and the accessibility of their relevant coding and regulatory genomic regions.

The concept of gene activity, i.e., the overall accessibility of a gene allowing its transcription inside the cell [26], facilitates comparison between accessibility and expression data. Gene activity is a necessary but not sufficient condition to transcribe a gene: a cell can have a coding region accessible at the epigenomic level and the corresponding gene either strongly, weakly, or not expressed at all at the transcriptomic level. This must be considered when comparing transcriptomic and epigenomic data and build approaches to analyze them jointly.

A Gene Activity Matrix (GAM) [26] is an effective way to summarize accessibility information deriving from single-cell experiments. In a GAM, columns identify cells while rows identify genes. An element of the matrix ( $GAM_{g,c}$ ) represents the Gene Activity Scores (GAS) of the gene  $g$  in cell  $c$  [26]. The GAS is a value describing the activity of a gene in a cell in a given model. The use of the same genes in expression and activity experiments makes transcriptomic data directly comparable with epigenomic data.

Current approaches to compute GAMs derive primarily from data-driven strategies, which show limitations in capturing the contextual meaning and the regulative implications of epigenomic data. This work takes a step towards integrating transcriptomic and epigenomic data to support consistency in the joint consideration of gene activity and gene expression. In particular, this paper introduces a data- and model-driven computation of a Genomic Annotated GAM (GAGAM), which leverages accessibility data and information from genomic annotations of regulatory regions to weigh the gene activity with the annotated functional significance of accessible regulatory elements linked to the genes. GAGAM helps improve the resolution, explainability, and interpretability of the

results of the clustering and differential activity analyses, supporting the study of cellular heterogeneity based on epigenomic data alone [26].

## 2 Background

Single-cell Assay for Transposase Accessibility Chromatin sequencing (scATAC-seq) is rapidly becoming the primary way to assess the accessibility of the whole genome at the single-cell resolution. ScATAC-seq datasets employ different ways to define meaningful features to allow their analysis, as shown in [9]. One of the most popular is the “peak calling”, which defines peaks (i.e., intervals on the genome that have a local enrichment of transposase cut-sites) from an experiment-dependent set of chromosomal regions [36]. Since resulting peaks directly derive from the experimental results, they are not univocal, as in transcriptomic data. This hampers comparison of different analyses results and identification of cell-type related marker genes.

As described before, a GAM is an effective way to define robust accessibility features. The GAM considers the overall accessibility of the genomic regions linked to a gene. Using scATAC-seq data, the gene activity scores composing the elements of a GAM can be computed as the accessibility of the peaks related to a gene in a cell. However, the way to link peaks to the correct genetic region on the genome is not unique, and in the literature, there are three main strategies:

1. The GeneScoring sums the peaks in a broad region before and after a gene’s Transcription Starting Site (TSS), weighted by their distance from it [18]. This is the easiest way to define the activity of a gene, but it does not consider all the regulatory aspects.
2. Cicero defines the activity of a gene as the accessibility of the peaks overlapping the TSS and the accessibility of all the co-accessible peaks [26]. This method is more structured than the previous one. However, it identifies the genes through a single DNA base, i.e., the TSS, limiting the effectiveness of the approach. Moreover, co-accessibility evaluation is a very long and computationally heavy process, and the GAS estimation does not consider the meaningfulness of the peak.
3. Signac GAM counts all the raw reads in the gene body [28]. The main limitation of this method is the necessity of a fragment file related to the dataset, which contains all the fragments read in each cell. It is a large file and rarely available, thus making the computation often impossible.

In general, all these methods oversimplify the relationship between a gene and its accessibility. The epigenetic mechanisms are related to the regulation and resulting expression of the genes. However, this association is not direct and linear. If a gene is accessible, it is not necessarily also expressed: the association only gives an insight on whether transcription is possible or not.

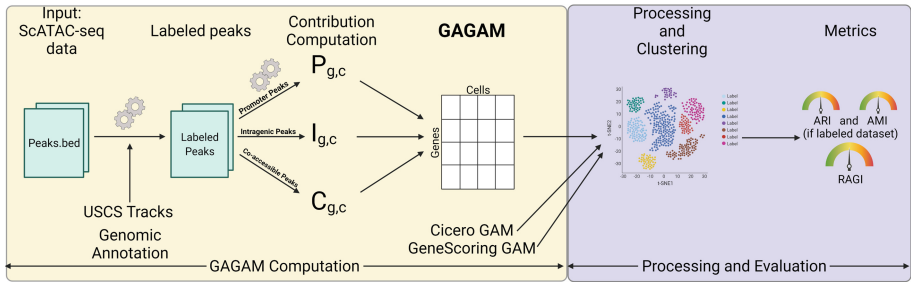
Studying how accessibility links to gene expression becomes relevant due to the emergence of new multi-omic Next-Generation Sequencing (NGS) techniques

allowing performing both scATAC-seq and Single-cell RNA sequencing (scRNA-seq) simultaneously. One way to achieve multi-omic consistent integration is to employ a model-driven approach.

For this reason, GAGAM introduces a new way to construct a GAM based on the functional annotation of the peaks. GAGAM is not only a new GAM but also a new way to interpret epigenomic data in perspective to link them with transcriptomic data. This method employs publicly available genomic annotations and evaluates the activity based on the regulatory elements linked to the gene by elaborating only the peaks related to the genes and their regulatory regions. Thus, it provides a model-driven GAS that reflects the accessibility to the whole transcription machinery, drawing a direct link to the gene expression.

### 3 Materials and Methods

Figure 1 introduces the workflow for computation and evaluation of GAGAM starting from a scATAC-seq dataset.



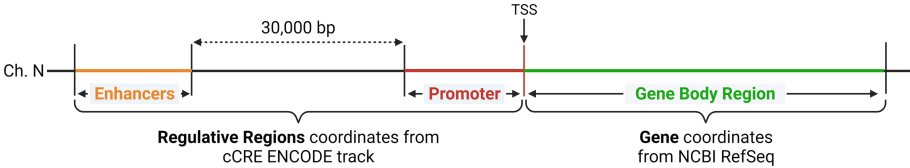
**Fig. 1. Workflow for computation and evaluation of GAGAM.** The workflow starts with scATAC-seq data, and labels the peaks with the help of genomic annotations and USCS tracks. Then it computes the three contributions forming GAGAM. GAGAM is evaluated and compared to other GAMs through clustering experiments with three well-established metrics: Adjust Rand Index (ARI) [17], and Adjust Mutual Information (AMI) [35] (if the dataset is labeled) or Residual Average Gini Index (RAGI) [6] (if the dataset is not labeled).

A scATAC-seq dataset contains a set  $P$  of peaks observed in a group of  $C$  cells. Each peak corresponds to a region of the target genome and is defined by its chromosome and a genomic coordinate pair  $p = (ch, start, stop)$ . The dataset is a binary matrix  $\mathbf{D}_{|P| \times |C|}$  where rows are associated with peaks and columns with cells. An element of  $\mathbf{D}$  equal to 1 denotes a peak (row) accessible in a cell (column).

The main contribution of GAGAM is to exploit information regarding overlaps of peaks, gene bodies, and genetic regulatory regions (i.e., promoters and enhancers) to build a GAM with higher information content.

### 3.1 Genomic Annotation

The genomic annotation of peaks is the first step to constructing GAGAM. This process aims to enrich information regarding peaks with data coming from different genomic annotations useful for a model-driven construction of a GAM. Figure 2 shows the genomic model considered in this work. It represents a genomic unit, which includes three parts: (i) the gene body region, starting at the Transcription Starting Site (TSS), (ii) the gene Promoter, preceding the coding region, and (iii) a set of Enhancers that are distal to the gene.



**Fig. 2. Genomic model.** The genomic model consists of the coordinates of all the genomic regions related to the gene. The gene body region (in green) comes from the NCBI RefSeq Genes annotations. The regulative regions, i.e., Promoter (in red), and Enhancers (in orange), come from the cCRE ENCODE tracks. (Color figure online)

The gene coding region is defined using NCBI RefSeq Genes [25] annotations, consisting of genes’ genomic coordinates. Therefore, a gene  $g$  in a target genome  $G$  is a tuple defining the gene’s chromosome and its genomic coordinates pair (i.e.,  $g = (ch, start, stop)$ ). The NCBI RefSeq annotations are accessible using the NCBI Eukaryotic Genome Annotation Pipeline [30]. It consists of an annotated and curated information list of protein-coding and non-protein-coding genes. The annotation also includes all the pseudogenes and miRNA regions. Since GAGAM aims to obtain something as close as possible to the transcriptomic information, it only considers the protein-coding and lncRNA regions.

The regulative genomic regions are elements on the DNA footprints for the trans-acting proteins involved in transcription, either for the positioning of the basic transcriptional machinery or for the regulation. The annotation tracks are associations between a genomic region and a label indicating the function of the region. Given a target genome  $G$  it is possible to define a set  $R$  of regulative regions with each region defined by the corresponding chromosome, the genomic coordinates pair, and a label (e.g., promoter or enhancer) indicating the function of the region ( $r = (ch, start, stop, l)$ ).

Information regarding regulative gene regions are available from the Encyclopedia of DNA Elements (ENCODE) project, which provides an extensive collection of cell- and tissue-based repertoires of genomic annotations, including, for example, transcription, chromatin organization, epigenetic landscape dynamics, and protein binding sites from the mouse and human genomes [24]. ENCODE data are available through the ENCODE data portal [11].

This work only considers genomic annotations relative to promoter and enhancer functions. These regulatory elements are derived from the ENCODE candidate cis-Regulatory Elements (cCREs). cCREs provides an extensive collection of annotated regions for the human and mouse genomes. Classification of cCREs is based on biochemical signatures, considering DNase hypersensitivity, histone methylation, acetylation, and CTCF binding data [24]. Since this work aims to label peaks from both human and mouse datasets, cCREs tracks (in BigBed format [16]) were collected from ENCODE for both the human [33], and mouse [34] genomes.

The goal of the genomic annotation process is to associate each peak  $p \in P$  obtained from a scATAC-seq dataset  $\mathbf{D}$  to a set of genomic annotation labels by analyzing how the peak overlaps to the different genomic regions.

GAGAM labels each peak  $p \in P$  with four possible labels: (1) **prom** for peaks overlapping a promoter region, (2) **enhD** for peaks overlapping a distal enhancer region and not a promoter region, (3) **intra** for the peaks contained into a gene body region, and (4) **empty** in all other cases. The rule to assign the label is summarized in the following equation:

$$\mathcal{P}\mathcal{L} : p \in P \mapsto \begin{cases} \text{prom} & \text{if } \exists r \in R | r \subseteq p \wedge r_l = \text{prom} \\ \text{enhD} & \text{if } (\exists r \in R | r \subseteq p \wedge r_l = \text{enh}) \wedge \\ & (\nexists r \in R | r \subseteq p \wedge r_l = \text{prom}) \\ \text{intra} & \text{if } \exists g \in G | p \subseteq g \\ \text{empty} & \text{otherwise} \end{cases} \quad (1)$$

The operator  $a \subseteq b$  is used here to denote that the two regions  $a$  and  $b$  belong to the same chromosome with  $b$  overlapping  $a$  (i.e.,  $a_{start} \geq b_{start} \wedge a_{end} \leq b_{end}$ ). The computation of the intersection between peaks and annotation regions leverages the **bigBedToBed** tool from ENCODE [32].

Performing genome annotation for the mouse genome is straightforward for all considered datasets since both datasets, and annotation tracks refer to the mm10 genomic assembly. Differently, for the human genome, the cCREs annotation track is only available for the hg38 genomic assembly, while the human dataset is based on the hg37 genomic assembly. For this reason, this work leverages the UCSC **LiftOver** tool [20] to convert the peaks' coordinate ranges from the hg37 to the hg38 assemblies before performing peak labeling.

Given the list of annotated peaks, GAGAM builds a gene activity matrix as a weighted sum of three separated matrices: (i) the promoter peaks matrix (**P**) indicating accessibility of genes associated with promoter peaks, (ii) the intragenic peaks matrix (**I**) indicating the accessibility of genes containing intragenic peaks, and (iii) the co-accessibility matrix (**C**) indicating the accessibility of genes associated with distal enhancer peaks, obtaining a final curated and model-driven evaluation of the activity of the genes:

$$\mathbf{GAGAM} = w_p \cdot \mathbf{P} + w_i \cdot \mathbf{I} + w_c \cdot \mathbf{C} \quad (2)$$

### 3.2 Promoter Peaks Matrix

The promoter peaks matrix exploits model-driven information about promoter peaks to identify relevant genes in the GAM. The golden rule applied in GAGAM is that *a gene in a cell is active if, and only if, its promoter peak is accessible*. This rule reduces the set of interesting genes to consider when constructing a GAM.

To follow this rule, let us denote with  $P^p \subseteq P$  the subset of peaks in the dataset  $\mathbf{D}$  annotated as promoters (i.e.,  $\mathcal{PL}(p) = \mathbf{prom} \forall p \in P^p$ ) and with  $\mathbf{D}_{|P^p| \times |C|}^p$  the submatrix of  $\mathbf{D}$  including only rows associated to promoter peaks.

GAGAM constructs a binary matrix  $\mathbf{GP}_{|G^p| \times |P^p|}$  associating the set of genes with active promoter peaks ( $G^p$ ) to their related peaks. To associate a promoter peak to a gene, GAGAM considers the overlapping of an enlarged gene body region including 500 bp before the TSS (i.e., an approximation of the mean peak length) with the peak region. Based on this, the promoter peaks matrix is a binary matrix computed as:

$$\mathbf{P}_{|G^p| \times |C|} = \mathbf{GP}_{|G^p| \times |P^p|} \times \mathbf{D}_{|P^p| \times |C|}^p \quad (3)$$

This matrix is a GAM including accessibility data for the subset of genes associated with the promoter peaks. In this way, GAGAM leverages available knowledge on transcriptional regulatory regions to define the active genes based on a model taking into account the knowledge of gene regulation and transcription.

### 3.3 Intragenic Peaks Matrix

GAGAM also considers the contribution of the intragenic peaks (i.e., peaks located in the gene body region) to the overall gene activity score. Similarly to what described before, let us denote with  $P^i \subseteq P$  the subset of peaks in the dataset  $\mathbf{D}$  annotated as intragenic (i.e.,  $\mathcal{PL}(p) = \mathbf{intra} \forall p \in P^i$ ) and with  $\mathbf{D}_{|P^i| \times |C|}^i$  the submatrix of  $\mathbf{D}$  including only rows associated to intragenic peaks.

GAGAM constructs a matrix  $\mathbf{GI}_{|G^p| \times |P^i|}$  associating genes with active promoter peaks ( $G^p$ ) to their related intragenic peaks. This matrix only considers genes with active promoter peaks to follow the GAGAM golden rule (Sect. 3.2). Some of the identified intragenic peaks could be part of genes that do not have a promoter peak. Moreover, it could happen that given a gene region inside a cell, intragenic peaks could be accessible even if the promoter peak is not.

Statistically, there will be more peaks inside the gene body region of a long gene, meaning it might have a higher score after its length. To prevent this bias, GAGAM employs a strategy from the GeneScoring [18] method to compute the elements of  $\mathbf{GI}$ . It weighs the contribution of the intergenic peaks with an exponentially decaying function of their distance from the TSS (i.e.,  $\mathbf{GI}_{g,p} = a \cdot e^{-\frac{d}{5000}}$  where  $a = 1$  if  $p \subseteq g$ , 0 otherwise and  $d$  is the distance of the peak from TSS). In this way, very long genes are not over-represented because the most crucial part of the gene's activity is near the promoter. Therefore, the

peaks near it are weighted more. Based on this, the intragenic peaks matrix is a matrix computed as:

$$\mathbf{I}_{|G^p| \times |C|} = \mathbf{GI}_{|G^p| \times |P^i|} \times \mathbf{D}_{|P^i| \times |C|}^i \quad (4)$$

### 3.4 Promoter-Enhancer Co-accessibility Matrix

The co-accessibility matrix accounts for the connections between promoters and enhancers. It leverages Cicero [26] to calculate the co-accessibility of the peaks. The co-accessibility represents how couples of peaks tend to be simultaneously accessible in the cells, expressing it in a range between 0 and 1. This calculation “connects regulatory elements to their putative target genes” [26], meaning it can find connections between the promoters and the distal regulatory regions where different elements like Transcriptional Factors (TF) bind and enable the transcription.

The first step to compute this matrix is to calculate the co-accessibility from the scATAC-seq data with the Cicero function `run_cicero` (for the explanation of the calculation, refer to [26]). The result is a list of peaks couples with their co-accessibility value ( $ca$ ) and distance ( $d$ ) in the form  $conn = (p^1, p^2, ca, d)$ .

GAGAM selects only couples of promoter-enhancer peaks, i.e., couples with  $p^1 \in P^p$  and  $p^2 \in P^e$  (or vice versa), with  $P^e \subseteq P$  representing the subset of peaks in the dataset  $\mathbf{D}$  annotated as enhancers (i.e.,  $\mathcal{PL}(p) = \text{enhD} \forall p \in P^e$ ).

Moreover, GAGAM keeps only couples with  $ca \geq ca_m$  (with  $ca_m$  the mean value of all the co-accessibility scores above zero) and  $d \leq d_{th}$  (with  $d_{th} = 30,000$  bp the distance threshold defined as suggested by the guidelines of Cicero [26]).

To calculate the co-accessibility matrix  $\mathbf{C}$ , GAGAM uses three matrices. First, the binary matrix  $\mathbf{GP}_{|G^p| \times |P^p|}$  previously defined in Sect. 3.2 and associating genes with promoter peaks. Second, the matrix  $\mathbf{PE}_{|P^p| \times |P^e|}$  associating promoter peaks and enhancer peaks. The elements of this matrix are the co-accessibility values  $ca$  of the couples of peaks available in the list produced by Cicero, and 0 otherwise. Third, the matrix  $\mathbf{D}_{|P^e| \times |C|}^e$  is a submatrix of  $\mathbf{D}$  including only rows associated with enhancer peaks.

Based on this, the co-accessibility matrix is computed as:

$$\mathbf{C}_{|G^p| \times |C|} = \mathbf{GP}_{|G^p| \times |P^p|} \times \mathbf{PE}_{|P^p| \times |P^e|} \times \mathbf{D}_{|P^e| \times |C|}^e \quad (5)$$

## 4 Results and Discussion

### 4.1 Evaluation Strategy

This section evaluates GAGAM by looking at different aspects.

First, GAGAM represents an interpretation of scATAC-seq data. As the majority of single-cell experiments, it must identify cellular heterogeneity. Based on this consideration, the first approach to evaluate the capabilities of this new



gene activity matrix is to employ one of the many available pipelines to process GAMs (Fig. 1). This work uses Monocle3 [31], given its simplicity and the fact that Cicero GAM (see Sect. 2) is dependent on it.

The standard Monocle workflow starts with a GAM, performs Principal Component Analysis (PCA), visualizes the cells in 2D using UMAP [23], and most importantly, performs cells clustering. The clustering results should at least partially represent the cellular heterogeneity of the dataset. This can be measured using a group of metrics thoroughly discussed in Sect. 4.2.

Moreover, it can be proved that GAGAM and, in particular, the selected genes are not just a product of data manipulation but are biologically meaningful in two ways. First, performing differential activity analysis on the GAM can show that the differentially active genes are cell-type specific. This would also demonstrate that employing marker genes allows classifying scATAC datasets, something not possible with raw data. Second, using the RAGI index (one of the metrics for the evaluation of the clustering performances defined in Sect. 4.2), it is possible to assess the informativity of the GAGAM.

## 4.2 Metrics Definition

The evaluation strategy proposed in Sect. 4.1 is based on unsupervised clustering of cells based on the selected genes. The obtained clusters are the outputs that must be analyzed to understand if they represent cell heterogeneity. There are two scenarios: (i) the starting dataset has cell labels; thus, each cell has a label identifying its cell type, or (ii) there are no available cell labels, so there is no ground truth to compare.

In the first case, the most direct way to measure the quality of the clustering process is to compare the clusters to the cell-type labels. To show how much the two classifications are similar, this paper uses the Adjust Rand Index (ARI) [17] and Adjust Mutual Information (AMI) [35] from information theory. These two metrics are often employed for this type of evaluation. In particular, [6] uses them for their benchmarking. Thus, they help compare their results with those produced in this paper. ARI and AMI range between 0 and 1, where 1 is a perfect match, and 0 is complete uncorrelation. This evaluation employs the R package ARICODE [8], which easily allows their calculation.

The second case requires a different approach. Since there is no reference classification, ARI and AMI cannot be used. One method is to calculate ARI and AMI comparing the results with the clustering-based labels obtained from the scATAC-data data processing. Otherwise, [6] proposes a very fitting way: the Residual Average Gini Index (RAGI) [6]. The RAGI investigates the differences in the Gini index of markers and housekeeping genes. The idea is that a good clustering should have marker genes active only in specific clusters and housekeeping genes over all the cells. Therefore, RAGI can measure the quality of the GAM itself. A good GAM should convey meaningful biological information that should translate into a difference between the two sets of genes. Therefore, the RAGI estimates if a GAM can correctly assess the gene activity. Anyway, RAGI has the problem of being highly dependent on the employed genes to calculate

it. Still, the concept of housekeeping genes and, even more, marker genes are not well-defined [22]. Therefore, it is essential to carefully choose the right set of marker genes strictly related to the dataset sample. In this work, the list of housekeeping genes derives from [10] for humans and [12] for the mouse. On the other hand, the marker genes list comes from the CellMarker [37] database, which provides a curated list of markers per tissue. For the mouse brain datasets, this work also employs markers from [14], and [21].

### 4.3 Datasets

GAGAM was tested on five datasets (see Table 1). Two datasets are from the 10XGenomic platform [27], and consist of a collection of respectively 5,335 (*10X V1.0.1 PBMC* [1]) and 4,623 (*10X V2.0.0 PBMC* [3]) cells from human Peripheral Blood Mononuclear Cells (PBMC) samples. From the 10XGenomic platform, there is also a mouse brain dataset with 5,337 cells (*10X V1.1.0 Brain* [2]). All three datasets do not have cell labels. Therefore, ARI and AMI evaluations are applicable only on the clustering-based labels. Next, this study employed a dataset of bone marrow (*Buenrostro2018*) from [4]. This dataset consists of 2,034 cells and provides cell-type classification. The last dataset comes from a multi-omic SNARE experiment (*SNARE* [7]). It consists of 10,309 cells from the mouse cortex and comes with a partial classification of the cells. Two of the considered datasets (*10X V1.0.1 PBMC* and *Buenrostro2018*) derive from [6], a paper performing a benchmarking analysis on different methods allowing for easy comparison of results.

**Table 1.** Datasets employed

Dataset	Species	Tissue	Cells labels	Reference
<i>10X V1.0.1 PBMC</i>	Human	PBMC	No	[1]
<i>10X V2.0.0 PBMC</i>	Human	PBMC	No	[3]
<i>10X V1.1.0 Brain</i>	Mouse	Brain cortex	No	[2]
<i>Buenrostro2018</i>	Human	Bone marrow	Yes	[4]
<i>SNARE</i>	Mouse	Brain cortex	Yes	[7]

### 4.4 Results

This section compares the performance of GAGAM with two state-of-the-art GAM computation pipelines (i.e., Cicero and GeneScoring) following the evaluation strategy proposed in Sect. 4.1. Since GAGAM is constructed from three contributions (see Eq. 2), it is advisable to evaluate different combinations to select the best one. This experimental setup considers two versions of GAGAM: GAGAM1 constructed considering only the promoter peaks and the co-accessibility (i.e.,  $w_p = 1$ ,  $w_i = 0$ , and  $w_c = 1$ ) and GAGAM2 created using the complete GAGAM workflow (i.e.,  $w_p = 1$ ,  $w_i = 1$ , and  $w_c = 1$ ).

Figures 3 and 4 reports AMI and ARI results comparing *Buenrostro2018* [4] and *SNARE* [7] clusters, with their ground truth labels, and the other datasets against the scATAC clustering results. Overall, Figs. 3 and 4 shows that both versions of GAGAM perform equally or better than Cicero and GeneScoring. Only on *10X V1.1.0 Brain*, GeneScoring has a higher metric value than GAGAM. However, comparing the results in [4] with the ones reported in [6] on the same dataset shows how the GAGAM performances are on the high end of the benchmarked paper methods (as shown in Table 5 from [6]).

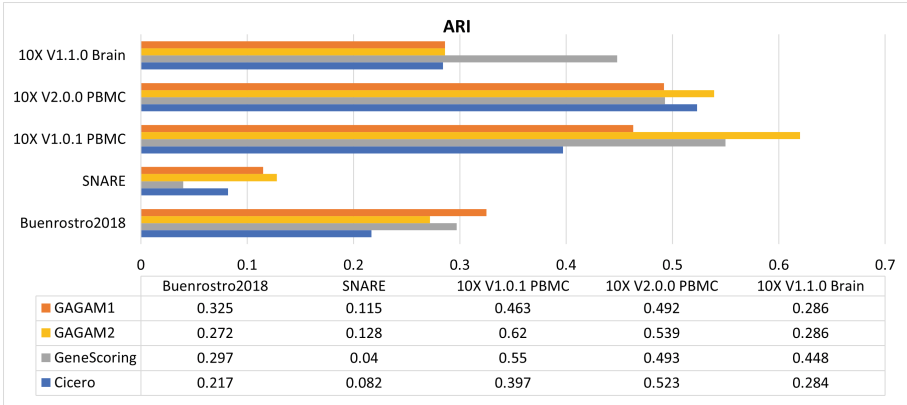


Fig. 3. ARI results of the four methods for the five datasets

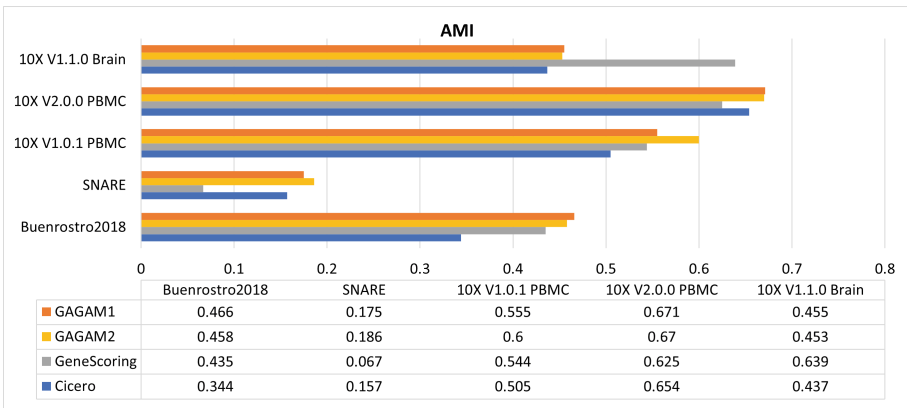
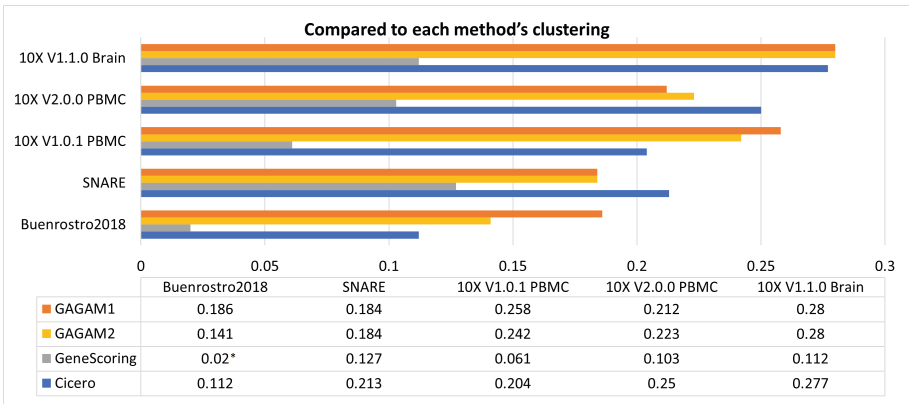


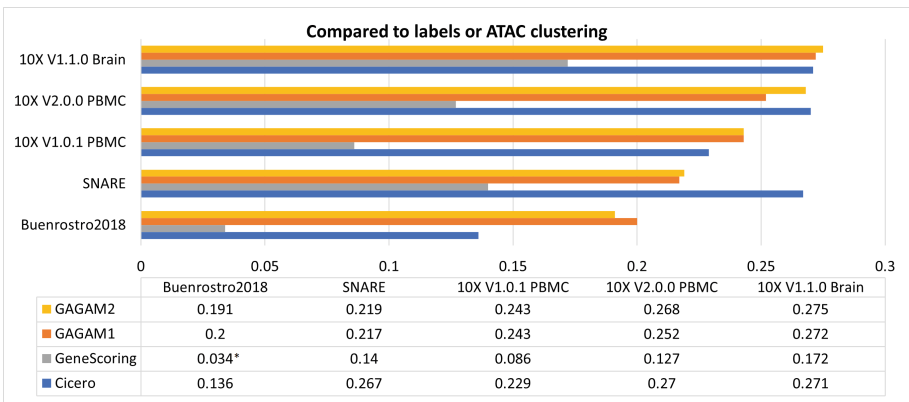
Fig. 4. AMI results of the four methods for the five datasets

Next, RAGI has been computed for all datasets to assess the clustering results and the information content of the GAMS. The results are in Figs. 5 and 6. For

the three different types of tissues, we employed three different sets of curated markers, while the housekeeping genes were shared between the same species datasets. For each method and dataset, there are two different results. One is the RAGI score calculated on each GAM concerning the clustering results. The other is computed on each GAM but resorting to the cell labels (when available) or the clustering-based labels obtained from the scATAC data processing. This way, all methods are evaluated against the same partition to understand which GAM is the most biologically consistent. In particular, the *10X V1.0.1 PBMC* dataset is assessed with this metric in [6], and GAGAM outperforms all the methods illustrated there.



**Fig. 5.** RAGI results of the four methods for the five datasets, when compared to each method’s clustering



**Fig. 6.** RAGI results of the four methods for the five datasets, when compared to labels or ATAC clustering

In general, the results show how GAGAM has consistently good performances. Nevertheless, in this case, Cicero performs better on the *SNARE* dataset. Instead, GeneScoring offers low performances. Although its clustering results are consistent with the ground-truth classification (as indicated by ARI and AMI), the actual scores are not well defined. This suggests the importance of evaluating the GAMs on both metrics. Therefore, although there are some cases where Cicero and GeneScoring have better results than GAGAM, the latter has a consistent behavior on all the metrics, meaning it is the most reliable method on both clustering results and actual GAS computation. It is essential to highlight that some of the RAGI results (marked with \*) have a p-value over the tolerable threshold (0.05), so they are not statistically meaningful, but we report them anyways.

## 5 Conclusions

In conclusion, GAGAM is a new method to obtain a Gene Activity Matrix from scATAC-seq data. It is based on a model-driven approach leveraging genomic annotations of genes and functional elements. It introduces the promoter peak accessibility into the score, which is necessary for the gene's activity. Then, it considers the contribution of intragenic peaks, weighted by their distance from the TSS and the enhancer peaks connected to the promoter. The score obtained this way represents a good model of the gene activity interpreted as the set of elements that should be accessible to allow gene transcription.

Experimental results demonstrate how GAGAM generally performs better against other GAMs concerning its ability to identify cellular heterogeneity. Specifically, the clustering obtained from GAGAM is evaluated with ARI, AMI, and RAGI and has better results than Cicero and GeneScoring on all of these metrics. In addition, GAGAM is a suitable method to interpret accessibility data in general. Indeed, since it employs genes as features, it allows analyzing scATAC-seq data through well-studied and investigated concepts like marker genes. The same analysis would not be possible with raw accessibility data. RAGI results support this claim and highlight the activity differences between marker and housekeeping genes. This activity proves that the features selected in GAGAM (i.e., the genes) and their activity scores are biologically meaningful. Therefore, GAGAM provides an optimal and reliable middle ground between the accessibility data and the gene expression data, crucial for future works in a field where multi-omics single-cell techniques are fastly growing.

In conclusion, GAGAM is a promising and reliable way to interpret scATAC-seq data, which focuses on the accessibility of the genes and their regulatory elements, acting as a direct link between epigenomic and transcriptomic.

## References




1. 10XGenomics: 5k peripheral blood mononuclear cells (PBMCs) from a healthy donor single cell ATAC dataset by cell ranger ATAC 1.0.1, 10x genomics, 17 December 2019

2. 10XGenomics: fresh cortex from adult mouse brain (p50) single cell ATAC dataset by cell ranger ATAC 1.1.0, 10x genomics, 16 April 2019
3. 10XGenomics: peripheral blood mononuclear cells (PBMCs) from a healthy donor single cell ATAC dataset by cell ranger ATAC 2.0.0, 10x genomics, 3 May 2021
4. Buenrostro, J.D., et al.: Integrated single-cell analysis maps the continuous regulatory landscape of human hematopoietic differentiation. *Cell* **173**(6), 1535–1548.e16 (2018)
5. Carter, B., Zhao, K.: The epigenetic basis of cellular heterogeneity. *Nat. Rev. Genet.* **22**(4), 235–250 (2021)
6. Chen, H., et al.: Assessment of computational methods for the analysis of single-cell ATAC-seq data. *Genome Biol.* **20**(1) (2019). Article number: 241. <https://doi.org/10.1186/s13059-019-1854-5>
7. Chen, S., Lake, B.B., Zhang, K.: High-throughput sequencing of the transcriptome and chromatin accessibility in the same cell. *Nat. Biotechnol.* **37**, 1452–1457 (2019)
8. Chiquet, J.: aricode: efficient computations of standard clustering comparison measures. <https://cran.r-project.org/web/packages/aricode/index.html>
9. Danese, A., Richter, M.L., Chaichoompu, K., et al.: EpiScanpy: integrated single-cell epigenomic analysis. *Nat. Commun.* **12**(D1), 5228 (2021)
10. Eisenberg, E., Levanon, E.Y.: Human housekeeping genes, revisited. *Trends Genet. (TIG)* **29**(10), 569–574 (2013)
11. ENCODE: encode data portal. <https://www.encodeproject.org>
12. Hounkpe, B.W., et al.: HRT Atlas v1.0 database: redefining human and mouse housekeeping genes and candidate reference transcripts by mining massive RNA-seq datasets. *Nucleic Acids Res.* **49**(D1), D947–D955 (2021)
13. Hu, Y., An, Q., Sheu, K., Trejo, B., Fan, S., Guo, Y.: Single cell multi-omics technology: methodology and application. *Front. Cell Dev. Biol.* **6**, 28 (2018)
14. Allen Institute: 2010 Allen cell types database. <https://portal.brain-map.org/atlases-and-data/rnaseq>
15. Kelsey, G., Stegle, O., Reik, W.: Single-cell epigenomics: recording the past and predicting the future. *Science* **358**(6359), 69–75 (2017)
16. Kent, W.J., Zweig, A.S., Barber, G., Hinrichs, A.S., Karolchik, D.: BigWig and BigBed: enabling browsing of large distributed datasets. *Bioinformatics* **26**(17), 2204–2207 (2010)
17. Hubert, L., Arabie, P.: Comparing partitions. *J. Classif.* **2**, 193–218 (1985). <https://doi.org/10.1007/BF01908075>
18. Lareau, C.A., Duarte, F.M., Chew, C.G., et al.: Droplet-based combinatorial indexing for massive-scale single-cell chromatin accessibility. *Nat. Biotechnol.* **37**, 916–924 (2019)
19. Li, Y., Ma, L., Wu, D., Chen, G.: Advances in bulk and single-cell multi-omics approaches for systems biology and precision medicine. *Briefings Bioinform.* **22**(5), bbab024 (2021)
20. Luu, P.L., Ong, P.T., Dinh, T.P., Clark, S.J.: Benchmark study comparing liftover tools for genome conversion of epigenome sequencing data. *NAR Genomics Bioinform.* **2**(3), lqaa054 (2020)
21. Martini, L.: Study of cellular heterogeneity of mouse cerebral cortex, through joint scRNA-seq and scATAC-seq analysis, derived from SNARE-seq technique (2020)
22. Martini, L., Bardini, R., Di Carlo, S.: Meta-analysis of cortical inhibitory interneurons markers landscape and their performances in scRNA-seq studies. In: 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 253–258 (2021), <https://doi.org/10.1109/BIBM52615.2021.9669888>

23. McInnes, L., Healy, J., Saul, N., Großberger, L.: UMAP: uniform manifold approximation and projection. *J. Open Source Softw.* **3**(29), 861 (2018)
24. Moore, J.E., et al.: Expanded encyclopaedias of DNA elements in the human and mouse genomes. *Nature* **583**(7818), 699–710 (2020)
25. O’Leary, N.A., et al.: Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. *Nucleic Acids Res.* **44**, D733–D745 (2016)
26. Pliner, H.A., et al.: Cicero predicts cis-regulatory DNA interactions from single-cell chromatin accessibility data. *Mol. Cell* **71**, 1–14 (2018)
27. Satpathy, A.T., Granja, J.M., Yost, K.E., et al.: Massively parallel single-cell chromatin landscapes of human immune cell development and intratumoral T cell exhaustion. *Nat. Biotechnol.* **37**, 925–936 (2019)
28. Stuart, T., Satija, R., et al.: Single-cell chromatin state analysis with Signac. *Nat. Methods* **18**(11), 1333–1341 (2021)
29. Subramanian, I., Verma, S., Kumar, S., Jere, A., Anamika, K.: Multi-omics data integration, interpretation, and its application. *Bioinform. Biol. Insights* **14** (2020). <https://doi.org/10.1177/1177932219899051>
30. Thibaud-Nissen, F., Souvorov, A., Murphy, T., et al.: Eukaryotic genome annotation pipeline. In: *The NCBI Handbook* [Internet]. 2nd edition. Bethesda (MD): National Center for Biotechnology Information (US) (2013)
31. Trapnell, C., Cacchiarelli, D., Grimsby, J., et al.: The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nat. Biotechnol.* **32**, 381–386 (2014)
32. USCS: bigbedtobed too. [genome.ucsc.edu/goldenPath/help/bigBed.html](http://genome.ucsc.edu/goldenPath/help/bigBed.html)
33. USCS: USCS human CCRE track download. [hgdownload.soe.ucsc.edu/gbdb/hg38/encode3/ccre/encodeCcreCombined.bb](http://hgdownload.soe.ucsc.edu/gbdb/hg38/encode3/ccre/encodeCcreCombined.bb)
34. USCS: USCS mouse CCRE track download. [hgdownload.soe.ucsc.edu/gbdb/mm10/encode3/ccre/encodeCcreCombined.bb](http://hgdownload.soe.ucsc.edu/gbdb/mm10/encode3/ccre/encodeCcreCombined.bb)
35. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: is a correction for chance necessary? In: *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009*, pp. 1073–1080. Association for Computing Machinery, New York (2009). <https://doi.org/10.1145/1553374.1553511>
36. Yan, F., et al.: From reads to insight: a hitchhiker’s guide to ATAC-seq data analysis. *Genome Biol.* **21**(1), 1–16 (2020)
37. Zhang, X., et al.: CellMarker: a manually curated resource of cell markers in human and mouse. *Nucleic Acids Res.* **47**(D1), D721–D728 (2019)



# Finding Significantly Enriched Cells in Single-Cell RNA Sequencing by Single-Sample Approaches

Anna Mrukwa<sup>1</sup>, Michal Marczyk<sup>1,2</sup> , and Joanna Zyla<sup>1</sup>  

<sup>1</sup> Department of Data Science and Engineering, Silesian University of Technology,  
Akademicka 16, 44-100 Gliwice, Poland

annamru993@student.polsl.pl, {michal.marczyk,  
joanna.zyla}@polsl.pl

<sup>2</sup> Yale Cancer Center, Yale, School of Medicine, New Haven, CT 06511, USA

**Abstract.** Gene set analysis is a leading bioinformatical technique allowing comparison of phenotypes on gene set level, which is applied to different transcriptome-wide gene expression platforms and omics levels. The aim of this study was to measure the performance of three single-sample gene set enrichment algorithms, based on their ability to obtain the statistical significance of enrichment in each cell separately using scRNA-Seq data. The peripheral blood mononuclear cell dataset was used in the evaluation process and individual enrichment within the B cell subtype was investigated based on reference gene set collection. Sensitivity, specificity, prioritization, and balanced accuracy were used as evaluation metrics, accompanied by correlation analysis between gene sets. AUCCell, originally designed for scRNA-Seq, showed the best sensitivity and balanced accuracy, good prioritization and acceptable specificity. However, large correlation between gene set size and specificity was observed, so we recommend its usage on large gene sets (>80). Moreover, the computational time is much longer compared to other tested methods. Among other algorithms, CERNO gave very high specificity and prioritization, but the sensitivity needs to be enhanced by algorithm improvement. Finally, the problem of the “gold standard” dataset and gene set collection that could be used for gene set analysis algorithms performance evaluation in scRNA-Seq, was stated and the initial solution was presented.

**Keywords:** Pathway enrichment analysis · Single-cell RNA sequencing · Single-sample algorithms · Algorithms effectiveness

## 1 Introduction

Ever since the invention of transcriptome-profiling technologies, the measurement of gene expression in different cells/samples is possible. As the abundant number of genes occurring in a cell would be hard to process individually, making both calculations and conclusions challenging, the genes are gathered into groups based on mechanisms they regulate. These groups of genes are called gene sets (GSs) and stored in databases



such as KEGG [1] or GO [2], expanding the knowledge about biological processes in living organisms. Such collections allow differentiating between various groups, aiding diagnostics or observing the characteristics of specific phenotypes and their reactions to the environment.

One of the methods of gene expression measurement is microarrays where the cell mixture from samples of different phenotypes is investigated. For the analysis of transcriptomic experiments in the form of gene set activity, approaches known as Gene Set Analysis (GSA) were introduced. Those bioinformatics algorithms can be divided based on study type: per group or per sample. Per group methods are based on the designation of control and test group where the genes' expressions are transformed to GS activation. In this category, such methods as Gene Set Enrichment Analysis (GSEA) [3], Link Enrichment of Gene Ontology (LEGO) [4], or Coincident Extreme Ranks in Numerical Observations (CERNO) [5] can be distinguished. The second group of algorithms is known as single-sample approaches, where activation of GS is explored within one sample instead of a group of samples with the same phenotypes. Here, such methods like Pathway Level Analysis of Gene Expression (PLAGE) [6], Gene Set Variation Analysis (GSVA) [7], Pathway Analysis for Sample-level Information (PASI) [8], *singscore* [9], or z-score [10], where the scores are calculated for each cell/sample and pathway separately, can be included. The second group of GSA methods could be easily applied in single-cell RNA sequencing (scRNA-Seq) as they will allow to reveal individual cell enrichment of particular GS/pathway in the form of pathway activation score (PAS).

Methods that are designed for microarrays tend to be used in other transcriptome-wide gene expression platforms and techniques like bulk RNA sequencing (RNA-Seq) and scRNA-Seq. In [11, 12] authors show that applying GSA algorithms designed for microarray analysis into bulk RNA-Seq does not impact their effectiveness. However, in analysis of scRNA-Seq different issues may occur. The main problem in scRNA-Seq is data sparsity and previously designed algorithms might not be robust to this form of count data. Moreover, the methods should be robust against the effects of normalization as the highest number of counts may be overestimated. Thus, a few approaches were designed especially for enrichment analysis in scRNA-Seq, e.g. AddModuleScore [13] from *Seurat* R package, where the presented problem requires knowledge of control and test groups (clusters) before performing the analysis. Also, comparisons are made across the dataset and the results strongly depend on cell data composition. The resulting values do not provide an explicit cut-off value dividing the cells into significant and non-significant. Another method, UCell [14], removes the issue of the normalization effect by introducing individual ranking scores for each sample (transformation to non-parametric approach). Moreover, this solution allows for the analysis of single cells (without including information about clusters) in comparison to AddModuleScore. Again, there is no threshold classifying the significance of the pathway in the studied cell. The desired cut-off value is introduced in AUCell [15], which analyses the AUC scores for each cell separately, then based on the results across all the samples for the pathway calculates several thresholds and chooses one of them as the final one.

In [16] authors test the performance of enrichment algorithms in terms of accuracy, scalability, and stability based on PAS matrix. Here, we focus on methods that not only give PAS but can state enrichment significance in each cell. We chose two methods

previously used in microarray data analysis (CERNO and z-score), which will produce the results in the form of p-values and therefore, the easily retrievable, and interpretable cut-off value for each GS can be obtained for scRNA-Seq data. We evaluate them against AUCell as the only enrichment method dedicated to scRNA-Seq analysis, which simultaneously allows for significant classification of a gene set in each cell. In the comparison, the following metrics were evaluated: sensitivity, specificity, prioritization, balanced accuracy, and correlation to gene set size.

## 2 Materials and Methods

### 2.1 Data Acquisition and Pre-processing

The raw scRNA-Seq data of peripheral blood mononuclear cells (PBMC) were downloaded from single-cell portal of Broad Institute [17]. Out of all measurements, the PBMC experiment 1 performed on 10x Chromium (v2) A was extracted. The initial dataset consists of 3,222 cells and 33,694 gene transcripts. Transcripts with low expression were filtered out using Gaussian Mixture Model (GMM) [18] on the logarithmic distribution of zero counts number per transcript. Next, the threshold for the intersection of two first components was extracted and transcripts with a number of zero counts above the threshold were removed [19]. Duplicated genes were removed by keeping genes with higher variance across cells. Finally, the 3,222 cells and 15,817 unique gene transcripts were analyzed. The cell labels were provided with the dataset to distinguish nine different cell types: B cell ( $n = 288$ ), CD14 + monocyte ( $n = 640$ ), CD16 + monocyte ( $n = 102$ ), CD4 + T cell ( $n = 550$ ), Cytotoxic T cell ( $n = 1,174$ ), Dendritic cell ( $n = 55$ ), Megakaryocyte ( $n = 221$ ), Natural killer cell ( $n = 166$ ) and Plasmacytoid dendritic cell ( $n = 26$ ).

For the collated dataset the pathway enrichment analysis was performed on a dedicated gene set (GS) collection. In the KEGG database [1] pathways from “Organismal Systems; Immune system” category were extracted. Furthermore, out of the collections of Chaussabel et al. [20] (DC) and Li et al. [21] (LI) available in *tmod* R package [5] gene sets whose names include one of the analyzed cell types were selected. The third group of gene sets was the LM22 list from CIBERSORT software which contains a signature of PBMC types [22]. Finally, the simulated GSs with random genes and various sizes were created (GS sizes: 20, 50, 80, 100, 150, 300, 500) as a negative control. In the process of random selection, the genes from CIBERSORT subcollection were excluded as they represent signatures of PBMC cell types. In summary, 103 gene sets were used (KEGG  $n = 21$ , LI/DC = 53, CIBERSORT = 22, Random = 7). The coverage of gene signature between GSs was investigated by the Jaccard index and the results were clustered using hierarchical clustering with Euclidean distance.

### 2.2 Single-Sample Pathway Enrichment Algorithms

Three different methods of single-sample enrichment were tested. The first algorithm is AUCell which is part of an R package *SCENIC* dedicated to complex scRNA-Seq analysis [15]. In this method, at first, the gene counts are ranked from high to low for

each cell separately, with ties being placed at random. For further calculations, the top 5% of the highest-ranking genes for the specific cell are used. The score for the chosen gene set and cell is calculated as the area under the curve (AUC) of a number of genes in the gene set and gene rank normalized by the maximum possible area under the curve for the used number of genes. Based on AUC distribution, for particular gene set across all cells, the significance cutoff is estimated. For the threshold choice, the following conditions are checked: (i) normal distribution assumption tested by Kolmogorov-Smirnoff test for all AUCs of the individual genes and then the calculation of:

$$p = 1 - \frac{0.01}{N} \quad (1)$$

where  $N$  is the total number of tested cells. The obtained result is treated as the percentile of a normal distribution with mean and standard deviation of all AUC scores across all cells for a particular gene set; (ii) Kernel fitting is conducted for the obtained AUC scores for the analyzed gene set. For the obtained distribution global maximum, local maximum and local minimum placed before the second-highest local maximum are found. If the ratio of the density of local maximum to the density of global maximum is higher than 0.05, then the local minimum is set as a threshold. If none of the above are satisfied, the threshold is calculated in the following way: (i) The following statistic is calculated:

$$p = 1 - \left( \frac{0.01}{N} - 0.25 \right) \quad (2)$$

The obtained value is again approximated to the percentile of a normal distribution with a mean and standard deviation of all AUC scores across all cells for a particular gene set; (ii) Two Gaussian distributions are fitted to the calculated AUC scores. Next, the cut-off is set as a percentile from Eq. (1) and the normal distribution with a mean and standard deviation of the second component of fitted GMM model; (iii) The same as for the above is repeated for the three-component GMM model and the cutoff is set as  $p = 0.01$  for the normal distribution with mean and standard deviation of the third component.

The next two algorithms were originally designed for single-sample microarray GS analysis. The reasoning of their selection is that out of the range of different single-sample approaches, they set significance for each individual sample without original algorithm modification and can easily be applied to scRNA-Seq analysis. The first one is the z-score [10] where the counts are normalized over all the genes in the cell. Next, genes in the GS are combined via Stouffer's method [23] for each cell separately:

$$Z_{GS} = \frac{\sum_{i=1}^N Z_i}{\sqrt{N}} \quad (3)$$

where  $Z_i$  is the normalized count of gene  $i$  and  $N$  is the total number of genes in the analyzed GS that occur for the tested cell. The obtained statistic can follow the standard normal distribution under the null hypothesis that the genes in the tested GS for the tested cell are distributed randomly.

The last tested algorithm is CERNO [5] where sorted rank lists of genes for each of the cells separately are used. The counts are ranked from high to low and ties are

placed at random (same as in AUCell). The null hypothesis of the test assumes a random distribution of genes belonging to individual GS for the tested sample. The  $F$  statistic for each cell and gene set is calculated as:

$$F_{GS} = -2 \sum_{i=1}^N \ln \left( \frac{r_i}{N_{tot}} \right) \chi_{2N}^2 \quad (4)$$

where  $N$  is the total number of genes in a given GS,  $N_{tot}$  is the total number of analyzed genes and  $r_i$  is the rank of gene  $i$  for an investigated cell at GS. The obtained F distribution statistic is approximated with  $\chi^2$  distribution with  $2*N$  degrees of freedom.

Both z-score and CERNO produce p-values, which were then corrected via the usage of the Benjamini-Hochberg procedure (BH) [24]. For both algorithms, GS is taken as significant if the p-value or q-value (after correction for multiple testing) is lower than 0.05. In total 5 different enrichment approaches were tested (AUCell, both CERNO and z-score with and without BH correction).

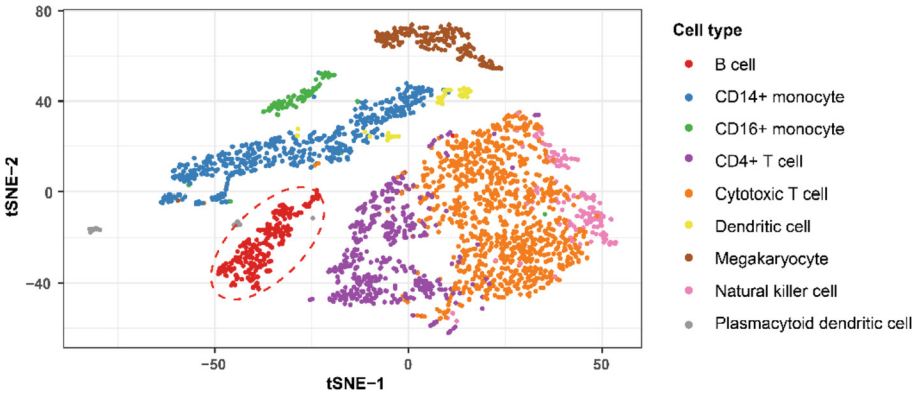
### 2.3 Algorithm's Evaluation

Each enrichment method was run on 103 GSs for each cell separately. As a result, the significance of each cell, in each GS was obtained. Out of all GSs, only pathways related to B cells were extracted, as the B cell type is the most homogenous cluster out of all investigated cell subtypes in PBMC dataset. Twenty-one pathways related to B cells were recognized in the investigated GS collection (KEGG  $n = 1$ , CIBERSORT  $n = 2$ , LI/DC  $n = 18$ ). Those pathways were set as a reference and based on their gene signature the B cell subtype detection was evaluated. To assess the performance of enrichment algorithms, sensitivity, specificity and balanced accuracy were calculated for each enrichment method and B cell type gene set. Next, prioritization was calculated as follows: (i) for each cell and each analyzed GS, the GSA results were ranked from the most to the least significant, (ii) next ranks were divided by the total number of analyzed GS ( $n = 103$ ), (iii) mean across all B cells was calculated for each GS which represents surrogate prioritization, (iv) finally, surrogate prioritization was extracted for reference pathways. The above prioritization metric was presented in [25] and it is adjusted for scRNA-seq data. As the next part of the algorithm's evaluation, Spearman rank correlation was calculated between evaluation metrics and GS size (number of analyzed genes of PBMC dataset in the reference gene set). Finally, the level of false positives (FP) in the randomly generated pathway was investigated as for those GSs none of the cells should be detected. All calculations were performed using Python version 3.8.12 and visualizations were prepared within R programming language version 4.1.1 *ggplot2* package.

## 3 Results

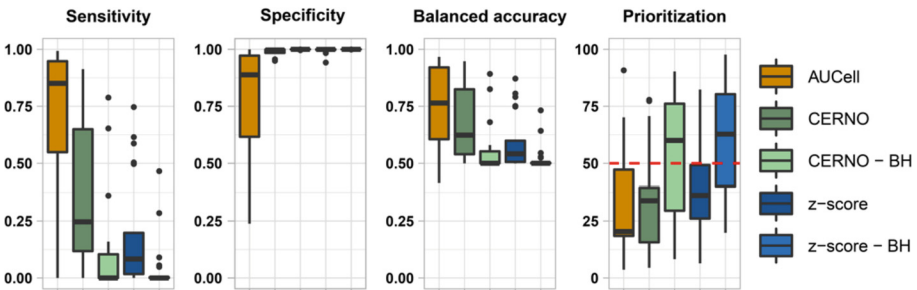
At first, the counts from pre-processed PBMC dataset were log2 normalized and the tSNE dimensionality reduction technique [25] was used to explore cell types grouping. In Fig. 1 the first and second tSNE instances are presented for all 3,222 cells and colors represent

each cell type. As can be observed, the majority of PBMC cell types are well grouped. Only Cytotoxic T, CD4 + T and Natural killer cells are aggregated together creating a large group of cells with the internal division to each cell type. B cells are notably separated from other groups which indicates that they have a unique gene expression pattern. Moreover, in gene set collection the largest group of pathways characterizes this cell type (21 gene sets). Thus, in further evaluation, only B cell type enrichment for reference 21 gene sets is investigated.



**Fig. 1.** The tSNE projection of investigated scRNA-Seq dataset. Colors represent cell types, while the dashed line distinguishes B cell cluster.

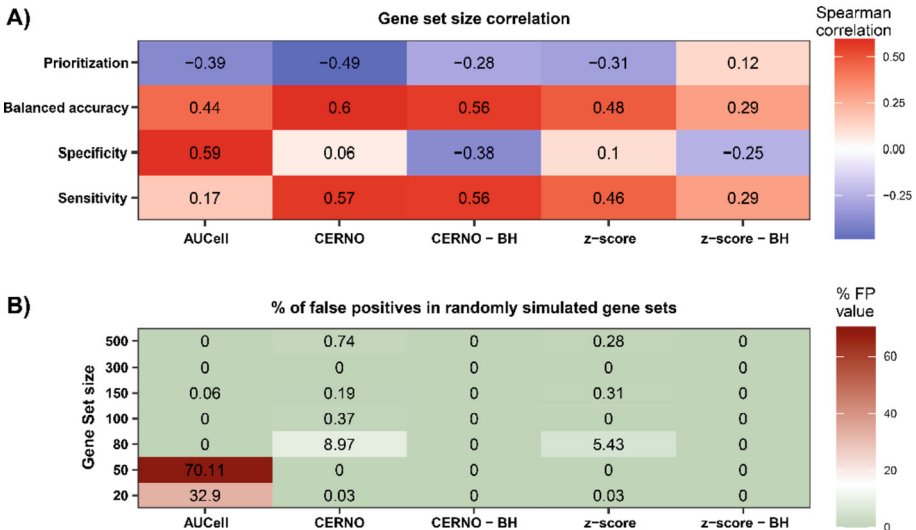
Next, for each of the reference 21 gene sets, detection of B cells was examined by calculating sensitivity, specificity, balanced accuracy, and prioritization for 5 different tested enrichment approaches (Fig. 2). AUCell obtained the best performance in terms of sensitivity, balanced accuracy, and prioritization (Fig. 2). However, CERNO and z-score with and without BH correction were characterized by better specificity. This indicated that both CERNO and z-score tend to not detect all cells in the reference cluster, but the classification of other cell types as B cell is limited. This trait is stronger when multiple



**Fig. 2.** Evaluation metrics results for tested algorithms based on detection of B cell in 21 reference gene sets. For sensitivity, specificity and balanced accuracy the higher value the better. The lower value of prioritization the better algorithm performance.

testing correction is applied (BH). In addition, CERNO has very good prioritization comparable to AUCell, which indicates that despite lower sensitivity, the GSs of interest are at the top of the list. Out of microarray-designed approaches, CERNO has the best performance across all four tested metrics.

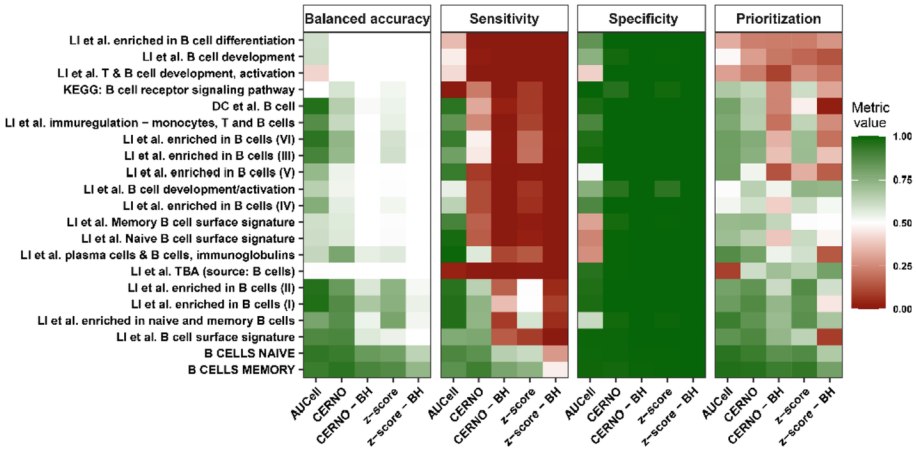
In the next step, the correlation of evaluation metrics with GS size was investigated and the results are presented in Fig. 3 panel A). Specificity for AUCell is strongly positively correlated with GS size which is further reflected in medium effect on balanced accuracy. Moreover, the medium effect is observed for prioritization and small for sensitivity. CERNO approaches show a large positive correlation in terms of sensitivity and balanced accuracy in parallel with a large negative correlation to prioritization (the larger GS size the better prioritization). Moreover, when BH correction is applied the specificity shows a medium negative correlation. However, this is the effect of a small variation in specificity and the estimated correlation may be inaccurate. Similar observations can be made for the z-score method with lower effects. In Fig. 3 panel B) the level of false positives in randomly generated GSs of different sizes is presented. AUCell has a high level of false positives on small GSs (<80 genes). This observation is concordant with Fig. 3 panel A) where together with the increase of GS size the specificity increases as an effect of false positive decreasing. In terms of CERNO and z-score, the robustness to GS size is observed (stronger for BH correction), which confirms the high specificity of those methods, also on randomly generated data.



**Fig. 3.** Algorithm performance in terms of gene set size. Panel A) shows correlation between evaluation metrics and gene set size of reference gene sets. Panel B) presents percentage of false positives regardless of simulated gene set size.

Further, evaluation metrics were checked for each reference GS separately and results are presented in Fig. 4. GSs from CIBERSORT (B CELL NAÏVE and B CELL MEMORY) were well classified by AUCell, CERNO (with and without BH correction) and

z-score, while for z-score with BH correction the sensitivity decreases compared to other methods. Taking into consideration all reference GSs, AUCell method has the best performance and only a few gene sets show medium performance. From methods designed for the microarrays analysis, CERNO without BH correction gives the best results and over half of its results have similar performance levels to AUCell. Out of reference GSs, surprisingly, the pathway from KEGG collection gives the poorest results in terms of sensitivity. However, the worst outcomes across all tested approaches were obtained for a pathway from Li et al. [21] collection: T & B cell development, activation.



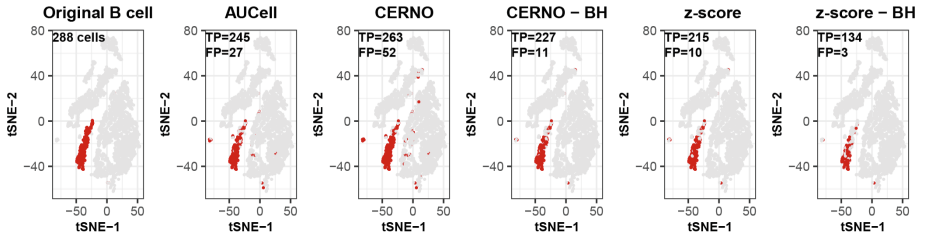
**Fig. 4.** Heatmap of evaluation metrics for each B cell gene set separately. Green color represents good performance while red represents poor performance. (Color figure online)

For the reference GSs with best (B CELL MEMORY) and the worst performance (Li et al. T & B cell development, activation) across all tested methods, the tSNE projection with marked recognized cells as enriched was prepared (Fig. 5). As can be found in Fig. 5 panel A), in B CELL MEMORY GS the best true positive rate on B cell recognition is observed for CERNO approach, while the best false positive control is obtained for z-score with BH correction. In terms of Li et al. T & B cell development, activation GS (Fig. 5 panel B) approaches designed for microarrays almost do not detect any cell as significantly enriched. AUCell pointed too many cells as significantly enriched in the B cell cluster (FP = 1,751 cells). However, the pathway itself contains the signature of both B and T cells, and the cells in the right bottom cluster represent mainly CD4+ T cell and Cytotoxic T cell. Thus, for design evaluation process judgment of performance as poor for AUCell method for Li et al. T & B cell development GS is ambiguous.

Finally, the Jaccard index was calculated to reveal coverage of genes within reference gene sets (Fig. 6). The investigated GSs are characterized by mainly very low coverage of gene signature. Thus, the presented reference GS collection does not contain redundant information and mainly each GS checks different characteristics of B cell activity. As can be expected from performance analysis (Fig. 4), B CELL MEMORY and B CELL NAIVE (Cluster 1) GSs are very similar and mostly correctly recognized within the B cell subtype by all algorithms. The next groups of pathways are: Li et al. enrichment in B



A) B CELL MEMORY



B) Li et al. T & B cell development, activation

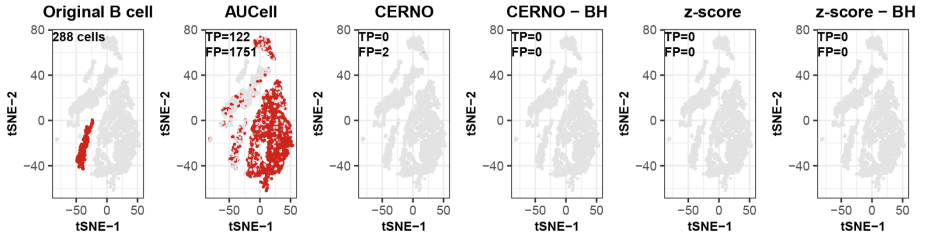


Fig. 5. The tSNE projection with marked detected cells for two different gene sets. Panel A) shows results for B CELL MEMORY gene set across each tested algorithm. Panel B) shows results for T & B cell development, activation gene set from Li et al. collection across each tested algorithm.

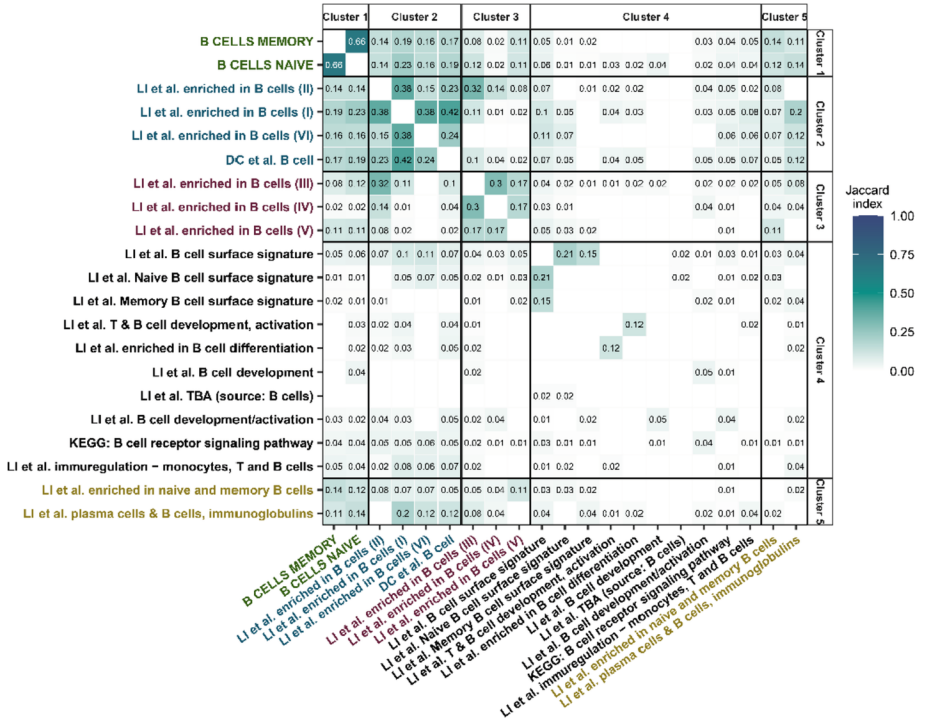


Fig. 6. Jaccard index value for the investigation of reference gene sets coverage. Clusters were established by average hierarchical clustering on Euclidian distance.



cells (I), (II) and (VI) and DC et al. B cells. Those GSs have medium similarity between each other and very small coverage with GSs from cluster 1. However, those GSs were correctly recognized by AUCell and CERNO. Similar observations can be noted for GSs in clusters 4 and 5. GSs in cluster 4 are characterized by very low signature coverage and the worst performance of algorithms in terms of B cell subtype detection.

## 4 Conclusions

The general comparison of three different enrichment algorithms which can estimate gene set significance of individual cells in scRNA-Seq data was presented. Each algorithm works with different assumptions but all of them are robust to data normalization of scRNA-Seq data and can work on both, raw and normalized counts.

Out of tested methods, only the AUCell was originally designed for scRNA-Seq data. The AUCell was characterized by the best sensitivity, prioritization, and balanced accuracy but the worst specificity. Moreover, the specificity is strongly positively correlated with GS size with only a small correlation to sensitivity. Thus, this method is reliable for larger size gene sets. This feature was also observed on false positive levels calculated on randomly generated gene sets. Two other algorithms (CERNO and z-score) were originally designed for microarray data analysis. In this group, better outcomes were obtained for CERNO algorithm without correction for multiple testing. CERNO surpassed z-score in all three metrics but the correlation of sensitivity with GS size was large and reflected in balanced accuracy. Moreover, better prioritization was obtained for larger GSs. Both CERNO and z-score have a very low false positive level on the randomly generated gene set, regardless of gene set size. Thus, the specificity of both methods seems to be robust to the size of the investigated pathway. In [16] they showed that Pagoda2 (designed for scRNA-Seq data) method is the best for GSA when only PAS matrix is in consideration. Moreover, PLAGEMethod (designed for microarrays) has also good performance. In the presented study, results indicated that methods originally designed for microarrays need adjustment before their usage in scRNA-Seq, when the significance of individual cell enrichment is of scientific interest. Both AUCell and CERNO at the beginning rank genes in each cell, but further AUCell takes only the top 5%. Thus, such modification of CERNO should be applied in terms of improving the results, especially sensitivity. This will also allow for faster execution on big datasets, and simultaneously should reduce the effect of the noise at the bottom of the gene ranking caused by zero counts. However, CERNO and z-score are very fast algorithms (mean calculation time in minutes: 0.08 with 95% CI [0.07 – 0.09] and 0.086 with 95% CI [0.084 – 0.088] respectively). The AUCell mean execution time is 2.5 min (95% CI [2.4 – 2.6]) where over 2 min are spent on the threshold estimation. Thus, a faster solution for threshold identification is needed for AUCell or an improvement in the original implementation. Finally, in the presented work significance of each cell was established, alongside each of the tested algorithms allowing presentation of results as a continuous heatmap (PAS matrix) like e.g. [14].

Lastly, there is still a lack of “gold standard” datasets but more importantly “gold standard” gene set collections which are assigned to cell type labels. Out of 21 reference pathways, those from CIBERSORT have the strongest signature of PBMC cell types, but some gene sets can be assigned to multiple cell types which must be incorporated

in future evaluation designs. By investigation of signature coverage, we propose to use only GSs from clusters 1, 2, 3 and 5 as the reference for B cell detection. Nevertheless, performance designs like those in GSA on microarrays (e.g. [5, 12, 26, 27]) are needed in scRNA-Seq data analysis.

**Acknowledgements.** This work was financed by Silesian University of Technology grant for maintaining and developing research potential (MM, JZ). Anna Mrukwa takes part in mentor program “Spread your wings” at Silesian University of Technology and was financed by reserve of the Vice-Rector for Student Affairs and Education (60/001).

## References

1. Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y., Morishima, K.: KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res* **45**, D353–D361 (2017)
2. Consortium, G.O.: The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Res.* **32**, D258–D261 (2004)
3. Subramanian, A., et al.: Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc Natl Acad Sci U S A* **102**, 15545–15550 (2005)
4. Dong, X., Hao, Y., Wang, X., Tian, W.: LEGO: a novel method for gene set over-representation analysis by incorporating network-based gene weights. *Sci. Rep.* **6**, 18871 (2016)
5. Zyla, J., Marczyk, M., Domaszewska, T., Kaufmann, S.H.E., Polanska, J., Weiner, J.: Gene set enrichment for reproducible science: comparison of CERNO and eight other algorithms. *Bioinformatics* **35**, 5146–5154 (2019)
6. Tomfohr, J., Lu, J., Kepler, T.B.: Pathway level analysis of gene expression using singular value decomposition. *BMC Bioinformatics* **6**, 225 (2005)
7. Hänzelmann, S., Castelo, R., Guinney, J.: GSVA: gene set variation analysis for microarray and RNA-Seq data. *BMC Bioinformatics* **14**, 7 (2013)
8. Jaakkola, M.K., McGlinchey, A.J., Klen, R., Elo, L.L.: PASI: a novel pathway method to identify delicate group effects. *PLoS ONE* **13**, e0199991 (2018)
9. Foroutan, M., Bhuvu, D.D., Lyu, R., Horan, K., Cursons, J., Davis, M.J.: Single sample scoring of molecular phenotypes. *BMC Bioinformatics* **19**, 1–10 (2018)
10. Lee, E., Chuang, H.-Y., Kim, J.-W., Ideker, T., Lee, D.: Inferring pathway activity toward precise disease classification. *PLoS Comput. Biol.* **4**, e1000217 (2008)
11. Zyla, J., Leszczorz, K., Polanska, J.: Robustness of pathway enrichment analysis to transcriptome-wide gene expression platform. In: *International Conference on Practical Applications of Computational Biology & Bioinformatics*, pp. 176–185. Springer (Year)
12. Geistlinger, L., et al.: Toward a gold standard for benchmarking gene set enrichment analysis. *Brief Bioinform* **22**, 545–556 (2021)
13. Stuart, T., et al.: Comprehensive integration of single-cell data. *Cell* **177**, 1888–1902 (2019). e1821
14. Andreatta, M., Carmona, S.J.: UCell: robust and scalable single-cell gene signature scoring. *bioRxiv* (2021)
15. Aibar, S., et al.: SCENIC: single-cell regulatory network inference and clustering. *Nat. Methods* **14**, 1083–1086 (2017)
16. Zhang, Y., et al.: Benchmarking algorithms for pathway activity transformation of single-cell RNA-seq data. *Comput. Struct. Biotechnol. J.* **18**, 2953–2961 (2020)

17. Ding, J., et al.: Systematic comparison of single-cell and single-nucleus RNA-sequencing methods. *Nat. Biotechnol.* **38**, 737–746 (2020)
18. Marczyk, M., Jaksik, R., Polanski, A., Polanska, J.: GaMRed—adaptive filtering of high-throughput biological data. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **17**, 149–157 (2020)
19. Widlak, P., et al.: Detection of molecular signatures of oral squamous cell carcinoma and normal epithelium—application of a novel methodology for unsupervised segmentation of imaging mass spectrometry data. *Proteomics* **16**, 1613–1621 (2016)
20. Chaussabel, D., et al.: A modular analysis framework for blood genomics studies: application to systemic lupus erythematosus. *Immunity* **29**, 150–164 (2008)
21. Li, S., et al.: Molecular signatures of antibody responses derived from a systems biology study of five human vaccines. *Nat. Immunol.* **15**, 195–204 (2014)
22. Chen, B., Khodadoust, M.S., Liu, C.L., Newman, A.M., Alizadeh, A.A.: Profiling tumor infiltrating immune cells with CIBERSORT. *Methods in molecular biology* (Clifton, NJ) **1711**, 243 (2018)
23. Demerath, N.J.: *The American Soldier: Volume I, Adjustment During Army Life.* By S. A. Stouffer, E. A. Suchman, L. C. DeVinney, S. A. Star, R. M. Williams, Jr. *Volume II, Combat and Its Aftermath.* By S. A. Stouffer, A. A. Lumsdaine, M. H. Lumsdaine, R. M. Williams, Jr., M. B. Smith, I. L. Janis, S. A. Star, L. S. Cottrell, Jr. Princeton, New Jersey: Princeton University Press, 1949. Vol. I, 599 pp., Vol. II, 675 pp. \$7.50 each; \$13.50 together. *Social Forces* **28**, 87–90 (1949)
24. Benjamini, Y., Hochberg, Y.: Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. Roy. Stat. Soc.: Ser. B (Methodol.)* **57**, 289–300 (1995)
25. Maaten, L.V.D., Hinton, G.: Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008)
26. Tarca, A.L., Bhatti, G., Romero, R.: A comparison of gene set analysis methods in terms of sensitivity, prioritization and specificity. *PLoS ONE* **8**, e79217 (2013)
27. Xie, C., Jauhari, S., Mora, A.: Popularity and performance of bioinformatics software: the case of gene set analysis. *BMC Bioinformatics* **22**, 191 (2021)



# Comparison of Stranded and Non-stranded RNA-Seq in Predicting Small RNAs in a Non-model Bacterium

Karel Sedlar<sup>1,2</sup>(✉)  and Ralf Zimmer<sup>1</sup> 

<sup>1</sup> Institute of Bioinformatics, Department of Informatics, Ludwig-Maximilians-Universität München, Munich, Germany  
sedlar@bio.ifi.lmu.de

<sup>2</sup> Department of Biomedical Engineering, Faculty of Electrical Engineering and Communication, Brno University of Technology, Brno, Czechia

**Abstract.** Thanks to their diversity, non-model bacteria represent an inexhaustible resource for microbial biotechnology. Their utilization is only limited by our lack of knowledge regarding the regulation of processes they are capable to perform. The problem lies in non-coding regulators, for example small RNAs, that are not so widely studied as coding genes. One possibility to overcome this hurdle is to use standard RNA-Seq data, gathered primarily to study gene expression, for the prediction of non-coding elements. Although computational tools to perform this task already exist, they require the utilization of stranded RNA-Seq data that must not be available for non-model organisms. Here, we showed that *trans*-encoded small RNAs can be predicted from non-stranded data with comparable sensitivity to stranded data. We used two RNA-Seq datasets of non-type strain *Clostridium beijerinckii* NRRL B-598, which is a promising hydrogen and butanol producer, and obtained comparable results for stranded and non-stranded datasets. Nevertheless, the non-stranded approach suffered from lower precision. Thus, the results must be interpreted with caution. In general, more benchmarking for tools performing direct prediction of small RNAs from standard RNA-Seq data is needed so these techniques could be adopted for automatic detection.

**Keywords:** Small non-coding RNA · *Clostridium beijerinckii* NRRL B-598 · RNA-Seq · Genome annotation

## 1 Introduction

It has been almost half a century since small non-coding RNAs (sRNAs) were discovered in bacteria [1]. During years, sRNAs were shown to play important regulatory roles in diverse cellular processes by participating in post-transcriptional regulation of gene expression [2]. This is the reason why sRNAs are drawing more attention than ever before. While the first experiments were done with a model bacterium, *Escherichia coli*, primarily its non-pathogenic strain K-12, later studies showed the role of sRNA in the virulence of pathogenic bacteria [3–5]. Besides their role in medicine, sRNAs

can be used in general biotechnologies for their involvement in other processes, for example, degradation of toxic compounds [6]. Finally, the latest research shows that the engineering of a novel sRNA can improve bacterial phenotype, for example, tolerance to acids [7], which could be utilized in various fermentation processes for the production of bio-based chemicals.

As the former widely used title small non-coding RNA suggests, it is a small molecule that is not translated into a protein. Although this is true in a majority of cases, it has been proved that some sRNAs can encode small proteins [8]. Therefore, it is common that these short regulatory RNAs are simply referred to as small RNA. Its length can vary but it typically spans within the interval 40–500 nucleotides [9]. Most commonly, sRNAs can be divided according to the locations of sRNA genes and their targets into two groups, *cis*-encoded and *trans*-encoded sRNAs [8]. A *cis*-encoded sRNA overlaps with a regulated gene but is coded by the antisense strand and during its regulation binds to the target mRNA by perfect base pairing. Binding can occur at any location depending on the location of sRNA expression [10]. There are three mechanisms that *cis*-encoded sRNAs use for regulation. They can act as transcription terminators, potential inhibitors of translation initiation, or modulators of mRNA degradation. A *trans*-encoded sRNA interacts with its target mRNA by imperfect base pairing because such sRNA is coded by an intergenic region (ITR) and its coding sequence does not overlap with a sequence of the target gene [11]. This also means that *trans*-encoded sRNAs can be coded by the same strand as target genes and they have a wider range of regulatory mechanisms. They can act as repressors of expression but also as activators. They can increase as well as block mRNA degradation.

Some of the early experiments showed that sRNA genes identified in *E. coli* were found in *Salmonella enterica* and vice versa [2]. This suggested their conservation across the bacterial domain and made them ideal targets for computational prediction. A wide range of tools has been proposed. In general, they can be divided into two groups: comparative genomics-based and machine learning-based techniques [8]. While the former techniques rely on sequence alignment and cluster analysis with phylogenetic profiling, the latter are taking advantage of widely used machine learning methods such as neural networks, support vector machines, and genetic algorithms. Nevertheless, these techniques can only predict a location of sRNA but cannot predict its target site, which can be cumbersome, primarily for *trans*-encoded sRNAs that pair imperfectly to target sites. Besides computational solution lying in *in silico* prediction of sRNA-target mRNA interaction, e.g., sRNATarget [12], IntaRNA [13], or RNApredator [14], there is a plethora of techniques based on RNA-Seq to reveal these interactions experimentally [15]. The main disadvantage of these specialized techniques such as GRIL-Seq [16], RIP-Seq [17], RIL-Seq [18], and many others, is their difficult implementation in non-model bacteria that limits their utilization to model organisms, mainly *E. coli* [15]. On the other hand, even standard RNA-Seq that became a commonly used technique in bacterial research, can be used to discover sRNA genes.

Despite existing algorithms as well as experimental techniques, identification of sRNA genes is still not a common procedure during annotation of non-model bacterial genomes. While there is currently more than a million bacterial genome assemblies in the GenBank database (27<sup>th</sup> January 2022), the number of annotated sRNAs for

particular genomes is very limited, usually in units of genes. The most commonly used tool for genome annotation, the PGAP pipeline [19], uses homology-based annotation by scanning the Rfam database [20] with infernal's cmsearch [21]. This suggests that computational prediction of sRNAs in non-model bacteria might be limited by low sequence similarity to model organisms whose sRNAs were discovered experimentally. This opens a door to the utilization of standard RNA-Seq data which is available for many non-model bacteria. Nevertheless, a systematic pipeline for such predictions is missing and various authors use different techniques. Zhu et al. [22] predicted approximately ten sRNAs in *Bifidobacterium animalis* by combining prediction using TargetRNA2 [23] with RNA-Seq data used to calculate RPKM (Reads per kilobase per million) values summarizing expression of identified sRNAs. Liu et al. [24] found 263 sRNAs candidates in *Mycobacterium neoaurum* by combining RPKM and IntaRNA predictions. On the contrary, Wang et al. [25] used RNA-Seq data itself for searching sRNAs in *Mycobacterium tuberculosis* by examining coverage of unannotated regions. Thanks to the utilization of strand-specific RNA-Seq, 192 sRNAs candidates were found in intergenic regions and additional 664 candidates coded by antisense strand in regions overlapping to target genes. Although their study is presented as an automated approach, it brings no computational tool that could be used for another organism.

It is the unavailability of computational tools that prevents the wider utilization of RNA-Seq data in the prediction of sRNA genes in non-model bacteria. There are only a few tools that suffer from various drawbacks. For example, APERO [26] needs paired-end reads which are usually not available for bacterial RNA-Seq data, Rockhopper [27] is very hard to be implemented to other pipelines due to its graphical user interface nature and utilization of obsolete formats such as protein table for genome annotation, and baerhunter [28] is no longer working with the current version of R/Bioconductor. Moreover, benchmarking for different tools is missing and a comparison of prediction possibilities regarding input data was never performed before. In this paper, we got inspired by current tools and performed sRNAs prediction in the non-model bacterium *Clostridium beijerinckii* NRRL B-598 [29] using two different RNA-Seq datasets taken under the same conditions. We showed that the current approach in sRNAs prediction can be, with some limitations, applied to both, stranded as well as non-stranded RNA-Seq data and that more benchmarking is needed to establish functional pipelines for sRNAs prediction using standard RNA-Seq data.

## 2 Materials and Methods

### 2.1 Genome and Annotation

To examine sRNAs prediction in a non-model bacterium, we selected *C. beijerinckii* NRRL B-598, a non-type strain, which is a promising butanol and hydrogen producer. Most importantly, it is a non-type strain with the highest number of RNA-Seq-based transcriptomic studies among solventogenic clostridia [30]. In this study, we used its third complete genome assembly, available at DDBJ/EMBL/GenBank under accession No. CP011966.3, which was constructed using a combination of Roche 454 GS Junior, PacBio RSII, and Illumina NextSeq500 reads [31]. The genome annotation was performed with PGAP v4.6 [19] and genome features are summarized in Table 1.

**Table 1.** Genome features of *Clostridium beijerinckii* NRRL B-598.

Feature	Chromosome
Length (bp)	6,186,993
GC content (%)	29.8
Protein coding genes	5,128
Pseudogenes	166
rRNAs (5S, 16S, 23S)	17, 16, 16
tRNAs	94
Non-coding RNAs	5
Riboswitches	31

## 2.2 Transcriptomic Data

RNA-Seq data used in this study comes from a publicly available study performing transcriptional profiling of the butanol fermentation using glucose as a substrate [32]. Two particular samples, A and B, from the exponential growth phase, after 3.5 h from the start of fermentation, were selected. These samples are available from the NCBI Sequence Read Archive (SRA) under the project accession number PRJNA229510. Cell samples for isolation of total RNA were collected from 3 ml of culture broth (OD600 0.9–1.0) by centrifugation at 10000 rpm for two minutes, washed with RNase free water and cell pellets were immediately stored at  $-70^{\circ}\text{C}$ . RNA from the cell pellet was isolated using High Pure RNA Isolation Kit (Roche). Isolated total RNA was stored frozen at  $-70^{\circ}\text{C}$ . The total RNA concentration was determined on DS-11 FX + Spectrophotometer (DeNovix). Quality and integrity of the samples were assessed using the Agilent RNA 6000 Nano Kit (Agilent) with the Agilent 2100 Bioanalyzer (Agilent). RNA integrity number was measured using 2100 Bioanalyzer Expert software. Frozen total RNA samples were thawed on ice and an aliquot of each sample containing 10  $\mu\text{g}$  of RNA was taken for 16S and 23S ribosomal RNAs removal using The MICROBExpress™ Bacterial mRNA Enrichment Kit (Ambion). Efficiency of ribosomal RNA depletion and concentration of RNA samples were checked on the Agilent 2100 Bioanalyzer (Agilent) with the Agilent RNA 6000 Nano Kit (Agilent).

For sample A, library construction and sequencing was performed by BGI Europe A/S (Copenhagen, Denmark). During the library preparation, cDNA was synthesized by using a random hexamer-primer and the sample was sequenced on Illumina HiSeq 4000, single-end, 50 bp. This means that resulting reads are non-stranded, i.e., it is not possible to determine a strand of DNA that codes genes producing sequence transcript as reads mapping to analyzed loci have both orientations.

For sample B, library construction and sequencing was performed by CEITEC Genomics core facility (Brno, Czechia). NEBNext Ultra II stranded kit was used for library preparation and the sample was sequenced on Illumina NextSeq500, single-end, 75 bp. This resulted in reads that are reversely stranded, i.e., the reads have the opposite orientation to the locus producing sequenced transcripts.

### 2.3 Data Preprocessing

Adapter and quality trimming was performed using Trimmomatic v0.36 [33]. Two different settings were used for comparison. In the first settings, parameters LEADING and TRAILING specifying minimum qualities (PHRED score) to keep a base, were both set to three. The length of the SLIDINGWINDOW parameter was set to four and required average quality of 15. Finally, only reads reaching the length of 36 bases were kept by setting up a parameter MINLEN. In the second settings, the parameters were stricter. Minimum qualities were both set to 10 and a sliding window of the length four required at least a quality of 25. On the other hand, reads of length 20 nucleotides and more were preserved.

Although laboratory ribodepletion was performed prior to sequencing, the step of computational rRNA filtering was done for comparison. This step was done with SortMeRNA v2.1 [34] using the SILVA database [35] of known bacterial 16S and 23S rRNA genes. Finally, the mapping to the reference genome was performed with STAR v2.5.4b [36]. Reads mapping to more than three loci were filtered out by setting up a parameter outFilterMultimapNmax.

Quality assessment after particular steps was performed using FastQC in combination with MultiQC [37] to summarize the reports. The resulting SAM (Sequence Read Alignment/Map) files were indexed and transformed into more compact BAM (Binary Read Alignment/Map) format using SAMtools v1.7 [38].

### 2.4 sRNAs Prediction

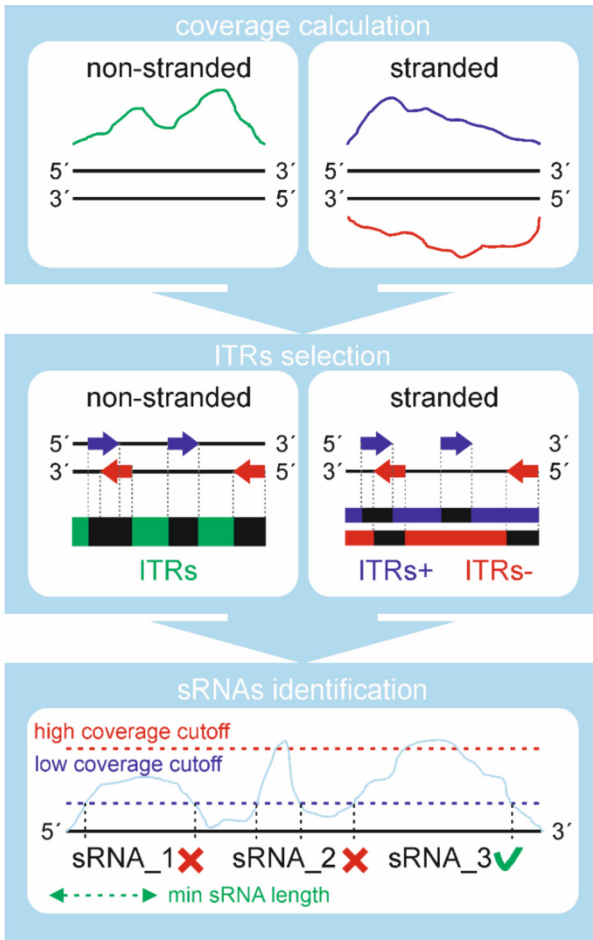
The prediction of sRNA loci was performed in R v4.1.2 and Bioconductor v3.14. The whole pipeline was inspired by *baerhunter* [28] that uses thresholding of coverage. *Baerhunter* itself cannot be used due to erroneous functions for counting sRNAs and untranslated regions (UTRs). Nevertheless, the pipeline was reproduced by rewriting these functions to be compatible with the current Bioconductor. Although *baerhunter* requires stranded RNA-Seq data, the whole pipeline can be reproduced by similar custom-made code that works also with non-stranded data. The main steps of the pipeline are summarized in Fig. 1.

The main idea of thresholding coverage requires coverage to be counted across the whole reference sequence in the first step. This can be achieved using *samtools depth* or by loading BAM files into R/Bioconductor and calculating coverage with suitable functions, for example “coverage” from the *GenomicAlignments* [39] package. In the case of non-stranded data, the coverage is calculated for the chromosome at once. However, for stranded data, coverages of particular strands of DNA have to be calculated separately. Before thresholding is performed, only ITRs are selected. This again requires selecting these regions separately for particular strands in the case of stranded RNA-Seq.

Selecting putative sRNAs inspired by *baerhunter* requires three input parameters. The first parameter “low coverage cutoff” is used to select potential sRNA loci. Once the coverage exceeds the threshold, the start of a potential sRNA is marked. The region is being continually expanded until coverage falls under the threshold again. Other parameters are used for additional filtering. The parameter “high coverage cutoff” sets another threshold for coverage. Only previously selected regions in which at least one



base is covered by more reads than the thresholds are preserved. The last parameter “min sRNA length” simply filters out regions that are shorter than the selected length.



**Fig. 1.** A schema of coverage-based identification of sRNAs. Coverage of ITRs in examined. Here, only a sRNA\_3 candidate is returned as a putative sRNA as it meets high coverage and min sRNA length cutoff value criteria.

The thresholds used in this study were, 10 for the low coverage cutoff, 50 for the high coverage cutoff, and 40 for the min sRNA length. The values were set empirically based on benchmarking study of baerhunter [28].

### 3 Results and Discussion

#### 3.1 Data Preprocessing

Sample A contained 21 million and sample B had 53 million raw sequences. The initial quality assessment showed high GC content suggesting remaining rRNA contamination. The resulting numbers of reads after filtering and mapping steps are summarized in Table 2. Particular parameters settings for quality trimming can be found in materials and methods.

**Table 2.** Results of data preprocessing

Sample	Trimming settings	rRNA removal	No. of reads in a sample (million)	No. of mapped reads (million)
A1	1	No	21.0	11.9
A2	2	No	20.6	11.7
A1r	1	Yes	12.3	11.8
A2r	2	Yes	12.2	11.6
B1	1	No	52.5	15.3
B2	2	No	48.9	14.3
B1r	1	Yes	15.2	14.6
B2r	2	Yes	15.7	13.7

The results showed very high, up to 73%, contamination by rRNA. Although rRNA is filtered during mapping as multi-mapped reads, numbers of mapped reads for samples with and without computational ribodepletion are different, therefore, this step may affect the final identification of sRNA genes.

#### 3.2 sRNAs Prediction in Stranded Data

Before comparison of stranded and non-stranded data, we performed prediction of sRNAs by the same procedure that is used in baerhunter to identify putative sRNA genes as they have never been reported in *C. beijerinckii* NRRL B-598 genome before. The sensitivity of baerhunter was tested against more complex tools, particularly Rockhopper, APERO, and ANNOgesic, using simulated as well as real datasets [28]. Thus, we used its predictions, summarized in Table 3, to estimate sRNAs counts.

**Table 3.** Numbers of sRNAs predicted by baerhunter

Sample	No. of sRNA genes		
	<i>trans</i> -encoded	<i>cis</i> -encoded	Total number
B1	121	115	236
B2	115	99	214
B1r	121	101	222
B2r	115	87	202

Although baerhunter was benchmarked in comparison to other tools, our result showed that its prediction is influenced by data preprocessing as the total number of predicted sRNAs ranged from 202 to 236. While the detection of *cis*-encoded sRNAs was influenced by quality trimming and rRNA removal, only quality trimming affected the identification of *trans*-encoded elements. The predicted *trans*-encoded sRNAs for B1 and B1r and for B2 and B2r were the same. More benchmarking would be needed to reveal the origin of these differences. Nevertheless, it is evident that direct prediction of sRNAs from RNA-Seq data is affected by computational data preprocessing and should be investigated in detail to ensure reliable prediction of non-coding genomic elements in bacteria.

### 3.3 Comparison of Stranded and Non-stranded Data

Because non-stranded RNA-Seq does not preserve information about the orientation of genomic elements producing sequenced transcripts, it cannot be used for the identification of elements that overlap. Thus, only *trans*-encoded sRNAs can be predicted using non-stranded data. Since the pipeline for non-stranded data is a little bit different (see Fig. 1), we recalculated the results for sample B using the pipeline for non-stranded data. The results are summarized in Table 4.

**Table 4.** Numbers of sRNAs predicted by approach for non-stranded RNA-Seq

Sample	A	B	$A \cap B$
X1	76	109	32
X2	75	108	30
X1r	76	109	32
X2r	75	108	30

Computational ribodepletion again did not affect the results. The sensitivity of detection by non-stranded approach was a little bit lower as the numbers of predicted sRNAs in B samples was slightly lower. The detection was not completely the same but very similar when only three sRNAs identified in the non-stranded approach were different

from those detected by the stranded approach in samples B1/B1r and six in samples B2/B2r. If baerhunter predictions of *trans*-encoded sRNA were considered as a reference, the sensitivity (or recall) and precision of the non-stranded approach could have been calculated, see Table 5.

**Table 5.** Precision and recall of approach for non-stranded RNA-Seq

Sample	A		B	
	Precision	Recall	Precision	Recall
X1/X1r	44.7%	28.1%	97.2%	87.6%
X2/X2r	42.7%	27.8%	94.4%	88.7%

Unfortunately, the prediction using non-stranded data from sample A was considerably worse. Not only was the total number of detected sRNAs lower, more than half of predicted loci did not match those predicted using data from sample B. Such a difference between both samples is surprising. Direct detection of sRNAs from RNA-Seq data can only capture those loci that are currently being transcribed [28]. Nevertheless, both samples, A and B, come from the biological replicates taken under the same conditions, and the data were preprocessed in the same manner. Thus, the prediction should be very similar. On the other hand, there is plenty of other parameters that could be responsible for the difference: sequencing depth, preparation of library, or platform used for sequencing, etc.

The only parameter whose influence can be examined computationally is the sequencing depth. Considering the number of mapped reads and their length, sample A contains only half of the sequenced bases in comparison to B. Therefore, we set the high coverage cutoff parameter to 25 for the following detection. This resulted in 180 identified sRNAs for both quality trimming settings. The number of sRNAs that were previously detected by baerhunter was considerably higher, 113 for A1/A1r and 114 for A2/A2r. This means that the resulting recall, 93.4% for A1/A1r and 99.1% for A2/A2r, was even higher than recall for B samples processed by the non-stranded approach. The improvement of precision was lower, resulting in 62.8% for A1/A1r and 63.3% for A2/A2r.

The results showed that non-stranded RNA-Seq can be used for the prediction of *trans*-encoded sRNAs with very high sensitivity, however, the results must be interpreted carefully due to lower precision. Detection by direct processing of RNA-Seq is also heavily influenced by the sequencing depth and detection thresholds must be adjusted according to it. Moreover, the results suggested that thresholds for achieving the same sensitivity in stranded and non-stranded data might be different even if the sequencing depth correction is performed.

## 4 Conclusions

Prediction of small RNAs in bacterial genomes can be performed by several computational as well as laboratory techniques. Direct prediction from standard RNA-Seq data

seems to be advantageous. Unlike fully computational approaches, it brings experimental evidence while recalculating data that are easily obtainable even for non-model bacterial genomes for the simplicity of technique that is widely used to measure expression on a genome-wide scale. Unfortunately, computational tools to perform such predictions are not widely adopted. Although current tools require the utilization of stranded RNA-Seq, we demonstrated that sRNAs can also be identified using non-stranded RNA-Seq with comparable sensitivity to the stranded approach. Nevertheless, only *trans*-encoded sRNAs can be identified. Moreover, we demonstrated that the prediction from non-stranded as well as stranded RNA-Seq is highly influenced by sequencing depth. Since the results depend on a threshold that has to be set up manually in current tools, more benchmarking is needed to ensure reliable and fully automatic prediction of small RNAs in bacterial genomes.

**Acknowledgment.** This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 101023766.

## References

- Ikemura, T., Dahlberg, J.E.: Small ribonucleic acids of *Escherichia coli*. I. Characterization by polyacrylamide gel electrophoresis and fingerprint analysis. *J. Biol. Chem.* **248**, 5024–5032 (1973). [https://doi.org/10.1016/S0021-9258\(19\)43666-1](https://doi.org/10.1016/S0021-9258(19)43666-1)
- Hör, J., Matera, G., Vogel, J., Gottesman, S., Storz, G.: Trans-acting small RNAs and their effects on gene expression in *Escherichia coli* and *Salmonella enterica*. *EcoSal Plus* **9**, (2020). <https://doi.org/10.1128/ecosalplus.esp-0030-2019>
- Bhatt, S., Egan, M., Jenkins, V., Muche, S., El-Fenej, J.: The tip of the iceberg: on the roles of regulatory small RNAs in the virulence of enterohemorrhagic and enteropathogenic *Escherichia coli*. *Front. Cell. Infect. Microbiol.* **6**, (2016). <https://doi.org/10.3389/fcimb.2016.00105>
- Koepfen, K., et al.: A novel mechanism of host-pathogen interaction through sRNA in bacterial outer membrane vesicles. *PLoS Pathog.* **12**, e1005672 (2016). <https://doi.org/10.1371/journal.ppat.1005672>
- Padalon-Brauch, G., et al.: Small RNAs encoded within genetic islands of *Salmonella typhimurium* show host-induced expression and role in virulence. *Nucleic Acids Res.* **36**, 1913–1927 (2008). <https://doi.org/10.1093/nar/gkn050>
- Peng, T., Kan, J., Hu, J., Hu, Z.: Genes and novel sRNAs involved in PAHs degradation in marine bacteria *Rhodococcus* sp. P14 revealed by the genome and transcriptome analysis. *3 Biotech* **10**(3), 1 (2020). <https://doi.org/10.1007/s13205-020-2133-6>
- Lin, Z., et al.: Engineering of the small noncoding RNA (sRNA) DsrA together with the sRNA chaperone Hfq enhances the acid tolerance of *Escherichia coli*. *Appl. Environ. Microbiol.* **87**, 1–15 (2021). <https://doi.org/10.1128/AEM.02923-20>
- Li, W., Ying, X., Lu, Q., Chen, L.: Predicting sRNAs and their targets in bacteria. *Genomics, Proteomics Bioinforma.* **10**, 276–284 (2012). <https://doi.org/10.1016/j.gpb.2012.09.004>
- Huang, H.Y., et al.: sRNAMap: genomic maps for small non-coding RNAs, their regulators and their targets in microbial genomes. *Nucleic Acids Res.* **37**, (2009). <https://doi.org/10.1093/nar/gkn852>

10. Cho, K.H., Kim, J.H.: Cis-encoded non-coding antisense RNAs in streptococci and other low GC Gram (+) bacterial pathogens. *Front. Genet.* **6**, 110 (2015). <https://doi.org/10.3389/fgene.2015.00110>
11. Rath, E.C., Pitman, S., Cho, K.H., Bai, Y.: Identification of streptococcal small RNAs that are putative targets of RNase III through bioinformatics analysis of RNA sequencing data. *BMC Bioinformatics* **18**, 111–120 (2017). <https://doi.org/10.1186/s12859-017-1897-0>
12. Cao, Y., et al.: sRNATarget: a web server for prediction of bacterial sRNA targets. *Bioinformatics* **3**, 364–366 (2009). <https://doi.org/10.6026/97320630003364>
13. Busch, A., Richter, A.S., Backofen, R.: IntaRNA: efficient prediction of bacterial sRNA targets incorporating target site accessibility and seed regions. *Bioinformatics* **24**, 2849–2856 (2008). <https://doi.org/10.1093/bioinformatics/btn544>
14. Eggenhofer, F., Tafer, H., Stadler, P.F., Hofacker, I.L.: RNAPredator: fast accessibility-based prediction of sRNA targets. *Nucleic Acids Res.* **39**, (2011). <https://doi.org/10.1093/nar/gkr467>
15. Saliba, A.E., C Santos, S., Vogel, J.: New RNA-seq approaches for the study of bacterial pathogens. *Curr. Opin. Microbiol.* **35**, 78–87 (2017). <https://doi.org/10.1016/j.mib.2017.01.001>
16. Han, K., Tjaden, B., Lory, S.: GRIL-seq provides a method for identifying direct targets of bacterial small regulatory RNA by in vivo proximity ligation. *Nat. Microbiol.* **2**, 1 (2016). <https://doi.org/10.1038/nmicrobiol.2016.239>
17. Chao, Y., Papenfort, K., Reinhardt, R., Sharma, C.M., Vogel, J.: An atlas of Hfq-bound transcripts reveals 3' UTRs as a genomic reservoir of regulatory small RNAs. *EMBO J.* **31**, 4005–4019 (2012). <https://doi.org/10.1038/emboj.2012.229>
18. Melamed, S., et al.: Global mapping of small RNA-target interactions in bacteria. *Mol. Cell.* **63**, 884–897 (2016). <https://doi.org/10.1016/j.molcel.2016.07.026>
19. Li, W., et al.: RefSeq: expanding the prokaryotic genome annotation pipeline reach with protein family model curation. *Nucleic Acids Res.* **49**, D1020–D1028 (2021). <https://doi.org/10.1093/nar/gkaa1105>
20. Nawrocki, E.P., et al.: Rfam 12.0: Updates to the RNA families database. *Nucleic Acids Res.* **43**, D130–D137 (2015). <https://doi.org/10.1093/nar/gku1063>
21. Nawrocki, E.P., Eddy, S.R.: Infernal 1.1: 100-fold faster RNA homology searches. *Bioinformatics* **29**, 2933–2935 (2013). <https://doi.org/10.1093/bioinformatics/btt509>
22. Zhu, D.Q., Liu, F., Sun, Y., Yang, L.M., Xin, L., Meng, X.C.: Genome-wide identification of small RNAs in *Bifidobacterium animalis* subsp. *lactis* KLDS 2.0603 and their regulation role in the adaption to gastrointestinal environment. *PLoS One.* **10**, e0117373 (2015). <https://doi.org/10.1371/journal.pone.0117373>
23. Kery, M.B., Feldman, M., Livny, J., Tjaden, B.: TargetRNA2: Identifying targets of small regulatory RNAs in bacteria. *Nucleic Acids Res.* **42**, W124–W129 (2014). <https://doi.org/10.1093/nar/gku317>
24. Liu, M., Zhu, Z.T., Tao, X.Y., Wang, F.Q., Wei, D.Z.: RNA-Seq analysis uncovers non-coding small RNA system of *Mycobacterium neoaurum* in the metabolism of sterols to accumulate steroid intermediates. *Microb. Cell Fact.* **15**, 1–17 (2016). <https://doi.org/10.1186/s12934-016-0462-2>
25. Wang, M., et al.: An automated approach for global identification of sRNA-encoding regions in RNA-Seq data from *Mycobacterium tuberculosis*. *Acta Biochim. Biophys. Sin. (Shanghai)* **48**, 544–553 (2016). <https://doi.org/10.1093/abbs/gmw037>
26. Leonard, S., Meyer, S., Lacour, S., Nasser, W., Hommais, F., Reverchon, S.: APERO: a genome-wide approach for identifying bacterial small RNAs from RNA-Seq data. *Nucleic Acids Res.* **47**, e88–e88 (2019). <https://doi.org/10.1093/nar/gkz485>
27. Tjaden, B.: A computational system for identifying operons based on RNA-seq data. *Methods* **176**, 62–70 (2020). <https://doi.org/10.1016/j.ymeth.2019.03.026>

28. Ozuna, A., Liberto, D., Joyce, R.M., Arnvig, K.B., Nobeli, I.: Baerhunter: An R package for the discovery and analysis of expressed non-coding regions in bacterial RNA-seq data. *Bioinformatics* **36**, 966–969 (2020). <https://doi.org/10.1093/bioinformatics/btz643>
29. Sedlar, K., Kolek, J., Skutkova, H., Branska, B., Provaznik, I., Patakova, P.: Complete genome sequence of *Clostridium pasteurianum* NRRL B-598, a non-type strain producing butanol. *J. Biotechnol.* **214**, 113–114 (2015). <https://doi.org/10.1016/j.jbiotec.2015.09.022>
30. Patakova, P., et al.: Transcriptomic studies of solventogenic clostridia, *Clostridium acetobutylicum* and *Clostridium beijerinckii*. *Biotechnol. Adv.* 107889 (2021). <https://doi.org/10.1016/j.biotechadv.2021.107889>
31. Sedlar, K., et al.: A transcriptional response of *Clostridium beijerinckii* NRRL B-598 to a butanol shock. *Biotechnol. Biofuels.* **12**, 1–16 (2019). <https://doi.org/10.1186/s13068-019-1584-7>
32. Sedlar, K., et al.: Transcription profiling of butanol producer *Clostridium beijerinckii* NRRL B-598 using RNA-Seq. *BMC Genomics* **19**, 1–13 (2018). <https://doi.org/10.1186/S12864-018-4805-8/TABLES/4>
33. Bolger, A.M., Lohse, M., Usadel, B.: Trimmomatic: A flexible trimmer for Illumina sequence data. *Bioinformatics* **30**, 2114–2120 (2014). <https://doi.org/10.1093/bioinformatics/btu170>
34. Kopylova, E., Noé, L., Touzet, H.: SortMeRNA: fast and accurate filtering of ribosomal RNAs in metatranscriptomic data. *Bioinformatics* **28**, 3211–3217 (2012). <https://doi.org/10.1093/bioinformatics/bts611>
35. Quast, C., et al.: The SILVA ribosomal RNA gene database project: Improved data processing and web-based tools. *Nucleic Acids Res.* **41**, D590–D596 (2013). <https://doi.org/10.1093/nar/gks1219>
36. Dobin, A., et al.: STAR: ultrafast universal RNA-seq aligner. *Bioinformatics* **29**, 15–21 (2013). <https://doi.org/10.1093/BIOINFORMATICS/BTS635>
37. Ewels, P., Magnusson, M., Lundin, S., Käller, M.: MultiQC: summarize analysis results for multiple tools and samples in a single report. *Bioinformatics* **32**, 3047–3048 (2016). <https://doi.org/10.1093/BIOINFORMATICS/BTW354>
38. Li, H., et al.: The sequence alignment/Map format and SAMtools. *Bioinformatics* **25**, 2078–2079 (2009). <https://doi.org/10.1093/bioinformatics/btp352>
39. Lawrence, M., et al.: Software for computing and annotating genomic ranges. *PLoS Comput. Biol.* **9**, e1003118 (2013). <https://doi.org/10.1371/journal.pcbi.1003118>



# Comparative Study of Synthetic Bulk RNA-Seq Generators

Felitsiya Shakola<sup>1</sup>✉, Dean Palejev<sup>1,2</sup> , and Ivan Ivanov<sup>3</sup> 

<sup>1</sup> GATE Institute, Sofia University, 125 Tsarigradsko Shosse, Bl. 2, 1113 Sofia, Bulgaria  
{felitsiya.shakola, dean.palejev}@gate-ai.eu

<sup>2</sup> Institute of Mathematics and Informatics,

Bulgarian Academy of Sciences, Acad. G. Bonchev St., Bl. 8, 1113 Sofia, Bulgaria

<sup>3</sup> Department of Veterinary Physiology and Pharmacology, Texas A&M University,  
College Station, Texas 77843, USA

ivanov@tamu.edu

**Abstract.** The bulk RNA-seq technology allows researchers to find differentially expressed genes or alternative splicing variants. Multiple methods for investigating these problems have been developed so far, however their performance can only be reliably evaluated on synthetic data or by performing expensive spike-in experiments. Because of the need for such evaluations, various methods for generating synthetic bulk RNA-seq data have been developed. Those methods deploy parametric, semiparametric, or nonparametric approaches to generate datasets based on different input data or parameters. Currently, there is no complete and systematic approach for evaluating different characteristics and performance metrics of such data-generation methods, especially in terms of “closeness” to the original data. In this work, we present an initial framework for comparing different aspects of the data generated by several of the currently available algorithms for synthetic bulk RNA-seq data generation. We demonstrate that there are noticeable differences between those data generators, even in cases when they use the same input data set. We also propose several metrics that could be used as components of a systematic approach for comparative evaluation of algorithms for synthetic bulk RNA-seq data generation.

**Keywords:** RNA-seq · Synthetic data · Comparison

## 1 Introduction

The development of RNA sequencing (RNA-seq) has brought tremendous insight into the dynamics of the transcriptome. Bulk RNA-seq has often outperformed microarrays with its higher signal-to-noise ratio, larger dynamic range of detection, and the ability to uncover and measure a priori unknown genes [1, 2]. It has become a standard technology to detect how gene expression is altered by various experimental conditions. The recent developments in single-cell RNA-seq (scRNA-seq) technologies provide the researchers with the opportunity to explore the transcriptome variation at the level of individual cells [3]. As it is no longer the experiment assumption that tissues are homogenous, researchers



are presented with new opportunities to characterize the cell-to-cell expression variability. However, bulk RNA-seq data is still widely used and is relatively cheaper to obtain compared to the scRNA-seq data [4]. Moreover, it is possible to estimate cell-type proportions (termed decomposition) from bulk RNA-seq data which allows to identify cell population-level associations [5]. The relevance of bulk RNA-seq is also supported by large public databases (dbGAP, GEO) and its common use in translational research [6]. Bulk RNA-seq is still widely used and therefore in this study we cover several methods for synthetically generating this type of data.

The standard bulk RNA-seq experiment consists of two parts - a wet lab and a computational one. The laboratory part includes RNA isolation, messenger RNA (mRNA) selection, ribosomal RNA depletion, complementary DNA (cDNA synthesis), fragmentation, adaptor-ligation to one or both ends and sequencing to a read depth of 5–200 million reads per sample on a high-throughput machine. The computational part incorporates aligning the sequencing reads to a reference transcriptome and/or in some cases *de novo* assembly [7], quantification of the gene expression by counting the reads that overlap transcripts and filtering and normalization between samples and batches. The most common data analysis procedure for bulk RNA-seq is a statistical estimation of the changes in the expression levels of individual genes between sample groups (e.g., tumor versus normal). This is also known as differential expression (DE) analysis and remains the primary and most used application of RNA-seq data.

The increase of the number of methods for analysis of bulk and single-cell RNA-seq data requires tests to assess whether the statistical methodology tools are performing correctly. The used metrics are often false discovery rate and sensitivity [8]. Accuracy tests should not be performed on real datasets due to lack of knowledge of true gene expression levels and expression differences between populations – one can only estimate these parameters. Such accuracy evaluation usually requires costly spike-in experiments. A cost-attractive and common alternative involves simulated data with known generation parameters and a built-in truth, closely recapitulating the characteristics of real data. Various synthetic data generators have been proposed to help with experimental design since the inception of the RNA-seq technology. Some of them mimic alternatively spliced transcript reads and other RNA editing events, [9, 10]. The authors discuss several methods that use FASTA, SAM or BED files as input and aim to evaluate RNA-seq alignment algorithms, mimic the major RNA-seq steps using empirical attributes for introducing approximate experimental biases of the specific sequencing platforms, or estimate transcript expression with or without a reference genome using a *de novo* transcriptome assembler. Such types of methods are not the focus of this review.

Because gene expression in RNA-seq is measured with nonnegative counts and the specific operating characteristics of the sequencing machines, the Poisson distribution was first proposed to model RNA-seq data, e.g. in [11]. However, this model is not widely utilized currently, due to the phenomenon of overdispersion observed in real data, where the biological variation of the read counts among biological replicates could be much larger than the mean whereas in the case of Poisson distribution the mean and the variance are identical. To address this issue, RNA-seq data are sometimes modelled with generalized Poisson distributions - the variance is expressed as a non-constant function of the mean that is always larger than it. Subsequently, the negative binomial distribution

model gained popularity because it captures most of the data overdispersion with just two parameters [12]. A basic implementation of the negative binomial distribution is the *rbinom* function from the built-in stats R package; one could also use the *CountDataSet* function from the *PoiClaClu* R package, having an option to generate 2-class data. Alternative solutions have been proposed in order to better model the global underlying biological variability, including a Poisson-Tweedie distribution [13] and a beta-binomial generalized linear model [14], which have certain limitations, as described in [13].

Given the variety of available software packages that aim to simulate real bulk RNA-seq data, it is important to systematically compare their performance. Such a comparison could guide the research community in selecting the software that is appropriate to generate synthetic data sets for the evaluation of a given statistical analysis method. In this review, we examine a few relatively recent applications for generating synthetic RNA-seq data that take as input either a combination of parameters, a count data set or both. They have been developed around three major goals, the first two being related: 1) DE studies - *compcoder* [8], *SimSeq* [15], *powsimR* [16], *Splatter* [17], *seqgendiff* [9]; 2) Classification studies - *NGSSPPG*, *Splatter*, 3) Correlation studies - *NGSSPPG*, *SPsimSeq* [18]. The synthetic RNA-seq data generators can be further divided into parametric, semiparametric, and nonparametric. The evolution of the nonparametric generators has been presented in [9]. It should be noted that Splat, a simulation method provided in the R package *Splatter*, is aimed specifically at the scRNA-seq data generation. However, it has been also used to model bulk RNA-seq data by disabling its feature for adding dropouts [18]. While there are other recently developed software packages that generate synthetic RNA-seq data and are either based on emulating transcriptional gene regulatory networks [19] or use sparsity constraints to simulate the 16S rDNA-seq metagenomic data processing pipelines [20] we do not consider them in this manuscript. The software packages developed for simulating RNA-seq data and discussed in this review are summarized in Table 1.

**Table 1.** List of generators of synthetic RNA-seq data considered in this article.

Name	Year	Input	Language	Modelling
NGSSPPG [21, 22]	2013	Parameters	C++	Parametric
compcoder [8]	2014	Parameters, can be estimated from read count matrix	R	Parametric
SimSeq [12]	2015	Read count matrix	Java, C, R	Nonparametric
powsimR [16]	2017	Parameters, read count matrix	R	Parametric
Splat [17]	2017	Parameters, can be estimated from read count matrix	R	Parametric
seqgendiff [9]	2020	Read count matrix	R	Nonparametric
SPsimSeq[18]	2020	Read count matrix	R	Semi-parametric

## 2 RNA-Seq Data Generators

In this study, we consider several parametric, semiparametric, and nonparametric methods for generating synthetic RNA-seq data sets.

The R package **compcodeR** [8] was developed for benchmarking methods for differential expression analysis for RNA-seq data. Counts for each gene can be generated from the Negative Binomial or Poisson distribution, with mean and dispersion parameters estimated either from the Pickrell [23] and Cheung [24] data sets or from user-supplied RNA-seq data. The approach for generating synthetic data is described in detail in [25]. The user-defined properties of the dataset include the number of genes and samples, the fraction of differentially expressed genes, their effect size distribution and the inclusion of outlier counts and filter thresholds. **compcodeR** provides an interface to several differential expression analysis methods and a large number of metrics for comparison of the DE results, many of which are general and suitable for test analysis results for other types of data, e.g. microarrays.

**SimSeq** [12] is a nonparametric algorithm implemented as an R package. It does not rely on parametric models for RNA-seq read count generation, as such testing strategy could result, according to the authors, in an overly optimistic view of the performance of an RNA-seq analysis method. **SimSeq** generates read counts with a joint distribution that closely resembles the distribution of a user provided RNA-seq dataset. **SimSeq** subsamples columns from an RNA-seq dataset and swaps individual read counts within genes adjusted by a correction factor in order to create differential expression. The algorithm requires as inputs: an RNA-seq dataset with two independent treatment groups, a vector of normalization factors with an element for each column of the source data, the number of equivalently expressed and DE genes in the simulated count matrix. A modification allows the user to work with source data with a paired treatment design. There is also a feature allowing for the simulation of three or more independent treatment groups from a dataset with two treatment groups. **SimSeq** has the advantage of preserving the source data's original complex gene dependence structure, while parametric simulations often generate data independently for each gene. **SimSeq** also permits the sampling of extreme values from the source data.

**powsimR** [16] can assess power and sample size requirements for detecting differential expression in single cell and bulk RNA-seq data. The synthetic data generation can be determined either by user-specified parameters or by real data. The default sampling distribution in **powsimR** is the negative binomial with the option to choose the zero-inflated negative binomial that is useful when simulating scRNA-seq data. The user can specify the number of genes, the effect sizes, the number of samples per group, their relative sequencing depth and the number of simulations. The count tables can be used directly for differential expression analysis with integrated R-packages (bulk and single cell data: *limma*, *edgeR*, *DESeq2*, *ROTS*, *baySeq*, *DSS*, *NOISeq*, *EBSeg*; specifically single cell data: *MAST*, *scde*, *BPSC*, *scDD*, *monocle*). The package also includes estimation of statistical power.

The **Splat** [17] method is part of the *Splatter* R Bioconductor package that provides an interface to multiple scRNA-seq data simulation methods. There are several scRNA-seq data generators available in *Splatter*, including **Splat**, which captures high

expression outlier genes, differing library sizes between cells, trended gene-wise dispersion, and zero-inflation, typical for real scRNA-Seq data, using parametric distributions with hyper-parameters estimated from real data (gamma-Poisson hierarchical model). The simulation process has two steps: 1) estimation of the simulation parameters from a real dataset, the result of which is a parameters object unique to each simulation model; 2) using the estimated parameters and/or additional user-defined parameters to generate synthetic data. A synthetic dataset can be generated by specifying user-defined parameters without using real data. Splat can also be used for generation of bulk RNA-seq data in [18] by disabling its feature for adding dropouts, specifically designed for scRNA-seq data simulation. Splatter can produce comparisons with multiple metrics between the simulations' output (from different models or generated with different parameters) and real datasets with estimated parameters.

The **seqgendif** [9] package extends the two-group model, most common in differential expression analysis, to arbitrary design matrices. Such design matrices have applications in multi-group RNA-seq experiments, and so the ability to simulate arbitrary designs provides the flexibility to evaluate statistical methods in more complicated scenarios. To keep the unwanted variation, a prespecified signal is added to real RNA-seq datasets, in a process called binomial thinning. Counts are heterogeneously subsampled using the binomial distribution for different individuals to add signal to the observed counts. This procedure can be applied to both single-cell and bulk RNA-seq. The package allows for extending data-based RNA-seq simulation beyond the two-group (finite-group) model. Seqgendiff can be applied in evaluating confounder adjustment approaches, as unobserved confounding, batch effects, surrogate variables, unwanted variation are trending problems for genomics. Other potential applications include evaluating the effects of library size heterogeneity on differential expression analyses and evaluating factor analysis methods.

**SPsimSeq** [18] is a semi-parametric method for generating bulk and single-cell RNA-seq data. Designed to simulate transcriptome data with the aim to preserve the characteristics of real data, it accommodates a wide range of experimental scenarios, including different sample sizes, biological signals (differential expression) and confounding batch effects. SPsimSeq uses the logarithmic counts per millions of reads (log-CPM) values from a real dataset for semi-parametric estimation of gene-wise distributions [26] and the between-genes correlation structure [27]. Datasets of different realistically varying library sizes can be sampled while maintaining the correlation structure between the genes. SPsimSeq simulates differential expression by separately estimating the distributions of the gene expression from the different populations (e.g. treatment groups) in the real dataset, and then sampling a new dataset from each group. SPsimSeq has been demonstrated [18] to simulate data closely resembling the characteristics of real data in terms of variability, distribution of mean expression levels, fraction of zero counts (per gene and sample/cells), relationship between mean and variability, relationship between mean expression and fraction of zero counts and the dependence between genes.

### 3 Methodology, Parameter Selection, and Results

The software packages we selected to evaluate and compare implement either parametric (compcodeR, powsimR, Splat), semiparametric (SPsimSeq), or nonparametric methods (SimSeq, seqgendiff) for generating synthetic RNA-seq data. They were originally designed to use real RNA-seq data as input for their respective algorithms. Therefore, we used the Hunley real RNA-seq data set [28] in our comparative study, Fig. 1. The cohort producing the samples consists of Alzheimer's disease patients (219 samples) and age-matched controls with no neurological disease diagnoses (70 samples). The samples are subjected to a standard polyA-selected Illumina RNA-Seq analysis and sequenced on Illumina HiSeq2500. We filtered out counts with median = 0 and used a subset of 100 samples (50 patients and 50 control samples) to have comparable datasets for all our simulations. Each sample includes 34616 genes. This real data set has two classes and thus, satisfies the input requirement for several of the data simulation packages we consider in our study, Fig. 1.

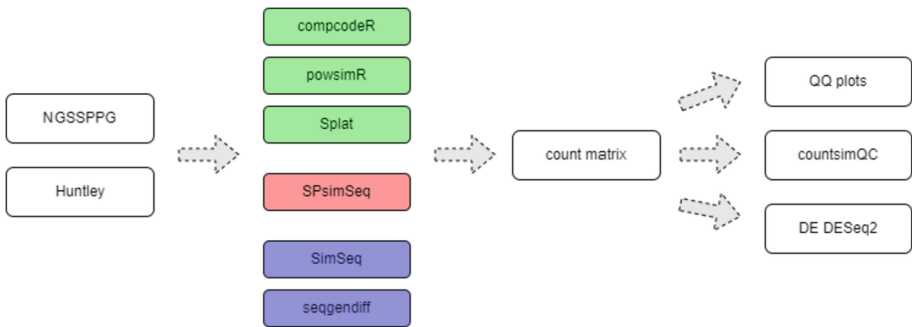
We also used the NGSSPPG package [21, 22] to generate RNA-seq data with a known covariance structure and used it as input for the comparison evaluation, Fig. 1. The NGSSPPG data (100 samples, 10000 genes or features) is 2-group (class ratio = 50/50), generated with two parameter combinations, resulting in different variances: small  $\sigma_0 = 0.4$  (NGSSPPG1) and large  $\sigma_1 = 0.7$  (NGSSPPG2). This synthetic data set has a clearly defined feature-feature block correlation of 0.4 for five of the features. The mean number of reads per feature was adjusted to be equal to 300 by an empirical procedure available in our code (<https://github.com/Felitsiya/Comparative-study-of-synthetic-bulk-RNA-seq-generators>). NGSSPPG package generates data by modelling the two-step process used in the real data sequencing procedure: 1) mRNA concentrations are modelled by a multivariate Gaussian model (MVN-GC); 2) NGS-reads obtained from the libraries produced by the mRNA are modelled by a Poisson process that uses the MVN-GC data as its input. The user-specified parameters in the package are: the number of features, the number of samples per group, the number of subgroups in group 1, the number of global, heterogeneous and non-markers, average expression and average standard deviation for each group, the number of correlated variables in a block structure, the strength of block covariance, the sequencing depth, and the noise. The model also includes a term that represents unknown technical effects associated with an NGS experiment, which follows a Gaussian distribution with zero mean and a modifiable variance.

Our simulation protocol consists of the following steps:

- (i) We used **compcodeR** with parameters calculated from the NGSSPPG data or Huntley data: means of class 1, specific dispersions of class 1 and class 2, as suggested by the package manual and default parameters, including effect size of 1.5 and default *minfact* and *maxfact* to generate individual sequencing depths for the simulated samples.
- (ii) **powsimR** includes a parameter estimation step with recommended settings for bulk RNA-seq data, a simulation setup step and a simulation step with the differential testing method from DESeq2. For the simulation we set the values above  $10^7$  to  $10^7$  in the Huntley data set, in order to avoid the effect of severe outliers.

- (iii) The input of **seqgendiff** was one of the two classes generated by NGSSPPG or the control group of the Huntley data set. The signal was added with an exponential function with rate of 0.5 and added effect size of 1.5.
- (iv) Because the **SimSeq** runs as a one-step process, we used it with its default parameters.
- (v) Because **SPsimSeq** method is a one-step process, where we set the *genewiseCor* parameter to FALSE, thereby not retaining the gene-to-gene correlation structure of the input data due to computational restrictions and possible unreliable estimation for that structure, considering the relatively small sample size of our data sets.
- (vi) **Splat** is a simulation algorithm designed for scRNA data simulation with a parameter estimation step and a simulation step, where we set the *dropout.type* parameter to FALSE, in order for the output data to resemble bulk RNA-seq data, as suggested in [18]. The output data sets have 10000 genes/features in the NGSSPPG input data sets, 34616 genes/features in the Huntley data set, 50 samples in the seqgendiff and SimSeq data sets and 100 samples in the rest of the generated data sets. The 2-group model of the output data is either due to added effect or due to the nature of the input data.

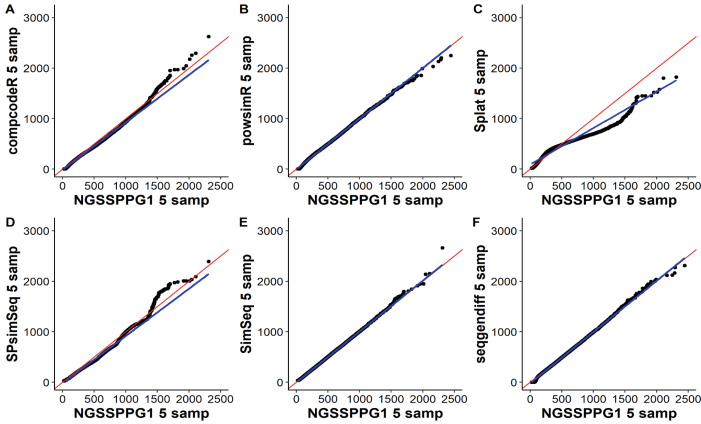
For the comparisons, we used the read count matrices produced by the respective software packages to generate quantile-quantile (Q-Q) plots [29] and descriptive statistics with the help of the R-package *countsimQC* [30], Fig. 1. In addition, the *DESeq2* R package [31] was used to compare the numbers of DE genes, Table 2, in the synthetically generated data to the numbers of DE genes detected in the input data sets.



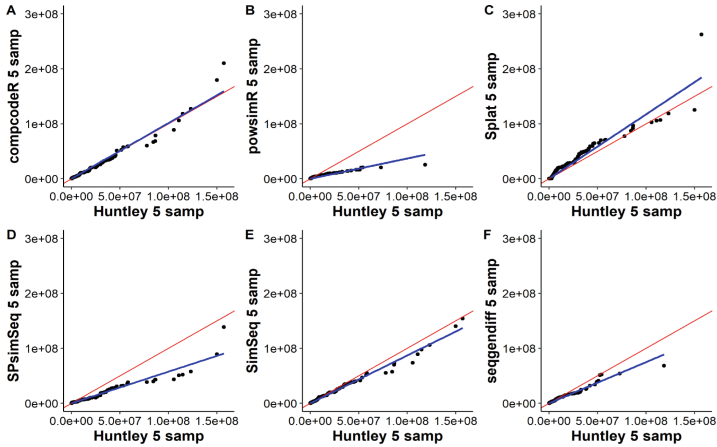
**Fig. 1.** Methodology for comparison. Different colors indicate different types of methods: compcodeR, powsimR and Splat are parametric, SPsimSeq - semiparametric, SimSeq and seqgendiff are nonparametric.

The Q-Q plots illustrate how similar two distributions are. If two data sets are characterised by similar distributions their quantiles should be close to the diagonal. This can be observed for all data sets generated with the NGSSPPG1 data set as input and the NGSSPPG1 data set itself (Fig. 2), except the Splat generated data, which is skewed right (Fig. 2 C). In this case, where the variation of the input data is relatively small, the distributions are close for most of their ranges. However, towards the right-hand

side they do not necessarily match closely. This skewness is even stronger for the larger variance NGSSPPG2 data set (Figure available at: <https://github.com/Felitsiya/Comparative-study-of-synthetic-bulk-RNA-seq-generators>). The closest distribution pair with the NGSSPPG1 and NGSSPPG2 data is the one with SimSeq where the regression line is over the diagonal. In Fig. 3 one can see the Q-Q plots for the simulations based on the real Huntley data. Note the much larger intrinsic variation (nature of real data and over

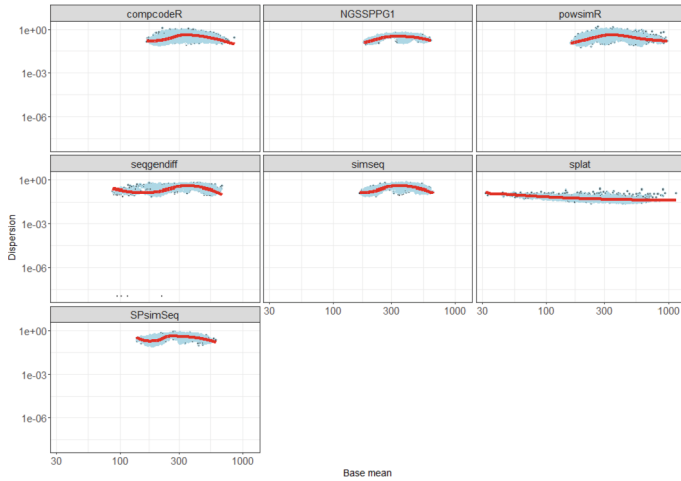


**Fig. 2.** Q-Q plots of five samples generated by the six packages with NGSSPPG1 as input data vs 5 samples of the NGSSPPG1 data set. The samples are from the same class. Each axis represents the quantiles of the respective distribution. Blue: linear regression line. Red: diagonal line. (Color figure online)

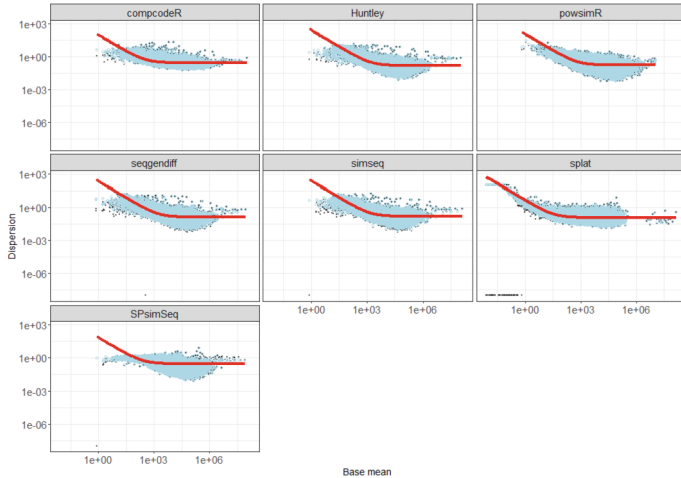


**Fig. 3.** Q-Q plots of five samples generated by the six packages with the Huntley data set as input vs the first 5 samples of the Huntley data set. The samples are from the same class. Each axis represents the quantiles of the respective distribution. Blue: linear regression line. Red: diagonal line. (Color figure online)

three times more features than the NGSSPPG case). The plots suggest that the real data set is best represented and simulated with the `compcoder` package.



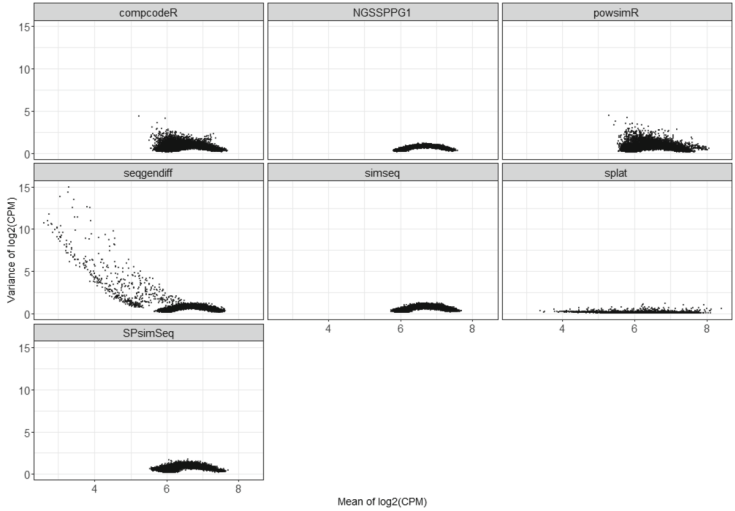
**Fig. 4.** Dispersion/BCV plots of the NGSSPPG1 data set and the data sets generated by the six packages with NGSSPPG1 as input data. Black dots: gene-wise dispersion estimates. Red curve: fitted mean-dispersion relationship. Blue circles: final dispersion estimates. (Color figure online)



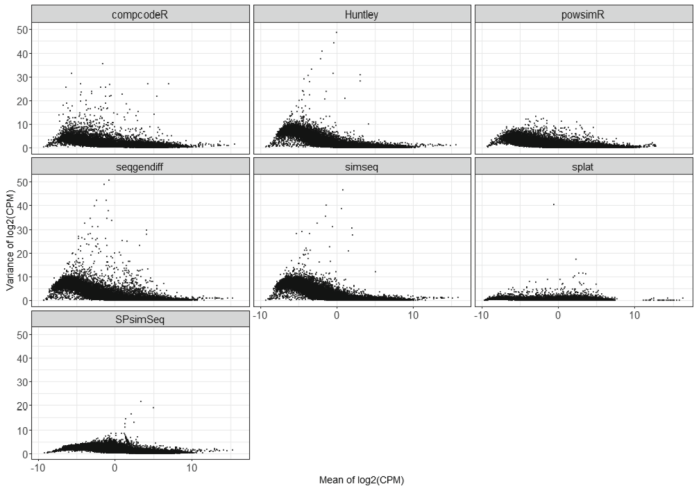
**Fig. 5.** Dispersion/BCV plots of the Huntley data set and the data sets generated by the six packages with the Huntley data set as input. Black dots: gene-wise dispersion estimates. Red curve: fitted mean-dispersion relationship. Blue circles: final dispersion estimates. (Color figure online)

The Dispersion/ Biological Coefficient of Variation (BCV) plots (Fig. 4 and Fig. 5) show the relationship between the dispersion or “biological coefficient of variation” and





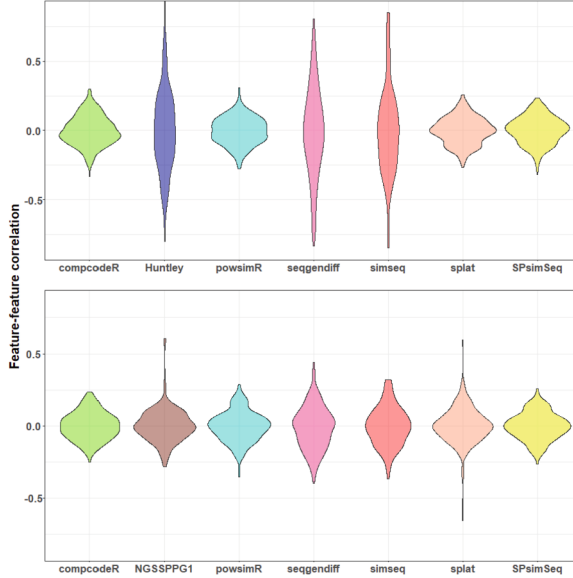
**Fig. 6.** Mean-variance plots of the NGSSPPG1 data set and the data sets generated by the six packages with the NGSSPPG1 data set as input.



**Fig. 7.** Mean-variance plots of the Huntley data set and the data sets generated by the six packages with the Huntley data set as input.

the mean of the  $\log_2$  of the counts per million, calculated by DESeq2. One can notice that the typical for many real data sets curve of the BCV plot (Fig. 5) is not present in NGSSPPG1 data (Fig. 4). This difference could be attributed to the specifics of the NGSSPPG algorithm, which is aimed at data generation for classification studies and generates samples from two classes with a relatively small dispersion. As a result, lower dispersion is also observed in the generated data. Note that the Splat algorithm increases the range of the mean the most.

The mean-variance scatter plots (Fig. 6 and Fig. 7) show how the variance of the features relates to their empirical mean, not considering the information about the experimental design and sample grouping. Note that seqgendiff and SPsimSeq add dispersion to data, regardless of its presence in the input data.



**Fig. 8.** Feature-feature correlation plots. The first plot includes estimations for the NGSSPPG1 data set and the derived form it synthetic data sets; the second plot includes estimations for the Huntley data set and the derived from it synthetic data sets.

The Feature-feature correlation plots display Spearman correlation coefficients' distribution for pairs of features and provides a visualisation of how well the correlation structure of the input data is preserved by the respective generation algorithms. Only non-constant features are considered - when more than 25 such features are present in a data set the pairwise correlations between 25 randomly selected features are displayed [30]. Note that the nonparametric methods SimSeq and seqgendiff outperform the other generation algorithms in capturing the feature-to-feature correlations in the real data set, Huntley (Fig. 8). However, when NGSSPPG was used as input for synthetic data generation, one can observe that the same two algorithms do not perform as well in capturing the feature-to-feature correlations present in the input data.

We also performed testing for DE of the generated data and compared it to the same statistical testing for the two types of input data, Huntley and NGSSPPG. We used the most recent version of the DESeq2 package with an adjusted p value threshold set at 0.05. The DE analysis with DESeq2 reveals many features adjusted for multiple comparison p-values  $< 0.05$  for the Huntley data set, which is due to the intrinsic nature of the tissues the samples are taken from. Similarly, all the synthetic data generators have been set to generate data with 5% DE genes, which is 500 genes for the NGSSPPG-based data sets

and about 1731 genes for the Huntley-based data sets, Table 2. The Splat-generation leads to higher numbers, possibly due to the spread of the counts' means. The DE genes found in the SimSeq data sets are much lower than the expected numbers.

**Table 2.** DE genes found by DESeq2 from data sets simulated either based on either the NGSSPPG1 (100 samples, 10000 features, class ratio = 50/50) data or the Huntley data (100 samples, 34616 features, class ratio = 50/50). All simulation packages are set to generate data with 5% DE genes. Asterisk (\*) denotes the original data set and the subsequent rows indicate data sets generated based on that original data set.

Data set	Number DE genes	Data set	Number DE genes
*NGSSPPG1	623	*Huntley	10066
compcodeR	560	compcodeR	1689
powsimR	543	powsimR	1751
Splat	680	Splat	2210
seqgendiff	573	seqgendiff	1649
SimSeq	77	SimSeq	297
SPsimSeq	629	SPsimSeq	1828

## 4 Discussion

One of the critical issues related to the evaluation of statistical methods for RNA-seq data analyses is to evaluate them on data sets with known properties, e.g. gene-to-gene correlation structure. Therefore, generation of synthetic RNA-seq data sets, which can serve as ground truth for such evaluation, has become a topic of significant interest in the research community. Currently, there are multiple methods for generating synthetic bulk RNA-seq data. One can find many studies comparing the performance of DE methods, however those typically compare the results, e.g. sensitivity, specificity and related performance metrics, and in some cases statistical power. However, little attention has been focused on using a systematic approach for comparing the quality of bulk RNA-seq simulated data. In this paper, we propose an initial approach to address this problem. We highlight several metrics that can be considered when comparing synthetic RNA-seq data while also taking into account the properties present in data used as input for the generation algorithms. The application of the proposed approach is illustrated by using six currently available software packages for synthetic RNA-seq data generation. In our opinion, it is important to include both real data sets (Huntley), and synthetic data (NGSSPPG) with a known structure in the proposed evaluation. Our results clearly show that different data-generation algorithms perform differently with respect to the proposed in this work metrics, underscoring the importance of performing such an evaluation before selecting which specific algorithm should be used to generate synthetic data in order to evaluate a specific data analysis method. While the steps outlined in Fig. 2

describe a general framework for performance evaluation, it is also clear that more work is needed to expand and refine the proposed framework.

**Acknowledgements.** F.S. and D.P. were supported by the GATE project. The project has received funding from the European Union's Horizon 2020 WIDESPREAD-2018–2020 TEAMING Phase 2 programme under Grant Agreement No. 857155 and Operational Programme Science and Education for Smart Growth under Grant Agreement No. BG05M2OP001–1.003–0002–C01.

## References

1. Wang, Z., Gerstein, M., Snyder, M.: RNA-Seq: a revolutionary tool for transcriptomics. *Nat Rev Genet.* **10**, 57–63 (2009). <https://doi.org/10.1038/nrg2484>
2. Ozsolak, F., Milos, P.M.: RNA sequencing: advances, challenges and opportunities. *Nat Rev Genet.* **12**, 87–98 (2011). <https://doi.org/10.1038/nrg2934>
3. Hwang, B., Lee, J.H., Bang, D.: Single-cell RNA sequencing technologies and bioinformatics pipelines. *Exp Mol Med.* **50**, 1–14 (2018). <https://doi.org/10.1038/s12276-018-0071-8>
4. Wang, J., et al.: Gene expression distribution deconvolution in single-cell RNA sequencing. *Proc Natl Acad Sci U S A.* **115**, E6437–E6446 (2018). <https://doi.org/10.1073/pnas.1721085115>
5. Jew, B., et al.: Accurate estimation of cell composition in bulk expression through robust integration of single-cell information. *Nat Commun.* **11**, 1971 (2020). <https://doi.org/10.1038/s41467-020-15816-6>
6. Thind, A.S., et al.: Demystifying emerging bulk RNA-Seq applications: the application and utility of bioinformatic methodology, *Brief Bioinform.* **22**, bbab259 (2021). <https://doi.org/10.1093/bib/bbab259>
7. Hölzer, M., Marz, M.: De novo transcriptome assembly: a comprehensive cross-species comparison of short-read RNA-Seq assemblers, *GigaScience.* **8**, giz039 (2019). <https://doi.org/10.1093/gigascience/giz039>
8. Sonesson, C.: compcodeR—an R package for benchmarking differential expression methods for RNA-seq data. *Bioinformatics* **30**, 2517–2518 (2014). <https://doi.org/10.1093/bioinformatics/btu324>
9. Gerard, D.: Data-based RNA-seq simulations by binomial thinning. *BMC Bioinformatics* **21**, 206 (2020). <https://doi.org/10.1186/s12859-020-3450-9>
10. Zhao, M., Liu, D., Qu, H.: Systematic review of next-generation sequencing simulators: computational tools, features and perspectives, *Brief Funct. Genomics* **16**, 121–128 (2017). <https://doi.org/10.1093/bfgp/elw012>
11. Marioni, J.C., Mason, C.E., Mane, S.M., Stephens, M., Gilad, Y.: RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome Res.* **18**, 1509–1517 (2008). <https://doi.org/10.1101/gr.079558.108>
12. Rigaiil, G., et al.: Synthetic data sets for the identification of key ingredients for RNA-seq differential analysis. *Brief Bioinform.* **19**, 65–76 (2018). <https://doi.org/10.1093/bib/bbw092>
13. Esnaola, M., Puig, P., Gonzalez, D., Castelo, R., Gonzalez, J.R.: A flexible count data model to fit the wide diversity of expression profiles arising from extensively replicated RNA-seq experiments. *BMC Bioinformatics* **14**, 254 (2013). <https://doi.org/10.1186/1471-2105-14-254>
14. Zhou, Y.-H., Xia, K., Wright, F.A.: A powerful and flexible approach to the analysis of RNA sequence count data. *Bioinformatics* **27**, 2672–2678 (2011). <https://doi.org/10.1093/bioinformatics/btr449>

15. Benidt, S., Nettleton, D.: SimSeq: a nonparametric approach to simulation of RNA-sequence datasets. *Bioinformatics* **31**, 2131–2140 (2015). <https://doi.org/10.1093/bioinformatics/btv124>
16. Vieth, B., Ziegenhain, C., Parekh, S., Enard, W., Hellmann, I.: powsimR: power analysis for bulk and single cell RNA-seq experiments. *Bioinformatics* **33**, 3486–3488 (2017). <https://doi.org/10.1093/bioinformatics/btx435>
17. Zappia, L., Phipson, B., Oshlack, A.: Splatter: simulation of single-cell RNA sequencing data. *Genome Biol.* **18**, 174 (2017). <https://doi.org/10.1186/s13059-017-1305-0>
18. Assefa, A.T., Vandesompele, J., Thas, O.: SPsimSeq: semi-parametric simulation of bulk and single-cell RNA-sequencing data. *Bioinformatics* **36**, 3276–3278 (2020). <https://doi.org/10.1093/bioinformatics/btaa105>
19. Grimes, T., Datta, S.: SeqNet: an R package for generating gene-gene networks and simulating RNA-Seq data, *J Stat Softw.* 98 (2021). <https://doi.org/10.18637/jss.v098.i12>
20. Patuzzi, I., Baruzzo, G., Losasso, C., Ricci, A., Di Camillo, B.: metaSPARSim: a 16S rRNA gene sequencing count data simulator. *BMC Bioinformatics* **20**, 416 (2019). <https://doi.org/10.1186/s12859-019-2882-6>
21. Dougherty, E.R., Hua, J., Sima, C.: Performance of feature selection methods. *Curr Genomics.* **10**, 365–374 (2009). <https://doi.org/10.2174/138920209789177629>
22. Ghaffari, N., Yousefi, M.R., Johnson, C.D., Ivanov, I., Dougherty, E.R.: Modeling the next generation sequencing sample processing pipeline for the purposes of classification. *BMC Bioinformatics* **14**, 307 (2013). <https://doi.org/10.1186/1471-2105-14-307>
23. Pickrell, J.K., et al.: Understanding mechanisms underlying human gene expression variation with RNA sequencing. *Nature* **464**, 768–772 (2010). <https://doi.org/10.1038/nature08872>
24. Cheung, V.G., et al.: Polymorphic cis- and trans-regulation of human gene expression. *PLoS Biol.* **8**, e1000480 (2010). <https://doi.org/10.1371/journal.pbio.1000480>
25. Robles, J.A., Qureshi, S.E., Stephen, S.J., Wilson, S.R., Burden, C.J., Taylor, J.M.: Efficient experimental design and analysis strategies for the detection of differential expression using RNA-Sequencing. *BMC Genomics* **13**, 484 (2012). <https://doi.org/10.1186/1471-2164-13-484>
26. Efron, B., Tibshirani, R.: Using specially designed exponential families for density estimation. *The Annals of Statistics* **24** (1996). <https://doi.org/10.1214/aos/1032181161>
27. Hawinkel, S., Mattiello, F., Bijnens, L., Thas, O.: A broken promise: microbiome differential abundance methods do not control the false discovery rate. *Brief. Bioinform.* **20**, 210–221 (2019). <https://doi.org/10.1093/bib/bbx104>
28. Srinivasan, K., et al.: Alzheimer’s patient microglia exhibit enhanced aging and unique transcriptional activation. *Cell Rep.* **31**, 107843 (2020). <https://doi.org/10.1016/j.celrep.2020.107843>
29. Wilk, M.B., Gnanadesikan, R.: Probability plotting methods for the analysis of data. *Biometrika* **55**, 1–17 (1968). <https://doi.org/10.1093/biomet/55.1.1>
30. Soneson, C., Robinson, M.D.: Towards unified quality verification of synthetic count data with countsimQC. *Bioinformatics* **34**, 691–692 (2018). <https://doi.org/10.1093/bioinformatics/btx631>
31. Love, M.I., Huber, W., Anders, S.: Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.* **15**, 550 (2014). <https://doi.org/10.1186/s13059-014-0550-8>



# Investigating Sources of Zeros in 10x Single-Cell RNAseq Data

Hanna Slowik<sup>1</sup>, Joanna Zyla<sup>2</sup> , and Michal Marczyk<sup>2,3</sup>  

<sup>1</sup> Faculty of Automatic Control, Electronics and Computer Science,  
Silesian University of Technology, 44-100 Gliwice, Poland

<sup>2</sup> Department of Data Science and Engineering, Silesian University of Technology,  
44-100 Gliwice, Poland  
michal.marczyk@polsl.pl

<sup>3</sup> Yale Cancer Center, Yale University, New Haven, CT 06511, USA

**Abstract.** Single-cell RNA sequencing allows expression profiling of hundreds of thousands of individual cells in a single experiment. The main drawback is that on the single-cell level observed proportion of zero counts is much higher than on the bulk level. In this study, we performed the analysis of potential sources of excessive zeros using multi-omics data from a homogenous breast cancer cell line. A comparison of the expression data at the population and single-cell level showed that variability between sequencing platforms is higher than when comparing replicates on the same platform. The non-linear model was used to estimate the difference in the expected and observed number of zeros per gene. Then, using gene set enrichment analysis, we discovered some biological pathways containing genes with an increased or reduced number of zeros, like ribosomal genes. Finally, we analyzed different technical factors potentially influencing the dropout rate, and found that the number of transcripts per gene, low mappability and difference in transcript coverage uniformity might cause fluctuations in gene expression estimate on a single-cell level.

**Keywords:** Single-cell sequencing · Transcriptomics · Missing data · Technical artifacts

## 1 Introduction

Single-cell RNA sequencing (scRNAseq) allows characterizing individual cells by measuring the expression levels for each gene within a population. In this way, the method helps to determine what types of cells are present in a heterogeneous sample, which enables a deeper understanding of cell biology and the evaluation of changes that could indicate the presence of the disease [1]. Several methods have been developed for carrying out scRNAseq, each with its own advantages and disadvantages [1]. These high-throughput technologies enable the profiling of hundreds of thousands of cells in parallel. In this work, the droplet-based 10x Genomics Chromium platform, which is based on the capture of single cells by gel beads, was used to obtain the gene expression data

for each cell [2]. Computational analysis of data obtained from scRNAseq is a certain challenge due to factors such as high dimensionality of the data, measurement noise or detection limits.

Appropriate analysis of the scRNAseq data is much more complex than bulk RNAseq data. ScRNAseq counts are naturally more variable than bulk RNAseq counts because the transcriptional signal is not averaged across thousands of individual cells, making cell-to-cell heterogeneity, cell-type mixtures, and stochastic expression bursts important contributors to between-sample variability [3]. Three main technical characteristics of scRNAseq are: (i) sensitivity – the probability to capture and convert a particular mRNA transcript present in a single cell into a cDNA molecule present in the library; (ii) accuracy – how well the read quantification corresponds to the actual concentration of mRNAs; (iii) precision – the technical variation of the quantification [4]. The key task is to accurately separate technical noise from heterogeneity in gene expression levels driven by biological factors.

A well-known characteristic of scRNAseq is the sparsity of the data (zeros), i.e., the high proportion of zero read counts (sometimes called dropout rate) [5]. Some zero expressions may reflect true biological non-expression, when a gene is simply not expressed in the cell or result from gene expression stochasticity (transcriptional bursting). On the other hand, zeros could also occur even when a transcript is expressed in a cell but is entirely undetected in its mRNA profile [6]. The reasons for this type of zeros could be inefficient cDNA polymerization, amplification bias, or low sequencing depth. Other factors that affect gene expression levels include differences in the nucleotide composition, the length of the tested RNA fragments and the differences resulting from variations in the degradation rate of the studied molecules [7]. Separating the biological and technical reasons as sources of zeros is not trivial [8]. Bulk RNAseq can be used as a ground truth assuming that the assayed cell populations are homogeneous.

In this work, we try to understand and quantify which of the known sources of zeros in scRNAseq are the most influential. For that, breast cancer cell line expression data measured on single-cell and bulk levels are used, accompanied by bulk profiles of chromatin openness and DNA methylation profiles. We searched for the biological pathways that contain genes with a higher proportion of zeros than expected and then quantified the effect of technical factors on decreased expression in scRNAseq data.

## 2 Methods

### 2.1 Data

Multi-omics data deposited in Sequence Read Archive under the project accession number PRJNA657088 were used in this study. Experiments performed in the original study involved repeated applications of cytotoxic agents to the *in vitro* cultures of triple-negative breast cancer cell line MDAMB-468 [9]. Available are the measurements of transcriptomic profiles on bulk and single-cell level and additionally different bulk epigenetic profiles (chromatin openness and DNA methylation levels). Only baseline samples before treatment were used here. MDA-MB-231 cells were grown for two parallel sets of identical experiments (biological replicates 1a and 1b). Single-cell RNAseq libraries were constructed using 10× Chromium technology [2]. Bulk RNAseq was done using

TruSeq Stranded Total RNA kit with poly-A selection (Illumina). DNA methylation profiles were obtained using DNA-SeqCap Epi CpGiant Probes (Roche). Chromatin accessibility was measured using ATAC-seq method [10]. All samples were sequenced on Illumina HiSeq4000 platform at Yale Center for Genome Analysis. The same transcriptome annotation files were used across different omics experiments, ENSEMBL release 84, and all sequencing reads were aligned to human reference genome hg38. From 19 797 protein-coding genes included in the analysis, 2 140 had no DNA methylation profile and 6 801 no chromatin openness measurement.

## 2.2 Non-linear Regression Model

To estimate the relationship between dropout rate measured on single-cell level and signal from different bulk modalities, a non-linear regression model with four-parameter logistic curve was used. The model is represented by the following formula:

$$f(x) = c + \frac{d - c}{1 + \exp(b(\log(x) - \log(e)))} \quad (1)$$

where  $d$  is the higher asymptote,  $c$  is the lower asymptote,  $e$  is the value halfway between the asymptotes of  $d$  and  $c$ , and  $b$  is the slope at the inflection point. The parameters of the model were estimated by analysis of the model residuals. Specifically, the Kolmogorov-Smirnov test was run on model residuals, excluding genes with a dropout rate equal to 0 or 1, and  $D$  statistic was calculated. The best parameters of the model were set as the one that minimizes  $D$ .

## 2.3 Gene Set Analysis

Gene Set Enrichment Analysis (GSEA) is a most common algorithm used for differential analysis at the level of gene sets [11]. An ordered list of genes according to their value of residual from the non-linear regression model was used as the input. GSEA uses a permutation-based test with the Kolmogorov-Smirnov running sum statistics to determine enriched expressions in groups of samples. Enrichment of KEGG pathways [12] was tested using GSEA implementation in the `fgsea` R package [13]. Normalized enrichment score (NES) was used as the measure of the effect of the gene set.

## 2.4 Sources of Technical Bias

Based on the literature search, five most probable reasons of observed shift in expression between bulk and single cell-levels were selected. These are: (i) biased reads coverage at 3' end of transcripts, (ii) difference in %GC content of the gene sequence, (iii) length of the gene, (iv) no. of transcripts per gene, (v) lower mappability in a gene region. The non-uniformity of the sequence coverage for each transcript was quantified using Transcript integrity score (TIN) [14]. TIN varies from 0 to 100, where 100 means perfect RNA integrity. For each gene, median TIN was calculated including all transcripts that belongs to this gene. Information about GC content, length of the gene and no. of transcripts per gene was retrieved from ENSEMBL database [15]. The mappability was



calculated as the percentage of gene coverage by single-read map to genomic region which is a fraction of that region that overlaps with at least one unique k-mer ( $k = 100$ ) [16].

## 2.5 Logistic Regression Model

To estimate the most important factors discriminating expressed genes and zeros, a multiple logistic regression model was fitted. In the full model, 8 variables were used including 5 technical factors described above and measurements from 3 modalities. After removing chromatin openness and DNA methylation profiles data (due to many missing values), the model was reduced using the Bayesian Information Criterion. Odds ratio (OR) with 95% confidence interval was used as an effect size of each variable capturing its importance in the model.

## 2.6 Statistical Analysis

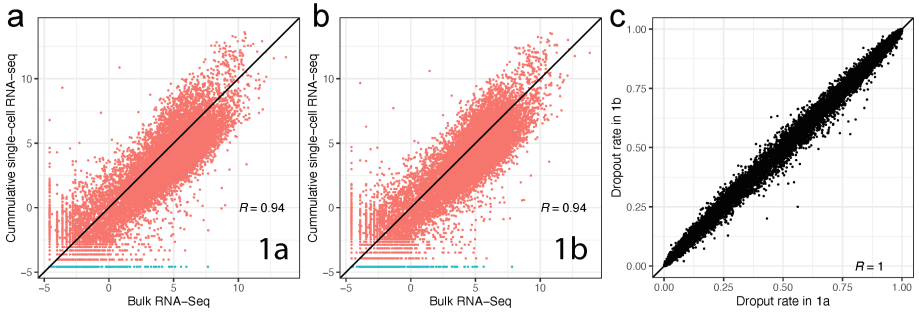
All analyses were performed using non-parametric statistical methods. Two-group comparisons of single variables were performed using the Mann-Whitney U test. The effect size was measured using Cliff's Delta estimate. The significance level was set to 0.05 in all analyses.

# 3 Results and Discussion

From the analyzed dataset, the measurements containing population (bRNAseq) and single-cell (scRNAseq) gene expression data were used, as well as 2 other bulk modalities such as methylation profiles (DNAm) and chromatin openness (ATACseq). The descriptions "1a" and "1b" refer to replicates of cells grown in two identical experiments performed in parallel.

## 3.1 Differences in Gene Expression Between Bulk RNAseq and Cumulative ScRNAseq

First, we compared the measurements of gene expression at the population (bRNAseq) and single-cell (scRNAseq) levels. For that, we needed to sum up counts from all individual cells, creating pseudo-bulk sample (cumulative single-cell RNAseq data). Pearson correlations were computed to evaluate the data similarity. In both replicates, we observed a strong correlation between measurements from two platforms ( $r = 0.94$ ; Fig. 1ab). When comparing two replicates, we found stronger correlation on both levels (on bulk  $r = 0.99$ ; on single-cell  $r = 0.99$ ). Also, the correlation in dropout rate between replicates was very strong ( $r \sim 1$ ; Fig. 1c). Thus, the technological differences are slightly greater than the biological differences between tested samples.



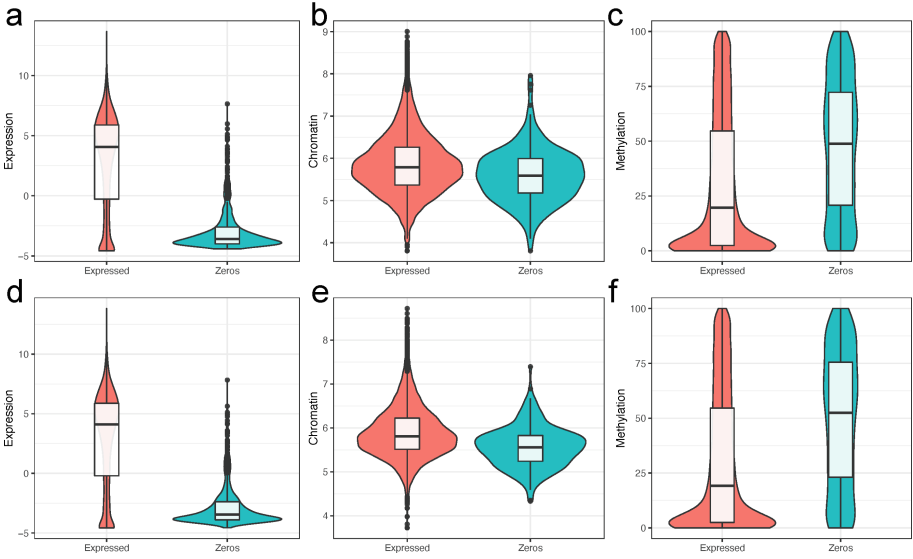
**Fig. 1.** Comparison of expression of individual genes between bulk RNA-seq and summarized single-cell RNA-seq data for two biological replicates (a and b). Panel c shows correlation in dropout rate between replicates

### 3.2 Characterization of Genes with Zeros Using Other Modalities

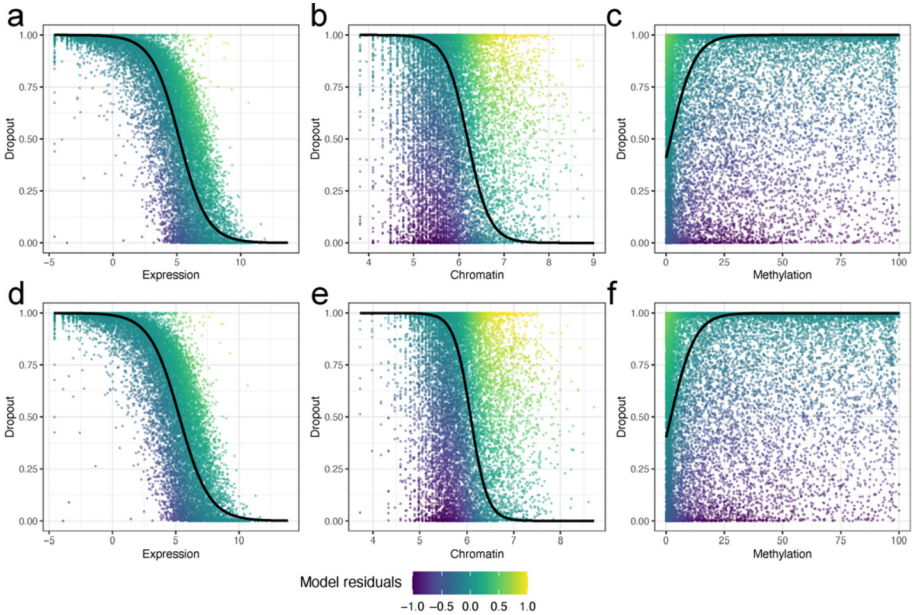
Genes, that have zero counts in bRNAseq in both samples were removed from further analysis, since we assume that they represent the true biological lack of expression ( $n = 16\,608$  genes left). Analysis was performed separately for each replicate. Using 3 modalities, the measurements were compared between expressed genes and zeros (Fig. 2). In the case of bRNAseq and ATACseq, the higher signal was found in expressed genes, while in the case of DNAm, the higher signal was in zeros, which is consistent with current biological knowledge (chromatin openness correlates with higher expression, while increased DNA methylation correlates with decreased expression). The statistical differences between groups of genes were the highest in bRNAseq ( $p = 3.97e-188$  in 1a,  $p = 3.67e-179$  in 1b; Fig. 2ad), moderate in DNAm ( $p = 8.2e-33$  in 1a,  $p = 1.76e-37$  in 1b; Fig. 2cf) and the lowest in ATACseq ( $p = 2.79e-7$  in 1a,  $p = 1.25e-14$  in 1b; Fig. 2be). The effect size calculations showed a strong effect for bRNAseq ( $ES = 0.67$  in both samples) and a small or medium effect for other modalities.

### 3.3 Finding Genes with a Higher Proportion of Zeros Than Expected

For each modality and sample, we fitted the non-linear regression model to quantify the relationship between the dropout rate measured on a single-cell level and bulk level signals (Fig. 3). Similar findings were observed than in the previous section; higher expression and chromatin openness, and lower methylation level relate to lower dropout rate. Model residuals were calculated to quantify how distant is the dropout rate for each gene from the expectations (represented by the model). For each modality, high values of residuals indicate that there is a higher proportion of zeros than the model indicates, while negative values of residuals indicate that there is a lower dropout rate than expected from the model. The best model fit was found for expression data (Fig. 3ad), while the worst was for DNA methylation data (Fig. 3cf).



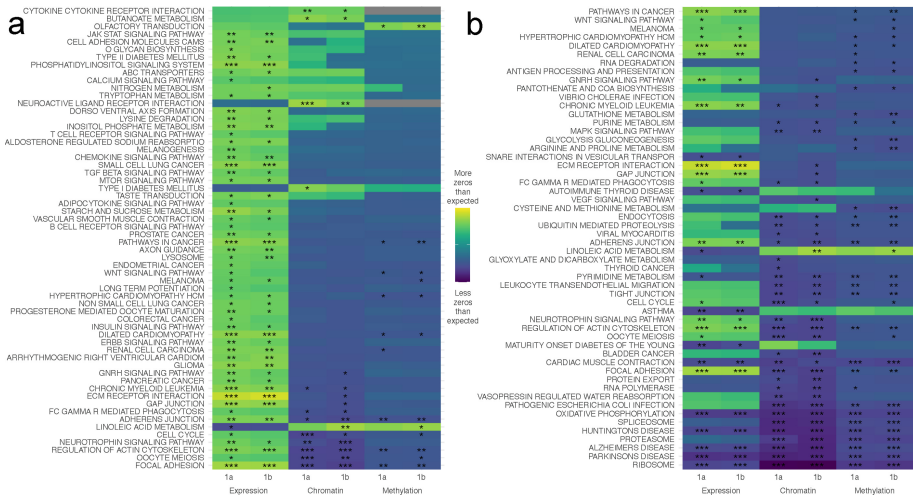
**Fig. 2.** Distribution of signals from different modalities, a bulk gene expression level (a and d), chromatin accessibility (b and e) and methylation level (c and f), between groups of expressed genes and the one with zeros on a single-cell level. The top row shows results from sample 1a, while the bottom from sample 1b.



**Fig. 3.** Modeling relationship between dropout rate and a signal from 3 different modalities: bulk gene expression level (a and d), chromatin accessibility (b and e) and methylation level (c and f). The top row shows results from sample 1a, while the bottom from sample 1b.

### 3.4 Searching of Biological Sources of Decreased Expression on a Single-Cell Level

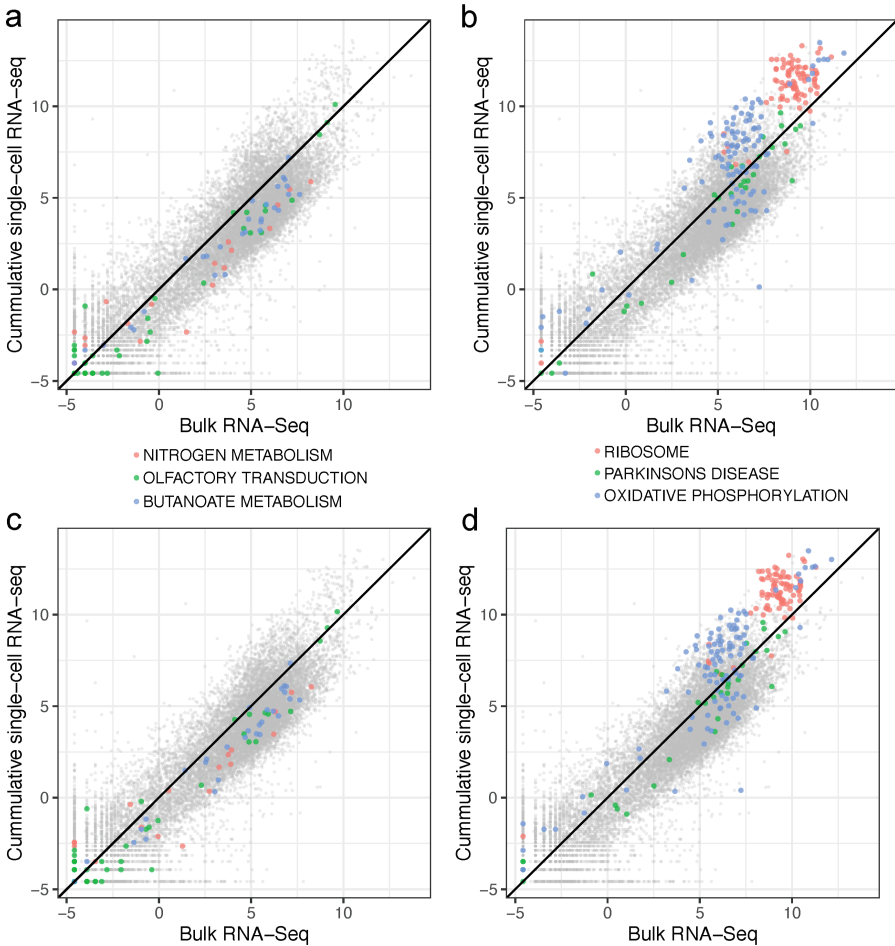
The information obtained from the non-linear regression models was used in further analysis to find biologically related groups of genes for which different modalities indicate that expression should be observed, but it is not seen in scRNAseq data. For that, GSEA was performed on 186 KEGG pathways ( $n = 186$ ) in each sample and modality. Sixty-five pathways showed statistically significant enrichment of positive residuals from the model in at least one comparison, indicating more zeros on a single-cell level than expected (Fig. 4a). Fifty-eight pathways showed the opposite trend (Fig. 4b). Most of the selected pathways had contrasting values of NES between expression and two epigenetic modalities. It means, that even we observe higher expression on a bulk level than on a single-cell level (which correlates with higher dropout rate), closed chromatin and/or high DNA methylation could explain this behavior and we probably observe some artifacts on bulk expression level.



**Fig. 4.** Gene set analysis of model residuals using KEGG pathways. In panel a gene sets with a significant proportion of genes with more zeros than expected are shown, while in panel b gene sets with higher proportion of genes with fewer zeros. \* indicates gene sets with  $p_{adj} < 0.05$ , \*\*  $p_{adj} < 0.01$  and \*\*\*  $p_{adj} < 0.001$ .

The top 3 pathways shown in Fig. 4a with the lowest average NES score across samples and modalities and the top 3 pathways shown in Fig. 4b with the highest average NES score were selected for further analysis (Fig. 5). Nitrogen metabolism, olfactory transduction and butanoate metabolism pathways contain genes that show systematically lower expression on a single-cell level than on bulk level (Fig. 5a). NES values for these pathways were moderate in comparison to others. Within those gene sets, we observe genes with both low and high expression levels, and the findings are consistent in both replicates. Ribosome, Parkinson's disease, and oxidative phosphorylation pathways

contain genes that had higher expression on a single-cell level than on bulk level (Fig. 5b). NES values for these pathways were strongly negative in all modalities. Within these gene sets, we observe more genes with high expression levels. These results suggest that maybe the mechanism of reduced expression of some genes on single-cell level is different to what was expected: some groups of genes, like ribosomal genes, have higher expression on a single-cell level than bulk level, so low expression genes could not be captured due to limited total number of sequencing read counts that are measured in a single experiment.



**Fig. 5.** Visualization of genes included in top3 up- and down-regulated pathways. In panels a and c genes from gene sets with a significant proportion of genes with more zeros than expected are shown, while in panels b and d genes from gene sets with higher proportion of genes with fewer zeros. The top row shows results from sample 1a, while the bottom from sample 1b.

### 3.5 Influence of Technical Factors to Decreased Gene Expression on a Single-Cell Level

Along with the measurements from 3 modalities, 5 technical factors were analyzed to check their influence on excessive zeros. Before modeling, the Spearman correlation between variables was calculated. The highest positive relationship was found between bulk expression and TIN score ( $r = 0.688$ ). Also, we observed a negative correlation between gene length and GC content ( $r = -0.554$ ) and mappability ( $r = -0.439$ ). Thus, these variables might be redundant and only one could be chosen in the stepwise selection procedure of model building.

In the full model, which includes all analyzed variables, TIN score and bulk expression level were significant factors in sample 1a, while TIN, chromatin openness and no. of transcripts in sample 1b. After eliminating less important variables in the stepwise approach, in both samples, the same factors were the most important (no. of transcripts, mappability, TIN score and bulk gene expression level). Since the correlation of TIN and Expression was high, only the importance of expression was highlighted (z-value equal 9.817 in 1a and 7.506 in 1b). Also, high importance was observed for mappability (z-value equal 8.257 in 1a and 7.232 in 1b). This suggests, that selected technical factors also might explain the decrease in gene expression on a single-cell level, however a real low expression level in the analyzed cells/sample is the main contributor (Table 1).

**Table 1.** Relationship between selected technical reasons of zeros and two gene groups (expressed genes vs zeros) measured by odds ratio with 95% CI, calculated using the multiple logistic regression model. Star indicates a significant variable in the model.

Factor	Sample 1a		Sample 1b	
	Full	Reduced	Full	Reduced
GC	1.011 (0.988;1.034)	–	1.014 (0.991;1.037)	–
Gene_length	0.999 (0.998;1)	–	0.999 (0.998;1)	–
N_transcripts	1.029 (0.979;1.078)	1.082 (1.046;1.119)*	1.067 (1.009;1.124)	1.11 (1.071;1.149)*
Mappability	0.978 (0.906;1.05)	1.018 (1.013;1.022)*	1.006 (0.972;1.04)*	1.016 (1.012;1.021)*
TIN	1.039 (1.025;1.05)*	1.011 (1.006;1.017)*	1.062 (1.046;1.078)*	1.017 (1.012;1.023)*
Expression	1.304 (1.168;1.440)*	1.325 (1.269;1.382)*	1.107 (0.978;1.237)	1.243 (1.186;1.3)*
Chromatin	1.119 (0.866;1.371)	–	1.554 (1.224;1.884)*	–
Methylation	1 (0.994;1.006)	–	1.001 (0.996;1.008)	–

## 4 Conclusions

We performed the analysis of potential sources of zeros in single-cell RNA sequencing data. In the first step, a comparison of the expression data at the population and the single-cell level was performed, and we found that differences between sequencing platforms are higher than when comparing replicates on the same platform. Then, the non-linear model was used to estimate the difference in expected and observed number of zeros per gene. Next, based on the gene enrichment analysis, we found some potential biological factors that could indicate why some genes have a higher dropout rate than others. Finally, we analyzed different technical factors potentially influencing excessive zeros rate, and found that no. of transcripts, mappability and transcript coverage uniformity might cause variance in gene expression estimate on a single-cell level.

**Acknowledgments.** This work was financed by the Silesian University of Technology grant no. 02/070/BK22/0033 for maintaining and developing research potential (MM, JZ).

## References

1. Ding, J., et al.: Systematic comparison of single-cell and single-nucleus RNA-sequencing methods. *Nat. Biotechnol.* **38**, 737–746 (2020)
2. Zheng, G.X.Y., et al.: Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.* **8**, 14049 (2017)
3. Buettner, F., et al.: Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. *Nat. Biotechnol.* **33**, 155–160 (2015)
4. Ziegenhain, C., et al.: Comparative analysis of single-cell RNA sequencing methods. *Mol. Cell* **65**, 631–643.e634 (2017)
5. Jiang, R., Sun, T., Song, D., Li, J.J.: Statistics or biology: the zero-inflation controversy about scRNA-seq data. *Genome Biol.* **23**, 31 (2022)
6. Silverman, J.D., Roche, K., Mukherjee, S., David, L.A.: Naught all zeros in sequence count data are the same. *Comput. Struct. Biotechnol. J.* **18**, 2789–2798 (2020)
7. Jaksik, R., Marczyk, M., Polanska, J., Rzeszowska-Wolny, J.: Sources of High variance between probe signals in affymetrix short oligonucleotide microarrays. *Sensors (Basel)* **14**, 532–548 (2013)
8. Van den Berge, K., et al.: Observation weights unlock bulk RNA-seq tools for zero inflation and single-cell applications. *Genome Biol.* **19**, 24 (2018)
9. Marczyk, M., et al.: Multi-omics investigation of innate navitoclax resistance in triple-negative breast cancer cells. *Cancers* **12**, 2551 (2020)
10. Buenrostro, J.D., Wu, B., Chang, H.Y., Greenleaf, W.J.: ATAC-seq: a method for assaying chromatin accessibility genome-wide. *Curr. Protoc. Mol. Biol.* **109**(1), 21.29.1–21.29.9 (2015)
11. Zyla, J., Marczyk, M., Domaszewska, T., Kaufmann, S.H.E., Polanska, J., Weiner, J.: Gene set enrichment for reproducible science: comparison of CERNO and eight other algorithms. *Bioinformatics* **35**, 5146–5154 (2019)
12. Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y., Morishima, K.: KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res.* **45**, D353–D361 (2017)
13. Korotkevich, G., Sukhov, V., Sergushichev, A.: Fast gene set enrichment analysis. *bioRxiv* 060012 (2019)
14. Wang, L., et al.: Measure transcript integrity using RNA-seq data. *BMC Bioinf.* **17**, 58 (2016)
15. Zerbino, D.R., et al.: Ensembl 2018. *Nucleic Acids Res.* **46**, D754–D761 (2018)
16. Karimzadeh, M., Ernst, C., Kundaje, A., Hoffman, M.M.: Umap and Bimap: quantifying genome and methylome mappability. *Nucleic Acids Res.* **46**, e120 (2018)

# **Bioinformatics and Biomarker Identification**





# Exhaled Breath Condensate Study for Biomarkers Discovery

S. Patsiris<sup>1,2</sup>(✉), T. Exarchos<sup>2</sup>, and P. Vlamos<sup>2</sup>

<sup>1</sup> General Hospital of Corfu, Corfu, Greece  
patsiris@hotmail.com

<sup>2</sup> Department of Informatics, Ionian University, Corfu, Greece  
{exarchos,vlamos}@ionio.gr

**Abstract.** Biomarkers seem to play an important role in understanding various diseases' nature, course and management, including respiratory ones. Yet, discovering verifiable and validated ones, that are useful in pulmonology, is challenging and constant. A special body specimen that has been characterized as a matrix of biomarkers, is the exhaled breath condensate (EBC). It is a fluid resulting from freezing the exhaled air. Water is its main constituent. The rest is a rich mix of water-soluble volatile compounds and aerosol droplets of airway lining fluid. The droplets carry non-volatile organic compounds. Their concentration is very small and the techniques applied to measure it are very accurate and sensitive. The content of the exhaled breath condensate reflects important processes taking place in the lungs, such as inflammation and oxidative stress, which are the basis of respiratory diseases' pathophysiology. It seems that it has a role in diagnosis, monitoring, stratification and therapy of respiratory diseases, including COVID19. This paper presents information on exhaled breath condensate and highlights its importance as a potential source of biomarkers.

**Keywords:** Exhaled breath condensate · Biomarkers · Inflammation · Oxidative stress · Non-volatile organic compounds

## 1 Introduction

Biomarkers are an established tool in medical science, providing significant help from diagnosis to management of diseases. There are many different types and groups of markers, possessing specific roles. They serve as indicators of normal and abnormal processes or even as a response to therapy (Vincent, Bogossian and Menozzi, 2020). Their discovery and application may be different in the various medical fields (Pahwa, Sharma and Arora, 2017). Respiratory medicine needs biomarkers. That is due to the inadequacy of the existing techniques and methods to clarify issues regarding the nature and treatment of pulmonary diseases. Different biological sample types are used for that purpose, aiming to find new ones and establish their role (Wu et al., 2018, Sears and Mazzone, 2020). A relatively new specimen in the field of respiratory medicine is the exhaled breath condensate (EBC) which is derived by freezing the exhaled air.

It is a biological fluid full of compounds. Its non-invasive collection has made it an attractive mean to study respiratory diseases, as well (Davis, Fowler and Montpetit, 2019). The purpose of this paper is to provide information regarding both the technique and the sample of the EBC and highlight the reasons for characterizing it as a matrix of biomarkers.

## 2 Exhaled Breath Condensate Definition and Formation

According to the European Respiratory Society, the EBC is a fluid or a frozen material, which can be obtained by cooling the exhaled air (Horváth et al., 2017). It is a type of breath matrix that produces a rich content of numerous compounds. The predominant constituent of the exhaled breath condensate is the water (>99%). The rest is a group of water-soluble volatile compounds and aerosol droplets of airway lining fluid that carry non-volatile compounds. This mix of components ranges from micromolecules to macromolecules such as inorganic ions, urea, organic acids, peptides and proteins (Rahimpour et al., 2018).

The number of the compounds that have been identified in EBC is large (more than 2000) but their concentration is extremely low (Khoubnasabjafari, Rahimpour and Jouyban, 2018). The daily volume of water released with breath in the form of vapour is 350 ml, which entrains the soluble volatile compounds. Apart from that, there are aerosol particles accompanying the exhaled air and they are of variable amount. Their average size is 0.3  $\mu\text{m}$  and their levels in a normal breath range from 0.1 to 4 particles  $\times \text{cm}^{-3}$ . Moreover, it is estimated that <0.1  $\mu\text{l}$  of the exhaled droplets is contained in 1 ml of exhaled breath condensate (Hayes et al., 2016, Konstantinidi et al., 2015).

This mix of volatile and non-volatile compounds originates from the whole respiratory tract. The mechanisms responsible for the presence of these compounds in the breath vary (Maniscalco et al., 2019). Volatile organic compounds travel through the airways and the condensing breath absorbs them. On the other hand, the origin of the aerosol droplets or exhaled particles is the airway lining layer. Two different theories can explain their generation a) the turbulence of the airflow able to shear the airway lining fluid and b) the bronchiole fluid film burst model. According to the latter, films or bubbles of respiratory fluid are formed and their burst that happens following the closure and reopening of the bronchioles is responsible for the droplets (Kazeminasab et al., 2020a).

## 3 Exhaled Breath Condensate Collection

The collection of the EBC is performed with a variety of apparatus. There are commercial devices and homemade systems. All of them follow the same principle which is to freeze the exhaled air. Based on that, they have main common parts, the chamber or condenser, where the EBC is formed and tubes with valves that guide the exhaled air into the condenser. The valves are used to prevent the mix of the ambient air with the exhaled one. An additional part is saliva traps, which are used to prevent the contamination of the exhaled air with saliva.

In the case of homemade systems, a detailed report of the structure and the materials used for their construction is necessary. Both of them can affect the concentration of the compounds in the sample (6). On the other hand, there are the available standardized commercial devices for the collection of the EBC, which are the Rtube, the EcoScreen, the Turbo-deccs and the Anacon Glass Condenser. Different and varied materials have been used for their construction. Moreover, each one has its characteristics regarding the structure of the device. There are guidelines of performance based on which one is used and deal with the sampling duration, the temperature of condenser and the device cleaning and possible reuse process (Kubán and Foret, 2013, Połomska, Bar and Sozanska, 2021).

## 4 Exhaled Breath Condensate Analysis

The analysis of the EBC is also performed with a variety of techniques. These can be grouped as optical (absorbance, fluorescence, chemiluminescence), separation (chromatography: ion, liquid and gas, electrophoresis: capillary and gel), electrochemical, ion mobility mass spectrometry, polymerase chain reaction-based assays, nanoparticle based, surface acoustic wave immunosensors, nuclear magnetic resonance-based metabolomic analyses and high electron mobility transistors. Each one has its strengths and weaknesses, especially regarding its sensitivity and specificity. None of the current single techniques is the gold standard for the analysis of the exhaled breath condensate because of the physiochemical properties of the compounds. The ones, that are widely used, are the nuclear magnetic resonance-based metabolomic analyses and mass spectrometry in combination with the separation ones (Khoubnasabjafari et al., 2021, Wallace and Pleil, 2018).

## 5 Exhaled Breath Condensate Application

The EBC study seems to be capable to provide information on the nature, course and treatment of diseases, especially the respiratory ones. Its application is wide and it has been used for diagnostic reasons and early detection of diseases, differential diagnosis, stratification, screening and monitoring the severity of diseases as well as the efficacy of pharmacotherapy (Chen et al., 2021).

This wide application relies on the advantages the EBC has. It is an entirely non-invasive technique and simple to perform. It can be repeated several times without causing discomfort or adverse effects. It has no age restriction and is well tolerated. It can be collected at different stages of a condition including the mechanically ventilated patient. It does not require any special skills of either the staff or the patient for its collection as well (Davis and Montpetit, 2018).

However, it has not yet been established as a clinical tool because of its lack of standardization. That is because of the variety of devices and techniques used for its collection and analysis. Apart from the technical issues, its high dilution is a crucial remaining problem that contributes to its weakness and does not allow its usage in the clinical setting (Peterová et al., 2018).

## 6 Exhaled Breath Condensate and Biomarkers

A biomarker is a 'defined characteristic that is measured as an indicator of normal biological processes, pathogenic processes or responses to an exposure or intervention' (Califf, 2018). The exhaled breath condensate is a specimen of fluid that is considered to contain biomarkers. The rationale for that relies on its characteristics and advantages.

The content of the exhaled breath condensate has components that are related to the main pathogenic processes taking place in the lungs such as inflammation, oxidative and nitrate stress. These components are not cells but compounds of the airway epithelial lining fluid as a result of the inflammatory response to various stimuli. The airway epithelial lining fluid is an aqueous layer that lines the respiratory tract and shows heterogeneity because it has a different cell population in various regions of the lungs (Youssef et al., 2016, Pouwels et al., 2021). It is a barrier and one of its functions is the defense which is expressed by inflammation (De Rose et al., 2018). The compounds from the airway epithelial lining fluid provide information on different patterns of inflammation in the lungs.

The components of the EBC are a rich mix of volatile and non-volatile compounds. Their variety, concentration and roles make them candidates for being biomarkers because they alter the breath profile. They are mediators of inflammation and markers of oxidative and nitrate stress. Their origin varies and involves different pathways. Some of them derive from the free radical catalyzed peroxidation of arachidonic acid (eicosanoids, prostanoids) while others from the nitric oxide (nitrite, nitrate, S-Nitrosothiols, 3-Nitrotyrosine) through the reaction of the amino acid L-arginine and the enzyme nitric oxide synthase. Their role varies and they seem to be responsible for the regulation and balance of inflammation, oxidative and nitrate stress. The increased levels of them in the exhaled breath condensate may reflect the underlying inflammation of the airways and the severity of the condition. Moreover, processes such as mucus production, bronchoconstriction and recruitment of different types of cells are affected by increased levels of some of the arachidonic acid derivatives (Lazar et al. 2018). Other components that have been detected in the EBC are proteins (cytokines, chemokines), some enzymes and metals. Cytokines possess a central inflammatory and immune role as they can either promote or inhibit inflammatory reactions (Hatami et al. 2019). Smooth muscle contraction may also be affected by metals found in it and some enzymes are involved in the remodelling process of the airways (Ghio, Madden and Esther, 2018). Nucleic acids (DNA and RNA) have been discovered in the EBC as well. Their presence in it is due to cell apoptosis and necrosis as well as cell death in the lungs as a result of the process of oxidative stress. The nucleic acids provide information of alterations in DNA gene sequence and the genes' expressions (Kazeminasab et al., 2020b, Pérez-Sánchez et al., 2021). In the case of cancer, they possibly provide information on the process of tumorigenesis (Kazeminasab et al., 2021). Additional findings in the EBC are viruses, bacteria and fungi. Even though COVID-19 is a new disease, studies have shown that it is detectable in exhaled breath condensate. That is because of the transmission of the coronavirus via respiratory droplets and aerosols as well as the alteration of the content of the exhaled breath condensate of patients with COVID-19 compared to healthy subjects (Khoubnasabjafari et al., 2020, Giovannini, Haick and Garoli, 2021, Barberis et al., 2021).

The performance of collection of the EBC is also an argument that contributes to the study of biomarkers. It is a simple direct sampling task as it requires only tidal breathing for its collection. This type of breathing is considered enough for obtaining the appropriate amount of EBC. The airflow carries particles of all sizes ranging from submicron to large ones (Bake et al., 2019). These particles have different sites of origin including the lower respiratory tract and small and medium – size airways. There are indications that alveoli are a source of some of them, as well (Finamore et al., 2019).

The non-invasive character of the technique provides a sample that is not affected by external interventions. Furthermore, the process of the collection does not influence the function of the airways nor the pathological processes taking place in them. That means there is no alteration in its content and the result reflects the state and the actions in the lungs. This is also enhanced by three technical features a) its application is feasible at any age group (Urs et al., 2021), b) the status of a disease does not prevent its collection and c) it is available for collection several times easily. As a result, a great variety and number of markers is obtained, leading to the identification of different profiles full of information of a condition or a disease (Campanella, De Summa and Tommasi, 2019).

## 7 Summary

The study of biomarkers in medicine is constant. A relatively new specimen the exhaled breath condensate has been characterized as a promising source of biomarkers of respiratory diseases. Many arguments support this opinion and they are based on its nature as well as the process of its collection and analysis. It has a rich content of compounds deriving from the whole respiratory tract. It is capable to provide information on different aspects of physiological and pathological processes happening in the lungs. Its collection is simple and non-invasive leading to a product without external interventions. The fact that it is not limited by age or disease status provides the chance to obtain more information and markers of various stages. However, it has not established its role in the clinical setting because of the lack of standardization. Further research to reveal its whole capacity is necessary.

## References





- Bake, B., Larsson, P., Ljungkvist, G., Ljungström, E., Olin, A.-C.: Exhaled particles and small airways. *Respir. Res.* **20**(1), 8 (2019). <https://doi.org/10.1186/s12931-019-0970-9>
- Barberis, E., et al.: Metabolomics diagnosis of COVID-19 from exhaled breath condensate. *Metabolites* **11**, 847 (2021). <https://doi.org/10.3390/metabo11120847>
- Califf, R.M.: Biomarker definitions and their application. *Exp. Biol. Med.* **243**, 213–221 (2018). <https://doi.org/10.1177/1535370217750088>
- Campanella, A., De Summa, S., Tommasi, S.: Exhaled breath condensate biomarkers for lung cancer. *J. Breath Res.* **13**(4), 044002 (2019). <https://doi.org/10.1088/1752-7163/ab2f9f>
- Chen, T., Liu, T., Li, T., Zhao, H., Chen, Q.: Exhaled breath analysis in disease detection. *Clin. Chim. Acta* **515**, 61–72 (2021). <https://doi.org/10.1016/j.cca.2020.12.036>
- Davis, M.D., Montpetit, A.J.: Exhaled breath condensate. An update. *Immunol. Allergy Clin. North Am.* **38**, 667–678 (2018). <https://doi.org/10.1016/j.iac.2018.06.002>

- Davis, M.D., Fowler, S.J., Montpetit, A.J.: Exhaled breath testing – A tool for the clinical and researcher. *Paediatr. Respir. Rev.* **29**, 37–41 (2019). <https://doi.org/10.1016/j.prrv.2018.05.002>
- De Rose, V., Molloy, K., Gohy, S., Pilette, C., Greene, C.M.: Airway epithelium dysfunction in cystic fibrosis and COPD. *Mediators Inflamm.* **2018**, 1309746 (2018). <https://doi.org/10.1155/2018/1309746>
- Finamore, P., Scarlata, S., Cardaci, V., Incalzi, R.A.: Exhaled breath analysis in obstructive sleep apnea syndrome: a review of the literature. *Medicina (Kaunas)* **55**(9), 538 (2019). <https://doi.org/10.3390/medicina55090538>
- Ghio, A.J., Madden, M.C., Esther, C.R.: Transition and post-transition metals in exhaled breath condensate. *J. Breath Res.* **12**(2), 027112 (2018). <https://doi.org/10.1088/1752-7163/aaa214>
- Giovannini, G., Haick, H., Garoli, D.: Detecting COVID-19 from breath: a game changer for a big challenge. *ACS Sensors* **6**, 1408–1417 (2021). <https://doi.org/10.1021/acssensors.1c00312>
- Hatami, H., Ghaffari, N., Ghaffari, J., Rafatpanah, H.: Role of cytokines and chemokines in the outcome of children with severe asthma: narrative review. *J. Pediatr. Rev.* **7**(1), 17–28 (2019). <https://doi.org/10.32598/jpr.7.1.17>
- Hayes, S.A., et al.: Exhaled breath condensate for lung cancer protein analysis: a review of methods and biomarkers. *J. Breath Res.* **10**(3), 034001 (2016). <https://doi.org/10.1088/1752-7155/10/3/034001>
- Horváth, I., et al.: A European Respiratory Society technical standard: exhaled biomarkers in lung disease. *Eur. Respir. J.* **49**, 1600965 (2017). <https://doi.org/10.1183/13993003.00965-2016>
- Kazeminasab, S., Emamalizadeh, B., Jouyban, A., Shoja, M.M., Khoubnasabjafari, M.: Macromolecular biomarkers of chronic obstructive pulmonary disease in exhaled breath condensate. *Biomark. Med.* **14**(11), 1047–1063 (2020). <https://doi.org/10.2217/bmm-2020-0121>
- Kazeminasab, S., Emamalizadeh, B., Jouyban-Gharamaleki, V., Taghizadieh, A., Khoubnasabjafari, M., Jouyban, J.: Tips for improving the quality and quantity of the extracted DNA from exhaled breath condensate samples. *Nucleos. Nucleot. Nucl. Acids* **39**(5), 688–698 (2020). <https://doi.org/10.1080/15257770.2019.1677910>
- Kazeminasab, S., Emamalizadeh, B., Khoubnasabjafari, M., Jouyban, A.: Exhaled breath condensate: a non-invasive source for tracking of genetic and epigenetic alterations in lung diseases. *Pharmaceut. Sci.* **27**(2), 149–161 (2021). <https://doi.org/10.34172/PS.2020.46>
- Khoubnasabjafari, M., Rahimpour, E., Jouyban, A.: Exhaled breath condensate as an alternative sample for drug monitoring. *Bioanalysis* **10**(2), 61–64 (2018). <https://doi.org/10.4155/bio-2017-0205>
- Khoubnasabjafari, M., Jouyban-Gharamaleki, V., Ghanbari, R., Jouyban, A.: Exhaled breath condensate as a potential specimen for diagnosing COVID-19. *Bioanalysis* **12**(17), 1195–1197 (2020). <https://doi.org/10.4155/bio-2020-0083>
- Khoubnasabjafari, M., et al.: Breathomics: review of sample collection and analysis, data modeling and clinical applications. *Crit. Rev. Anal. Chem.* 1–17 (2021). <https://doi.org/10.1080/10408347.2021.1889961>
- Konstantinidi, E.M., Lappas, A.S., Tzortzi, A.S., Behrakis, P.K.: Exhaled Breath Condensate: Technical and Diagnostic Aspects. *Sci. World J.* 435160 (2015). <https://doi.org/10.1155/2015/435160>
- Kubán, P., Foret, F.: Exhaled breath condensate: determination of non-volatile compounds and their potential for clinical diagnosis and monitoring. A review. *Anal. Chim. Acta* **805**, 1–18 (2013). <https://doi.org/10.1016/j.aca.2013.07.049>
- Lazar, Z., Horvath, I., Vestbo, J., Bikov, A.: Exhaled breath condensate in chronic obstructive pulmonary disease: methodological challenges and clinical application. *Minerva Pneumol.* **57**(2), 42–56 (2018). <https://doi.org/10.23736/S0026-4954.18.01816-3>
- Maniscalco, M., Fuschillo, S., Paris, D., Cutignano, A., Sanduzzi, A., Motta, A.: Clinical metabolomics of exhaled breath condensate in chronic respiratory diseases. *Adv. Clin. Chem.* **88**, 121–149 (2019). <https://doi.org/10.1016/bs.acc.2018.10.002>

- Pahwa, S., Sharma, V., Arora, V.: Biomarkers – its role in medicine. *Int. J. Pharm. Sci. Res.* **8**(7), 2776–2788 (2017). [https://doi.org/10.13040/IJPSR.0975-8232.8\(7\).2776-88](https://doi.org/10.13040/IJPSR.0975-8232.8(7).2776-88)
- Peterová, E., et al.: Exhaled breath condensate: pilot study of the method and initial experience in healthy subjects. *Acta Medica* **61**(1), 8–16 (2018). <https://doi.org/10.14712/18059694.2018.17>
- Pérez-Sánchez, C., et al.: Clinical utility of microRNAs in exhaled breath condensate as biomarkers for lung cancer. *J. Pers. Med.* **11**(2), 111 (2021). <https://doi.org/10.3390/jpm11020111>
- Połomska, J., Bar, K., Sozanska, B.: Exhaled breath condensate—a non-invasive approach for diagnostic methods in asthma. *J. Clin. Med.* **10**(12), 2697 (2021). <https://doi.org/10.3390/jcm10122697>
- Pouwels, S.D., Burgess, J.K., Verschuuren, E., Slebos, D.J.: The cellular composition of the lung lining fluid gradually changes from bronchus to alveolus. *Respir. Res.* **22**(1), 285 (2021). <https://doi.org/10.1186/s12931-021-01882-x>
- Rahimpour, E., Khoubnasabjafari, M., Jouyban-Gharamaleki, V., Jouyban, A.: Non-volatile compounds in exhaled breath condensate: review of methodological aspects. *Anal. Bioanal. Chem.* **410**(25), 6411–6440 (2018). <https://doi.org/10.1007/s00216-018-1259-4>
- Sears, C.R., Mazzone, P.J.: Biomarkers in lung cancer. *Clin. Chest Med.* **41**(1), 115–127 (2020). <https://doi.org/10.1016/j.ccm.2019.10.004>
- Urs, R., Stoecklin, B., Pillow, J.J., Hartmann, B., Hall, G.L., Simpson, S.J.: Collecting exhaled breath condensate from non-ventilated preterm-born infants: a modified method. *Pediatr. Res.* (2021). <https://doi.org/10.1038/s41390-021-01474-x>
- Vincent, J.L., Bogossian, E., Menozzi, M.: The future of biomarkers. *Crit. Care Clin.* **36**(1), 177–187 (2020). <https://doi.org/10.1016/j.ccc.2019.08.014>
- Wallace, M.A.G., Pleil, J.D.: Evolution of clinical and environmental health applications of exhaled breath research: review of methods: instrumentation for gas-phase, condensate, and aerosols. *Anal. Chim. Acta* **1024**, 18–38 (2018). <https://doi.org/10.1016/j.aca.2018.01.069>
- Wu, A.C., et al.: Current status and future opportunities in lung precision medicine research with a focus on biomarkers. an American Thoracic Society/National Heart, Lung, and Blood Institute research statement. *Am. J. Respir. Crit. Care Med.* **198**(12), e116–e136 (2018). <https://doi.org/10.1164/rccm.201810-1895ST>
- Youssef, O., Sarhadi, V.K., Armengo, G., Piirilä, P., Knuutila, A., Knuutila, S.: Exhaled breath condensate as a source of biomarkers for lung carcinomas. A focus on genetic and epigenetic markers – a mini-review. *Genes Chromosom. Cancer* **55**(12), 905–914 (2016). <https://doi.org/10.1002/gcc.22399>



# Statistical Learning Analysis of Thyroid Cancer Microarray Data

Iván Petrini<sup>2</sup>, Rocío L. Cecchini<sup>1,2</sup> , Marilina Mascaró<sup>3</sup> ,  
Ignacio Ponzoni<sup>1,2</sup> , and Jessica A. Carballido<sup>1,2</sup> 

<sup>1</sup> Institute for Computer Science and Engineering (UNS-CONICET),  
Bahía Blanca, Argentina  
[ip@cs.uns.edu.ar](mailto:ip@cs.uns.edu.ar)

<sup>2</sup> Department of Computer Science and Engineering, Universidad Nacional del Sur,  
Bahía Blanca, Argentina

<sup>3</sup> Cancer Biology Laboratory (UNS-CONICET), Universidad Nacional del Sur,  
Bahía Blanca, Argentina  
<https://icic.conicet.gov.ar/>, <https://cs.uns.edu.ar/>

**Abstract.** The classification of human cancers constitutes to date a significant challenge in the context of microarray data analysis. The discovery of gene hallmarks for biological processes involves the examination of large gene expression matrices in a broad and massively parallel manner. In this article, a comprehensive and comparative analysis of thyroid cancer datasets is presented, including stages for feature selection, hypothesis testing, and classification. Also, datasets are integrated, and results for this integration are reported and analyzed. To conclude, text mining is used to investigate some biological information regarding the main resulting characteristic genes. Some genes found during the research, HINT3 in particular, appear to be worth to be further studied.

**Keywords:** Microarray · Classification · Feature selection · Cancer · Statistical learning

## 1 Introduction

Computer science applied to problems in biology is so relevant that it gave birth to a rising branch of research: bioinformatics. In this new discipline, machine learning and statistical-based inference methods play a central role. This allows the discovery of new knowledge in molecular biology [5, 13], including in the context of covid-19 pandemic [8]. However, we must not lose sight of the significance of statistics. It is thus mandatory to become almost an expert in hypothesis testing to perform a robust examination of biological data. A particularly challenging problem is the discovery of relations between gene expression at the molecular level and the phenotype. The activity of many genes can be determined by the measurement of mRNA levels using multiple techniques, including

Supported by CONICET (112-2017-0100829) and SGCyT-UNS (24/N052).

© Springer Nature Switzerland AG 2022

I. Rojas et al. (Eds.): IWBBIO 2022, LNBI 13347, pp. 90–102, 2022.

[https://doi.org/10.1007/978-3-031-07802-6\\_8](https://doi.org/10.1007/978-3-031-07802-6_8)



DNA microarray [3,4], Serial Analysis of Gene Expression [20,22], and various in situ hybridization applications [26]. In this context, microarray has emerged as a precise, productive, and efficient tool for measuring specific gene expression levels.

The goal is to use microarray data for gene selection and classification. In this context, machine learning constitutes a powerful tool, and its applications extend across many scientific disciplines. Specifically, feature selection, classification, and grouping algorithms have emerged as significant helpers to identify strong correlations, associations, and separations in large data sets, especially for gene expression data. Regarding the case of study, thyroid cancer is the most common malignant endocrine disease, and year after year, its incidence worldwide increases substantially [16,17,19]. Papillary Thyroid Cancer (PTC) is the most common type of thyroid cancer, constituting between 70% and 80% of all cases of thyroid cancer. Anaplastic Thyroid Cancer (ATC) is a rare but highly aggressive type of cancer that accounts for only 1–2% of all thyroid cancer cases. Nonetheless, more than 50% of thyroid cancer deaths are related to ATC, and the mean survival rate is six months. Then, as a value-added contribution of this work, the knowledge of molecular mechanisms underlying ATC may provide new potential therapeutic targets in its treatment. In this context, a comprehensive study of microarray datasets obtained from different thyroid cancer experiments is presented in this paper. Theoretical and practical procedures are briefly explained and compared, covering feature selection, classification, and description of biological relevance. Also, multiclass problems are considered, and the integration of datasets is analyzed.

## 2 Methods

Machine learning consists of algorithms that iteratively learn from input data to improve their performance, describe data, and predict results. There are different categories in which machine learning algorithms can be categorized [11]. Depending on the objective, on the nature of the problem being treated, and on the type and volume of the data. The primary separation is between supervised and unsupervised learning. Unsupervised learning is generally associated with clustering, used for class discovery. On the other hand, when a class prediction is a goal, supervised learning is the option since it uses known class information (such as tumor/control labels). This rule stands for with class comparison, used for feature selection. All these strategies are being assiduously used in the analysis of biological data to infer new knowledge.

Statistics also plays a leading role in data analysis. Hypothesis testing is broadly used for class comparison. Genes are usually identified as differentially expressed (DE) between known classes of specimens using univariate analyses on each gene, such as Wilcoxon tests. In doing so, it is fundamental to consider the problem of multiple testing to avoid generating many false positives (sometimes referred to as “false discoveries”). In this article, we focus on supervised machine learning using feature selection (class comparison) and classification

(class prediction). Also, feature selection is performed applying statistical tests, and results from both approaches are compared and discussed.

Python was used to program all the analyses carried out in this article, with different libraries for statistical and machine learning methods. The methods used to perform the selection of the most characteristic features were statistical tests and Recursive Feature Elimination (RFE). Scikit-learn (Python library) offers three different methods: SelectKBest, SelectPercentile, and GenericUnivariateSelect, which can be used with a series of various statistical tests to select a specific number of features. For classification problems, it is possible to use the chi-square and ANOVA statistical tests. SelectKBest eliminates all the characteristics except the K with the highest score (more significant relationship with the output variable); SelectPercentile removes all features, except the highest score percentage; and GenericUnivariateSelect allows the selection of univariable characteristics with a configurable strategy [11].

Finally, we chose to work with two of the most used classifiers in the context of data mining for microarray data: KNN (K-Nearest Neighbors) and SVM (Support Vector Machine). As for the multi-class problem, different decomposition strategies exist. One way is to decompose the original problem into multiple binary problems and perform a classification through training and combining several binary classifiers. Many of these strategies are found in the framework of Error-Correcting Output Codes (ECOC), among which the One-vs.-One scheme stands out due to its simplicity but high effectiveness [10]. Figure 1 illustrates an abstraction of the main stages of the workflow and schematically shows the tasks explained in the following sections.

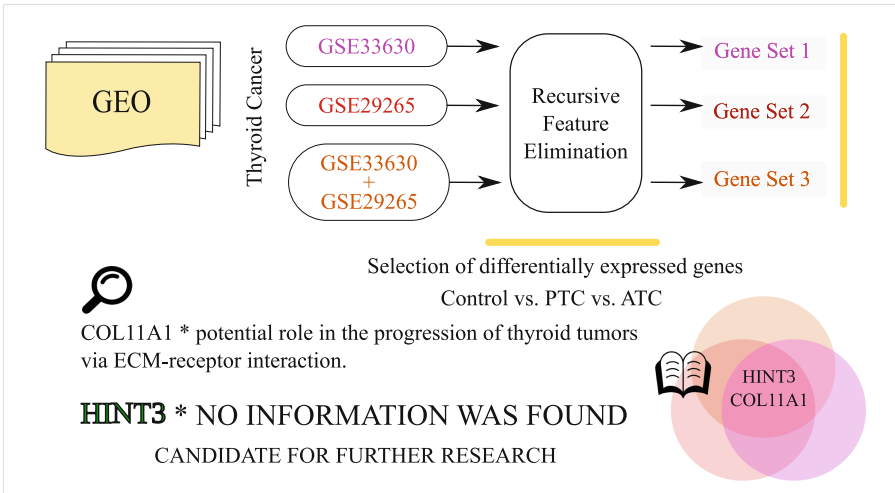


Fig. 1. Main stages of the workflow for thyroid data analysis.

### 3 Experimentation

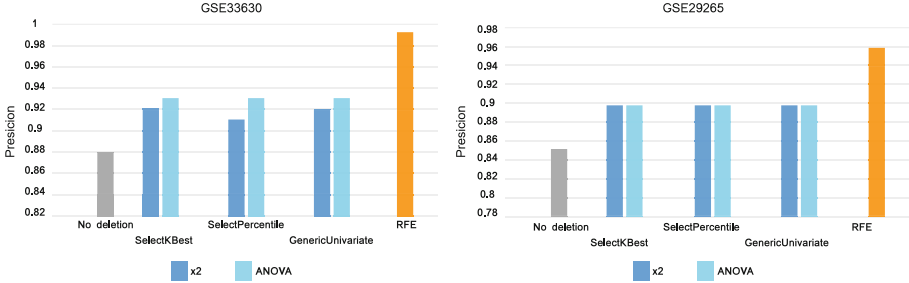
Experimentation was delineated as follows. First, the objective was to test how statistical methods performed regarding chi-square and ANOVA approaches. Later, statistical methods were compared with the most renowned machine learning method, the RFE. Regarding the data, several public databases provide numerous microarray experiments, most notably GEO [9] and ArrayExpress [1]. GEO is an international public repository that freely archives and distributes results from microarray experiments, Next-Gen sequences and other forms of high performance functional genomic data presented by the research community, and is the repository used here.

The first dataset selected for the analysis was GSE33630 [15]. It captures the expression levels of genes from ATC, PTC as well as healthy tissues. This dataset was selected for various reasons. First, it includes three different classes, which will allow us to analyze the behavior of this experiment under the different multiclass classification. Second, it presents a distribution of the samples of each class, which, despite not being ideal, is superior compared to the vast majority of datasets, composed of a large number of classes, and in some cases, with only one or two samples in certain classes. Regarding the source of the tissues, ATC samples were obtained from different hospitals in France and Belgium, while the samples of PTC and healthy thyroid tissues were obtained from Ukraine through the Chernobyl Tissue Bank. Members of the international pathology panel confirmed the diagnoses. The second selected dataset was GSE29265 [21], which, like the previous dataset, includes ATC, PTC, and standard samples. The reasons why this dataset was selected are similar to the previous ones; the distribution of the samples is acceptable, it includes the same three classes, and the data were preprocessed using the same normalization strategy, which will allow both experiments to be combined easily for analyzing them together. Samples from this dataset were obtained through the Chernobyl Tissue Bank, and also from French patients with no history of exposure to radiation.

#### 3.1 Phase 1: Individual Analysis of the Datasets

The first dataset analyzed is GSE33630, which outlines the gene expression profiles of 11 samples of ATC, 49 samples of PTC, and 45 samples of healthy tissues, including the full range of expression levels. The initial transformation applied was the elimination of those genes whose variance was low, justifying this decision under the fact that, if the value of a particular gene varies very little between standard thyroid samples and those of ATC or PTC, then the likelihood that it has a direct relationship to the onset of carcinoma is very low. However, to eliminate the genes with low variance is not enough; the dimensionality of the matrix is still very high, so it is necessary to apply other transformations. We then analyzed the impact of different feature selection procedures in the precision of the classifier. Figure 2-(left side) illustrates the precision obtained after selecting 50 genes using three univariate characteristics selection methods with various statistical tests, RFE, and the accuracy of the classifier without having made

any reduction. The classifier is constructed using SVM here in all cases, with the GridSearch strategy provided by Python to obtain the best set of parameters, and K-fold cross-validation with  $k = 5$  was set for all the trials.



**Fig. 2.** Precision comparison of SVM classifier after univariate feature selection (chi-square vs. ANOVA) and RFE to the GSE33630 dataset (left) and the GSE29265 dataset (right).

As it can be observed, although the accuracy of the classifier without performing feature selection is high, it is notorious how reducing the number of characteristics using statistical approaches improves its accuracy, especially with ANOVA, obtaining practically the same results for the three Python implementations. Furthermore, RFE achieved an even more significant improvement.

The next dataset is GSE29265; it outlines the gene expression of 9 samples of ATC, 20 samples of PTC, and 20 samples of healthy tissues. Initial transformations were the same as those in the previous dataset. Redundant information was eliminated and prepared to be analyzed by the Python libraries. Genes with low variance value were excluded, and methods of feature selection were analyzed. As seen in Fig. 2-(right side), the accuracy of the classifier without removing characteristics was also high, although slightly less than in the previous dataset. The tests with the three univariate selection methods and the different statistical tests yielded similar results for all the combinations: all increased the accuracy by almost 5%, but none stood out from the rest. Precision using RFE increased by about 6% compared to univariate methods, and close to 12% compared to a classifier working with a total of genes.

These results ratify several studies that have shown that SVM is particularly well suited for the analysis of gene expression from DNA microarray data, and when applied to the RFE method, the positive impact of gene selection on the performance of the classifier is remarkable. Likewise, the best-ranked genes found by the SVMs when applying RFE have mostly a plausible relationship with the diseases studied, in contrast to other selection methods, whose selected genes are related to the separation of the classes but are not relevant to diagnoses of these diseases [12].

A biological analysis of the selected genes was performed by doing a literature investigation. Next, we list the gene symbols of DE genes found by RFE:

- **GSE33630:** GABRB2, Hs.544373, CDH2, CCL21, TFF3, BCHE, MMRN1, COMP, DPT, NFAT5, TFPI2, CHI3L1, GRB10, NR4A3, SCN3A, TFPI, POSTN, APLNR, MFAP5, HRC13275, CNTNAP2, BMS1P20, IGLV1-44, COL10A1, SFTPA2, LYVE1, FLRT3, CLIC3, TRIM36, SLC27A6, COLEC12, C2orf40, SLC1A2, ENTPD1, HINT3, GJC1, CPNE4, WISP1, F2RL2, COL3A1, SNORD3D, HECW2, PLEKHA2, ARHGAP36, LPP, and COL11A1.
- **GSE29265:** WASIR2, LTF, TACSTD2, ATF3, ADM, NELL2, DPP4, BUB1B, IGF2BP3, DUSP4, MMP7, PROM1, EIF1AY, GAP43, PLN, DEFA1B, PRSS2, RASGRP1, TENM1, EGR3, RYR3, GRP, HMGA2, PLAUR, TMEM158, LRRC15, CXCL5, YME1L1, GREM1, RERGL, MECOM, ANLN, HHATL, INMT, FOXQ1, ZNF595, AGR3, TDRD9, HINT3, RIMS2, TCERG1L, Hs.720692, LOC101930164, LIPH, Hs.443967, SCARNA2, LCN10, Hs.553068, and COL11A1.

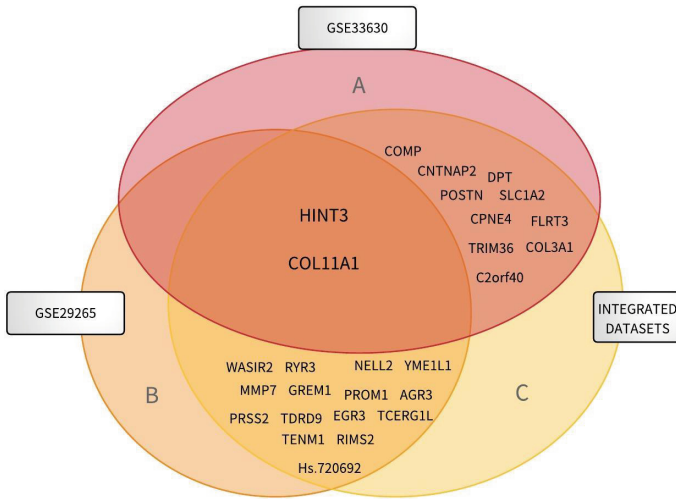
A bibliographic search was performed to seek the biological relevance of these genes. Several articles were examined, in which the identification of biomarkers from DE genes to improve the diagnosis of thyroid carcinomas was discovered. Regarding the GSE33630 dataset, four of the genes found in this work were also reported. Notably, in a previous study where the same dataset was analyzed using a different bioinformatics approach, COL11A1 was found as DE in ATC suggesting a potential role in the progression of such kind of tumor via ECM-receptor interaction [15]. Interestingly, other authors analyzed three different datasets in collaborative microarray analysis, and they found that COL3A1 gene expression was differently expressed in ATC [14]. Also, using a different set of tumor tissues, GABRB2 gene expression was found as the most upregulated gene in PTC compared to healthy thyroid tissue and its expression alone was reported to be able to discriminate between both kinds of tissues with excellent performance [2].

On the other hand, regarding DE genes found in this work for GSE29265, other authors previously reported only three of them. One is COL11A1, as already mentioned. Also, DPP4 was published as a secreted protein with a potential diagnostic marker of PTC [24]. Lastly, the BUB1B gene, whose high expression was associated with recurrence in PTC [6] and that was found DE in ATC when compared to healthy thyroid tissue [14]. To the best of our knowledge, the rest of the genes have not been highlighted in any study as possible biomarkers nor with any relationship to thyroid cancer.

### 3.2 Phase 2: Analysis of Integrated Datasets

One of the objectives of this work was to analyze how the integration of datasets affects the selection of DE genes. The significant difference in the genes selected

by the two datasets provides a good starting point. Various situations could occur, the dataset with more samples may have a preponderance in the selected genes, or a combination of all the genes might be obtained. The integration process consisted of joining both expression matrices in the same file and joining the labels of the samples in another file. It is important to remark that expression matrixes could be mixed since they were both identically normalized with an RMA procedure. Then, the transformations were the same as in the individual datasets: first, eliminate genes with low variance, and then apply RFE to the remaining set. The list of selected genes is HS6ST2, PDZK1IP1, WASIR2, AADACP1, COL1A1, NELL2, MMP7, PROM1, MMP1, SSPN, PRSS2, COMP, TENM1, EREG, EGR3, RYR3, CALCA, POSTN, FCER1A, PXDN, DPT, CNTNAP2, GUSBP9, YME1L1, GREM1, FLRT3, TRIM36, ZNF750, C2orf40, PCDHB5, KCNK17, XIST, AL521247, SLC1A2, LRRN1, COL3A1, TCEAL7, AGR3, TDRD9, HINT3, CPNE4, RIMS2, SCARA5, TCERG1L, Hs.720692, PLCXD2, FAR2, and COL11A1.



**Fig. 3.** DE genes (Gene Symbols). Sets A, B, and C are the genes not in the intersections.

From the 50 genes selected with the integrated dataset, ten belonged to the group chosen in GSE33630, fifteen to the group in GSE29265, two belonged to both, and twenty-three genes that had not been chosen before. For illustrative purposes, the intersections between the lists of genes are shown in Fig. 3 with a Venn diagram. At first, it might seem that the most significant dataset (GSE33630), whose size is twice the size of the GSE29265 dataset, did not have a substantial impact on the selection of the genes since only ten were selected again, compared with the fifteen selected genes that matched the ones chosen in the smallest dataset.

So, although within the 50 genes selected in the integrated dataset, a more significant number coincided with the dataset GSE29265, most of the genes highlighted in GSE33630 due to their potential relationship with thyroid cancer in the bibliography were selected again, while only one of the reported genes in GSE29265 coincided. All in all, this information gives an excellent start-up scenario to study those genes that are not yet associated with a cancer diagnosis. As a starting point for the analysis the Kaplan-Meier curve of these genes can be studied. Figure 4 shows the difference between the curves corresponding to high and low cohorts measured by log-rank for each gene (COL11A1 and HINT3).

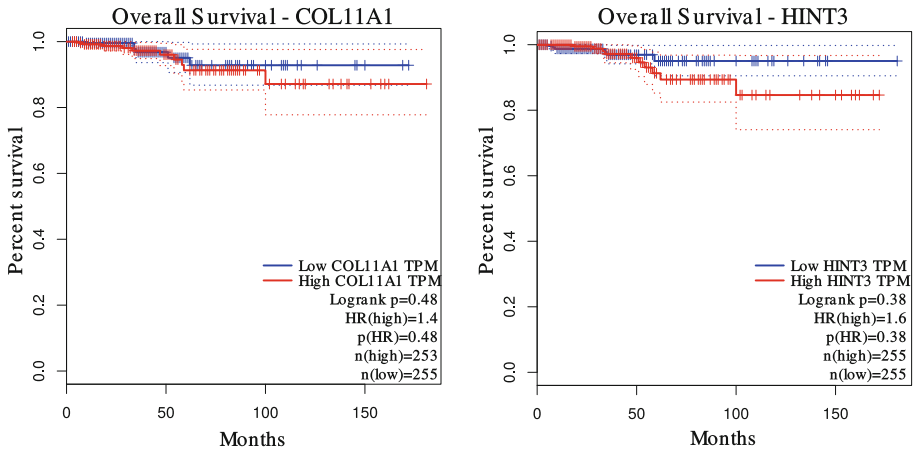


Fig. 4. Kaplan-Meier curve for COL11A1 (left) and HINT3 (right)

In particular, special attention should be paid to HINT3, since it was selected in all the experiments. This gene will be later approached in a separate section.

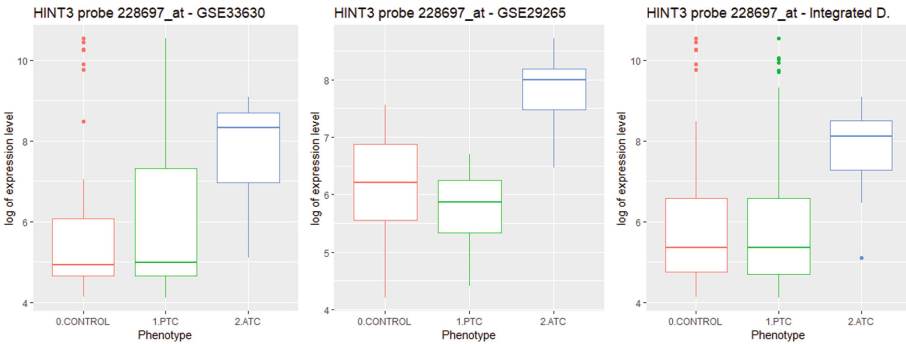
### 3.3 Phase 3: Multiclass Classifiers' Analysis

As was aforementioned, one of the reasons why these datasets were selected is that they are composed of more than two classes, which allows the analysis of different multiclass approaches. Python provides the possibility to specify which multiclass classification strategy (OVR or OVO) one wishes to use for an SVM classifier. This new analysis was performed with the data obtained after the selection of genes using the RFE strategy in each of the datasets. In the case of the first dataset (GSE33630), SVM with OVR was the best strategy, although just above SVM with OVO, while KNN obtained the worst accuracy of the three. In the second dataset (GSE29265), both SVM with OVO and SVM with OVR proved to be the best strategies obtaining the same accuracy, although KNN was only one point below both. Finally, when integrating both datasets, similar results were obtained, SVM was superior to KNN, this time with the

OVO technique being slightly better to the OVR technique. SVM was the best strategy to classify data from thyroid cancer datasets, imposing KNN in all cases. Regarding the multiclass classification techniques, no significant differences were observed.

### 3.4 HINT3: Statistical and Biological Hypothesis

HINT3 is a gene located at the human chromosome 6q22, which codifies by Histidine triad nucleotide-binding protein 3 (HINT3). Using microarray analysis HINT3 up-regulation was reported in hepatocellular carcinoma [23] and neurodegenerative disorders [7,25]. In hepatocellular carcinoma, HINT3 up-regulation was reported as being related to the effect of all-trans retinoic acid, which in serum-starved condition prevents cell death but induces cell migration and invasion [23]. Interestingly, HINT3 up-regulation was also related to the induction of apoptosis of neurons in the context of neurodegenerative disorders [18]. To date, only the previously mentioned *In Vitro* studies related the expression of HINT3 to pathophysiological conditions. Nevertheless, HINT3 remains poorly characterized since a deep understanding of its involvement in such pathologies was not carried out. In this work, we found that HINT3 is upregulated in human tissue samples from ATC compared to healthy thyroid tissue as well as PTC (see box-plots in Fig. 5)



**Fig. 5.** Box-plots for HINT3’s expression profiles, according to the phenotypes class, analyzed in each dataset (GSE33630, GSE29265, and both datasets integrated).

Regarding the notorious differences that can be visually observed between the expression values of control and ATC cases, statistical analysis was performed to ratify this evidence. ANOVA tests were first carried out. Once it was determined that there are significant differences between the means (see p values for ANOVA in Table 1), post-hoc rank tests (Tuckey) and the multiple pairwise comparisons allowed us to determine which means differ.



**Table 1.** Statistical results for differential expression of HINT3.

	ANOVA	Tuckey test		
	p value	Control-PTC	Control-ATC	ATC-PTC
GSE33630	0.00778	0.79	0.005	0.017
GSE29265	6.52e-07	0.301	0.000024	4.0e-7

Table 1 shows that, in ATC, HINT3 exhibits the highest difference of expression. Given that ATC is highly aggressive and resistant to therapies, we hypothesize that HINT3 may play a role in the progression from PTC to ATC favoring cell migration and invasion as well as cell death resistance. However, further work is needed to evaluate HINT3 function in the thyroid cancer context.

## 4 Discussion and Conclusions

Throughout this work, we presented different aspects of the application of statistics and machine learning to data obtained from microarray experiments, specifically to thyroid cancer studies. First, various approaches were evaluated to reduce the size of the matrix, taking them from an initial length of approximately 50,000 genes to only 50. This reduction allows for improving the accuracy of the classifiers. Several algorithms were applied to the datasets, starting with univariable feature selection by different statistical tests, and ending with RFE, which is postulated as one of the most effective for microarray problems in a large number of articles.

Results confirmed that the use of RFE improved the accuracy of the constructed classifiers in comparison to the precision without applying any reduction. Genes selected in each dataset were analyzed in-depth, and several of them have been part of numerous studies as possible biomarkers of both PTC and ATC. In the first dataset, most of the selected genes had a relationship with PTC, and only one was found to have a connection to ATC. In the second dataset, one prominent gene was related to PTC, and two to ATC, one being the same as relevant as it was in the previous dataset. These differences can be mainly due to two reasons. The first dataset had almost twice as many samples as the second, and the RFE selection method constructs models with the available attributes so that the fewer samples there are, the model will be less precise, and the selected genes might not be the most appropriate. The distribution of the samples is another of the influential factors in RFE, so the difference in the distribution of the samples in each dataset can generate a different result. When both datasets were integrated, the number of samples increased to 154, but the distribution did not improve: 12% ATC, 44% PTC, 44% normal. Most of the relevant genes selected in the first dataset also appeared when integrating both datasets. Only one of the second dataset was repeated which coincided in both datasets. Several ATC-related genes, according to the literature, were lost when integrating. A new gene with importance was selected in the integration, which is

related to PTC. With the results obtained, both individually and jointly, we can infer that the process of integrating datasets can be useful when selecting genes in microarray experiments. On the one hand, it can help confirm the importance of the selected genes by analyzing each dataset individually, as well as allowing, by increasing the number of samples, to choose new genes whose relevance had not been previously detected.

Once a small set was obtained, it was proposed to compare SVM and KNN. Given that the datasets are composed of three classes, it was also possible to differentiate within the alternatives of SVM according to the multiclass classification schemes, facing One-Vs-One and One-Vs-Rest. Results showed that both SVM and KNN are good choices when classifying microarray data, although SVM proved to be better. Regarding the multiclass classification techniques, it is not possible to give a satisfactory conclusion since there were practically no differences in the results obtained.

Finally, one issue must be remarked as a significant contribution presented in this work, besides the whole computational analysis. Regarding the selection of representative genes, two appeared in all the sets. As aforementioned, one of them, COL11A1, has been related to thyroid cancer in various studies. However, the remaining one, HINT3, seems so far not been associated with this type of disease. So, it is interesting to analyze a potential relationship between the expression of HINT3 and thyroid cancer.

## References






1. Athar, A., et al.: Arrayexpress update - from bulk to single-cell expression data. *Nucleic Acids Res.* **47**(Database-Issue), D711–D715 (2019)
2. Barros-Filho, M.C., Marchi, F.A., Pinto, C.A., Rogatto, S.R., Kowalski, L.P.: High diagnostic accuracy based on CLDN10, HMGA2, and LAMB3 transcripts in papillary thyroid carcinoma. *J. Clin. Endocrinol. Metabolism* **100**(6), E890–E899 (2015). <https://doi.org/10.1210/jc.2014-4053>
3. Behzadi, P., Ranjbar, R.: Dna microarray technology and bioinformatic web services. *Acta Microbiol. Immunol. Hung.* **66**(1), 19–30 (2019). <https://doi.org/10.1556/030.65.2018.028>
4. Brazma, A., et al.: Minimum information about a microarray experiment (MIAME) - towards standards for microarray data. *Nat. Genet.* **29**, 365–371 (2001)
5. Chandrababu, S., Bastola, D.: A novel prediction model for discovering beneficial effects of natural compounds in drug repurposing. In: Rojas, I., Valenzuela, O., Rojas, F., Herrera, L.J., Ortuño, F. (eds.) *IWBIO 2020. LNCS*, vol. 12108, pp. 811–824. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-45385-5\\_72](https://doi.org/10.1007/978-3-030-45385-5_72)
6. Chien, M.N., Yang, P.S., Lee, J.J., Wang, T.Y., Hsu, Y.C., Cheng, S.P.: Recurrence-associated genes in papillary thyroid cancer: an analysis of data from the cancer genome atlas. *Surgery* **161**(6), 1642–1650 (2017). <https://doi.org/10.1016/j.surg.2016.12.039>
7. Chou, T.F., Cheng, J., Tikh, I.B., Wagner, C.R.: Evidence that human histidine triad nucleotide binding protein 3 (hint3) is a distinct branch of the histidine triad (hit) superfamily. *J. Mol. Biol.* **373**(4), 978–989 (2007). <https://doi.org/10.1016/j.jmb.2007.08.023>

8. Chukwudozie, O.S., et al.: The relevance of bioinformatics applications in the discovery of vaccine candidates and potential drugs for covid-19 treatment. *Bioinform. Biol. Insights* **15** (2021). <https://doi.org/10.1177/11779322211002168>
9. Edgar, R., Domrachev, M., Lash, A.E.: Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res.* **30**(1), 207–210 (2002). <https://doi.org/10.1093/nar/30.1.207>
10. Escalera, S., Pujol, O., Radeva, P.: Error-correcting output codes library. *J. Mach. Learn. Res.* **11**(20), 661–664 (2010)
11. Géron, A.: Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems. O'Reilly Media, Sebastopol (2017)
12. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. *Mach. Learn.* **46**(1–3), 389–422 (2002). <https://doi.org/10.1023/A:1012487302797>
13. Haley, B., Roudnicky, F.: Functional genomics for cancer drug target discovery. *Cancer Cell* **38**(1), 31–43 (2020). <https://doi.org/10.1016/j.ccell.2020.04.006>
14. Hu, S., Liao, Y., Chen, L.: Identification of key pathways and genes in anaplastic thyroid carcinoma via integrated bioinformatics analysis. *Med. Sci. Monitor* **24**, 6438–6448 (2018). <https://doi.org/10.12659/MSM.910088>
15. Huang, Y., et al.: Bioinformatics analysis of key genes and latent pathway interactions based on the anaplastic thyroid carcinoma gene expression profile. *Oncol. Lett.* **13**, 167–176 (2017). <https://doi.org/10.3892/ol.2016.5447>
16. Kitahara, C.M., Schneider, A.B., Brenner, A.V.: Thyroid Cancer, chap. 44. Oxford University Press (2017). <https://doi.org/10.1093/oso/9780190238667.003.0044>
17. Miranda-Filho, A., et al.: Thyroid cancer incidence trends by histology in 25 countries: a population-based study. *Lancet Diabetes Endocrinol.* **9**(4), 225–234 (2021). [https://doi.org/10.1016/S2213-8587\(21\)00027-9](https://doi.org/10.1016/S2213-8587(21)00027-9)
18. Morte, B., Martínez, T., Zambrano, A., Pascual, A.: Monocyte-mediated regulation of genes by the amyloid and prion peptides in SH-SY5Y neuroblastoma cells. *Neurochem. Int.* **58**(6), 613–619 (2011). <https://doi.org/10.1016/j.neuint.2011.01.019>
19. Rossi, E.D., Pantanowitz, L., Hornick, J.L.: A worldwide journey of thyroid cancer incidence centred on tumour histology. *Lancet Diabetes Endocrinol.* **9**(4), 193–194 (2021). [https://doi.org/10.1016/S2213-8587\(21\)00049-8](https://doi.org/10.1016/S2213-8587(21)00049-8)
20. van Ruissen, F., Baas, F.: Serial Analysis of Gene Expression (SAGE), pp. 41–66. Humana Press, Totowa (2007)
21. Tomas, G., Vincent, D.: Sporadic vs. post-chernobyl papillary vs. anaplastic thyroid cancers (2012)
22. Tovar, H., Alvarez-Suarez, D.E., Gómez-Romero, L., Hernández-Lemus, E.: Bioinformatics of genome-wide expression studies. In: *Bioinformatics and Human Genomics Research*, chap. 5, pp. 73–99. CRC Press (2021)
23. Wang, W., Xu, G., Ding, C.L., Zhao, L.J., Zhao, P., Ren, H., Qi, Z.T.: All-trans retinoic acid protects hepatocellular carcinoma cells against serum-starvation-induced cell death by upregulating collagen 8a2. *FEBS J.* **280**(5), 1308–1319 (2013). <https://doi.org/10.1111/febs.12122>
24. Wu, C.C., et al.: Integrated analysis of fine-needle-aspiration cystic fluid proteome, cancer cell secretome, and public transcriptome datasets for papillary thyroid cancer biomarker discovery. *Oncotarget* **9**(15), 12079–12100 (2018). <https://doi.org/10.18632/oncotarget.23951>

25. Yan, T., Ding, F., Zhao, Y.: Integrated identification of key genes and pathways in Alzheimer's disease via comprehensive bioinformatical analyses. *Hereditas* **156**(25) (2019). <https://doi.org/10.1186/s41065-019-0101-0>
26. Young, A.P., Jackson, D.J., Wyeth, R.C.: A technical review and guide to RNA fluorescence in situ hybridization. *PeerJ* **8**, March 2020. <https://doi.org/10.7717/peerj.8806>



# Migrating CUDA to oneAPI: A Smith-Waterman Case Study

Manuel Costanzo<sup>1</sup> , Enzo Rucci<sup>1</sup> , Carlos García-Sánchez<sup>2</sup> ,  
Marcelo Naiouf<sup>1</sup> , and Manuel Prieto-Matías<sup>2</sup> 

<sup>1</sup> III-LIDI, Facultad de Informática, UNLP - CIC, 1900 La Plata, BA, Argentina  
{mcostanzo,erucci,mnaiouf}@lidi.info.unlp.edu.ar

<sup>2</sup> Dpto. Arquitectura de Computadores y Automática, Universidad Complutense de Madrid, 28040 Madrid, Spain  
{garsanca,mpmatias}@dacya.ucm.es

**Abstract.** In order to tackle the programming challenges related to heterogeneous computing, Intel recently introduced oneAPI, which is a new programming environment that allows code developed in the Data Parallel C++ (DPC++) language to be run on different devices such as CPUs, GPUs, and FPGAs, among others. To handle CUDA-based legacy codes, oneAPI provides a compatibility tool (`dpct`) that facilitates the migration to DPC++. In view of the large amount of existing CUDA-based software in the bioinformatics context, this paper presents our experiences porting *SW#db*, a well-known sequence alignment tool, to DPC++ using `dpct`. From the experimental work, it was possible to prove the usefulness of `dpct` for *SW#db* code migration and the cross-vendor GPU, cross-architecture portability of the migrated DPC++ code. In addition, the performance results showed that the migrated DPC++ code reports similar efficiency rates to its CUDA-native counterpart, or even better in some tests (by approximately 5%).

**Keywords:** oneAPI · SYCL · GPU · CUDA · Bioinformatics

## 1 Introduction

At present, heterogeneous computing and massively parallel architectures have proven to be an effective strategy for maximizing the performance and energy efficiency of computing systems [20]. That is the main reason why the programmers typically rely on a variety of hardware, such as CPUs, GPUs, FPGAs, and other kinds of accelerators. This creates the need for specialized libraries, tools, and APIs, which increase the programming costs and complexity, and complicate future code maintenance and extension.

On the one hand, Khronos Group has proposed SYCL<sup>1</sup>, which is an open standard, to face some of the programming issues related to heterogeneous computing. Although SYCL shares some characteristics with OpenCL (such as being

<sup>1</sup> <https://www.khronos.org/registry/SYCL/specs/sycl-2020/pdf/sycl-2020.pdf>.

royalty-free and cross-platform), it can actually be considered as an improved, high-level version of the latter. SYCL is an abstraction layer that enables code for heterogeneous systems to be written using standard, single-source C++ host code including accelerated code expressed as functions or *kernels*. SYCL implementations are often based on OpenCL, but also have the flexibility to use other backends such as CUDA or OpenMP. Furthermore, SYCL features asynchronous task graphs, buffers defining location-independent storage, interoperability with OpenCL, among other characteristics aimed to increase productivity [6, 18].

On the other hand, Intel recently introduced the *oneAPI* programming ecosystem, which provides a unified programming model for a wide range of hardware architectures. The core of the oneAPI environment is a simplified language for expressing parallelism on heterogeneous platforms, named Data Parallel C++ (DPC++), which can be summarized as C++ with SYCL. In addition, oneAPI also comprises a runtime, a set of domain-focused libraries and supporting tools [1].

In this scenario, GPUs can be considered the dominant accelerator, and CUDA is the most popular programming language for them nowadays [14]. Bioinformatics and Computational Biology are two fields that have been exploiting GPUs for more than two decades [12]. Many GPU implementations can be found in sequence alignment [3], molecular dynamics [9], molecular docking [13], and prediction and searching of molecular structures [11], among other application areas. Even though some applications achieve a better performance with CUDA, their portability to other architectures is severely restricted due to their proprietary nature.

To tackle CUDA-based legacy codes, oneAPI provides a compatibility tool (*dpct*) that facilitates the migration to the SYCL-based DPC++ programming language. A few preliminary studies assessing the usefulness of *dpct* can be found in simulation [1], math [2, 19], and cryptography [10]; however, to the best of our knowledge, no study has assessed their utility in Bioinformatics. In this paper, we present our experiences porting a biological software tool to DPC++ using *dpct*. In particular, we selected *SW#db* [8], which is a CUDA-based, memory-efficient implementation of the Smith-Waterman (SW) algorithm, which can be used either as a stand-alone application or a library. Our contributions are:

- An analysis of the effectiveness of *dpct* effectiveness for the CUDA-based *SW#db* migration, including a detailed summary of the porting steps that required manual modifications.
- An analysis of the DPC++ code’s portability, considering different target platforms and vendors (Intel CPUs and GPUs; NVIDIA GPUs).
- A comparison of the performance on different hardware architectures (Intel CPUs and GPUs; NVIDIA GPUs).

This work can be considered the starting point for a more exhaustive evaluation of CUDA-based biological tool migration to oneAPI. The remaining sections of this article are organized as follows. In Sect. 2, the background is presented, and in Sect. 3 we describe the migration process. Section 4 contains the experimental

work carried out and an analysis of the results. Finally, in Sect. 5, the conclusions and possible lines for future work are presented.

## 2 Background

### 2.1 The oneAPI Programming Ecosystem

Intel oneAPI<sup>2</sup> is a unified programming model for application development that can be used on different architectures, such as CPUs, GPUs, and even FPGAs. It seeks to facilitate the hard task of developing applications on a different set of hardware. By using oneAPI, the coding task can be performed at various levels: (1) invoking one of the multiple optimized libraries (oneMKL, oneDAL, oneVPL, etc.) that takes advantage of offloading technology in a transparent way to the programmer; or (2) via direct programming using the SYCL heterogeneous programming language supported by the Data Parallel C++ (DPC++) language. The DPC++ programming language (supported by Intel’s `dpcpp` compiler) combines the C++ language with SYCL, allowing the same source code to be compiled and executed across different accelerators.

Intel oneAPI comprises several programming tools and one of the most interesting with regards to code migration is a compatibility with regards tool named `dpct`. This tool converts applications written in the proprietary CUDA language to SYCL. According to Intel, this tool automatically migrates 80%-90% of the original CUDA code to SYCL. In addition, when it comes to non-ported code, `dpct` inlines comments (through warning messages) that help the programmer to migrate and tune the final DPC++ code. [4].

The migration process consists of 3 stages:

1. Running the `dpct` tool, which performs the automatic code migration.
2. Modification of the migrated code, attending to all the `dpct` warnings in order to obtain a first, executable version following the Diagnostics Reference<sup>3</sup>.
3. Verification of the correctness and efficiency of the resulting oneAPI program and implementation of the necessary modifications.

### 2.2 Smith-Waterman Algorithm

This algorithm was proposed by Smith and Waterman [17] to obtain the optimal local alignment between two biological sequences. SW employs a dynamic programming approach and presents quadratic time and space complexities. Furthermore, it has been used as the basis for many subsequent algorithms and is often employed as a benchmark when comparing different alignment techniques [5].

<sup>2</sup> <https://www.oneapi.com/>.

<sup>3</sup> Diagnostics Reference of Intel® DPC++ Compatibility Tool available at: <https://software.intel.com/content/www/us/en/develop/documentation/intel-dpcpp-compatibility-tool-user-guide/top/diagnostics-reference.html>.

The SW algorithm can be used to compute: (a) pairwise alignments (one-to-one); or (b) database similarity searches (one-to-many). Both cases have been parallelized in the literature. In case (a), a single SW matrix is calculated and all Processing Elements (PEs) work collaboratively (*intra-task parallelism*). Due to inherent data dependencies, neighbouring PEs communicate in order to exchange border elements. In case (b), multiple SW matrices are calculated simultaneously without communication between the PEs (*inter-task parallelism*) [3].

### 2.3 SW#

SW# is a tool for computing biological sequence alignments that can be used as an API-based library or as a standalone command-line executable [7]. It is considered a versatile tool since it works with both protein and DNA sequences, being able to compute pairwise alignments as well as database similarity searches.

SW#db is the package for fast exact similarity searches, which works by simultaneously utilizing the CPU and GPU(s). The GPU part is based on CUDA and follows both inter-task and intra-task parallelism approaches (depending on the sequence length). For its part, CPU just exploits inter-task parallelism through multithreading and SIMD instructions<sup>4</sup>. Through dynamic work distribution and dynamic communication between the CPU and GPU, SW#db significantly reduces the execution time.

## 3 Implementation

### 3.1 Differences Between CUDA and DPC++

Before migrating a code from CUDA to oneAPI, certain differences should be considered.

**Memory Model:** on the one hand, CUDA provides two different types of memory model:

1. Conventional model: based on explicit memory operations between the CPU and GPU to be specified.
2. Unified Memory (UM): introduced in CUDA 6, this model allows to the programmer to address the CPU and GPU memory in a transparent manner based in such a way of a shared memory pool.

On the other hand, oneAPI offers three abstractions for managing memory:

1. Buffers: these are data abstractions that represent one or more objects of a given C++ language type. Buffers represent data objects rather than specific memory addresses, so the same buffer can be allocated to several different memory locations on different devices, or even on the same device, for performance reasons.

---

<sup>4</sup> In particular, it makes use of the OPAL library for the CPU part <https://github.com/Martinos/opal>.



2. Images: these are a special type of buffer created specially for image processing. They include support for special image formats, and image reading through sampling objects, among other features.
3. Unified Shared Memory (USM): this consists of creating a unified virtual memory space in which pointers are shared between the CPU and the device (similar to the CUDA UM).

**Verbosity:** in DPC++, all the variables used within a kernel must be declared and explicitly sent to the functions, as well as other aspects that in CUDA are not mandatory. On the contrary, in CUDA it is possible to indicate the variables that you wish to send to the device and implicitly use them in the kernels. These issues may cause the oneAPI code to be longer than its CUDA counterpart.

### 3.2 Migrating CUDA Codes to DPC++

In general, `dpct` is not able to generate fully functional DPC++ code. Thus, it is necessary to perform hand-tuned adaptations. However, the `dpct` tool reports a list of warnings, which facilitates successful refactoring.

**Warnings Generated by `dpct`:** these warnings range from simple recommendations (i.e. to improve performance) to more complex issues, such as fragments of code that have not been successfully migrated.

This section details the messages reported by the migration `dpct` tool when porting the SW#, and the manual adaptation is carried out to obtain the final DPC++ code.

```
DPCT1003: Migrated API does not return error code. (*, 0)
         is inserted. You may need to rewrite this code

DPCT1009: SYCL uses exceptions to report errors and does
         not use the error codes. The original code was
         commented out and a warning string was inserted. You
         need to rewrite this code.
```

Both warnings occur when using native CUDA functions, such as CUDA error codes (Fig. 1a). Since `dpct` cannot translate them, it modifies the code to still keep it functional (Fig. 1b). Generally, this technique is used when exchanging data with the device. Figures 1a) and 1b) show memory allocations on the GPU using CUDA and oneAPI, respectively. By default, `dpct` tries to use the USM model because it produces a smaller volume of code and allows `dpct` to support more memory-related APIs.

<pre> 1  size_t valuesSize = 2      databaseLen * sizeof(double); 3  double* valuesGpu; 4 5  CUDA_SAFE_CALL( 6      cudaMalloc( 7          &amp;valuesGpu, valuesSize 8      ) 9  ); </pre>	<pre> 1  size_t valuesSize = 2      databaseLen * sizeof(double); 3  double* valuesGpu; 4 5  CUDA_SAFE_CALL(( 6      valuesGpu = (double *)sycl::malloc_device( 7          valuesSize, dpct::get_default_queue(), 8      0)); </pre>
(a) CUDA	(b) DPC++

**Fig. 1.** CUDA\_SAFE\_CALL example

DPCT1005: The SYCL device version is different from CUDA Compute Compatibility. You may need to rewrite this code.

This problem is related to the previous one and appears when querying for intrinsic CUDA attributes. While `dpct` can obtain information from the GPU, such as the number of registers, or maximum memory size, among others<sup>5</sup>, some CUDA-proprietary attributes (e.g. CUDA driver information) are not translatable. Figure 2a shows that, in the original code, the number of CUDA blocks and threads depends on the driver version. Figure 2b presents the migrated code, showing that it is possible to obtain information about GPU properties, with the exception of those specific to CUDA.

DPCT1049: The workgroup size passed to the SYCL kernel may exceed the limit. To get the device limit, query `info::device::max_work_group_size`. Adjust the workgroup size if needed.

<pre> 1  cudaDeviceProp properties; 2  cudaGetDeviceProperties( 3      &amp;properties, card 4  ); 5 6  bool major = properties.major &lt; 2; 7  int threads = major ? 64 : 128; 8  int blocks = major ? 360 : 480; </pre>	<pre> 1  dpct::device_info properties; 2 3  dpct::dev_mgr::instance() 4      .get_device(card) 5      .get_device_info(properties); 6 7  bool major = false; 8  int threads = major ? 64 : 128; 9  int blocks = major ? 360 : 480; </pre>
(a) CUDA	(b) DPC++

**Fig. 2.** Querying device properties

To run the CUDA kernel, both block and thread sizes must be configured; however, each device has a different size limit. `dpct` alerts the programmer that

<sup>5</sup> <https://docs.oneapi.io/versions/latest/dpcpp/iface/device.html>.

the migrated code may exceed the maximum work-group limit that the underlying architecture supports. In addition, it recommends adjusting the code if necessary. Figure 3a shows how to run the kernel in CUDA, while Fig. 3b shows the DPC++ counterpart.

<pre> 1  solveShort&lt;&lt;&lt;blocks, threads&gt;&gt;&gt;(...); </pre> <p style="text-align: center;">(a) CUDA</p>	<pre> 1  dpct::get_default_queue() 2    .submit([&amp;](sycl::handler &amp;cgh) { 3    ... 4    cgh.parallel_for( 5      sycl::nd_range&lt;3&gt; 6        (sycl::range&lt;3&gt;(1, 1, blocks) * 7         sycl::range&lt;3&gt;(1, 1, threads), 8         sycl::range&lt;3&gt;(1, 1, threads)), 9      [=](sycl::nd_item&lt;3&gt; item_ct1) { 10         solveShort(...); 11       }); 12 }); </pre> <p style="text-align: center;">(b) DPC++</p>
---	--

**Fig. 3.** Kernel launch with dynamic work-group size

DPCT1065: Consider replacing `sycl::nd_item::barrier()` with `sycl::nd_item::barrier(sycl::access::fence_space::local_space)` for better performance if there is no access to global memory.

In this situation, `dpct` recommends the programmer to use an additional parameter when synchronizing threads within the kernel as long as no global memory is used. By default, the tool does not automatically optimize this aspect because it cannot discern whether this memory is being used. An example of the CUDA thread synchronization and the migrated oneAPI code can be seen in the Figs. 4a and 4b, respectively.

<pre> 1  ... 2  __syncthreads(); 3  ... </pre> <p style="text-align: center;">(a) CUDA</p>	<pre> 1  ... 2  item_ct1.barrier(); 3  ... </pre> <p style="text-align: center;">(b) DPC++</p>
--	--

**Fig. 4.** Thread synchronization

```

DPCT1084: The function call has multiple migration
         results in different template instantiations that
         could not be unified. You may need to adjust the code
         .

```

In CUDA, generic functions are a common way of reducing code size, since they permit code reuse for data of different types. Although oneAPI supports this programming feature, it cannot automatically port this kind of code due to the multiplicity of possible migration options. Figure 5a shows a CUDA example in which instructions depend on the type of parameter sent to the kernel function. Figure 5b shows the corresponding migrated code.

```

DPCT1059: SYCL only supports 4-channel image format.
         Adjust the code.

```

In CUDA, texture memory variables can be allocated through 1 to 4 channels, while in SYCL, texture memory is accessed through images. As is reported by the `dpct` warning, SYCL only supports the use of 4-channel images, so the programmer must adapt the parts of the code in which images of different sizes are used. In Fig. 6a a 1-channel texture variable is declared in CUDA (placed in the device) and finally a data is read from it. Figure 6b presents a possible adjustment to the corresponding code to convert a 1-channel texture variable to an equivalent 4-channel one. As can be seen, this conversion requires to the modification of the indexes through which the memory is accessed to obtain the correct data. Thus, a 2-bit right shift (equivalent to `DIV 4`) combined with a logical `AND 3` operation (equivalent to `MOD 4`) must be performed in the corresponding read operation.

**Runtime and Results Check:** once the code compiles correctly, it must be verified that there are no execution errors and that the results obtained are correct. In this case, although the oneAPI program compiled correctly, the following runtime error appeared:

```

For a 1D/2D image/image array, the width must be a Value
    >= 1 and <= CL_DEVICE_IMAGE2D_MAX_WIDTH.

```

```

1  class SubVector {
2  public:
3      __device__ int operator()(...) {
4          ...
5      }
6  };
7
8  template <class Sub>
9      __global__ static void solveLong(
10         ..., Sub sub) {
11         sub(...);
12     }
13
14     solveLong<<<blocks, threads>>>(
15         ..., SubVector());

```

(a) CUDA

```

1  class SubVector {
2  public:
3      int operator()(...) {
4          ...
5      }
6  };
7
8  template <class Sub>
9      static void solveLong(..., Sub sub) {
10         sub(...);
11     }
12
13     cgh.parallel_for(
14         sycl::nd_range<3>
15         (sycl::range<3>(1, 1, blocks) *
16         sycl::range<3>(1, 1, threads),
17         sycl::range<3>(1, 1, threads)),
18         [=](sycl::nd_item<3> item_ct1) {
19             solveLong(..., SubVector());
20         });
21 };

```

(b) DPC++

**Fig. 5.** Generic functions

This error appears because SYCL images have a limited size, with the maximum size of the 1D images (vectors) being smaller than their 2D counterparts (matrices). To solve this issue, this image object must be converted to another DPC++ memory abstraction: either buffers or USM. We chose USM because the required modifications were simpler compared with the other option.

Figure 7a shows how a 2-level texture memory is allocated on the GPU, while Fig. 7b illustrates how to use USM to send an array to the device. In this way, the read mechanism also changes, both in CUDA (Fig. 8a) and in DPC++.

```

1  texture<char> colTexture;
2
3  int colSize = colsGpu * sizeof(char);
4  char *colGpu;
5  cudaMalloc(&colGpu, colSize);
6  cudaMemcpy(colGpu, colCpu,
7             colSize, TO_GPU);
8  cudaBindTexture(NULL, colTexture,
9                 colGpu, colSize);
11 char v = tex1Dfetch(colTexture, 10);

```

(a) CUDA

```

1  //dpct::image_wrapper<char, 1> colTexture;
2  dpct::image_wrapper<sycl::char4, 1> colTexture;
3  int colSize = colsGpu * sizeof(char);
4  char* colGpu;
5
6  colGpu = (char *)sycl::malloc_device(
7      colSize, dpct::get_default_queue());
8
9  dpct::get_default_queue()
10     .memcpy(colGpu, colCpu, colSize).wait();
11
12 colTexture.attach(colGpu, colSize);
13
14 // DIV 4 y MOD 3
15 char v = colTexture.read(10 >> 2)[10 & 3];

```

(b) DPC++

**Fig. 6.** 4-channel texture memory

```

1 texture<int, 2,
2   cudaReadModeElementType> seqsTexture;
3
4 cudaArray *sequencesGpu;
5 cudaChannelFormatDesc channel =
6   seqsTexture.channelDesc;
7 cudaMallocArray(&sequencesGpu,
8   &channel, sequencesCols, sequencesRows);
9 cudaMemcpyToArray(sequencesGpu, 0, 0,
10  sequences, sequencesSize, TO_GPU);
11 cudaBindTextureToArray(
12   seqsTexture, sequencesGpu);
1   static int *seqsGpu;
2
3 seqsGpu = (int *)sycl::malloc_device(
4   sequencesCols * sequencesRows * sizeof(int),
5   dpct::get_default_queue());
6
7 dpct::get_default_queue(
8   .memcpy(seqsGpu, sequences, sequencesCols *
9   sequencesRows * sizeof(int))
10  .wait());

```

(a) CUDA (b) DPC++

**Fig. 7.** CUDA 2-D texture memory adaptation using the DPC++ USM

```

1 int columnCodes = tex2D(
2   seqsTexture, colOff, j + rowOff);
1   int columnCodes =
2   seqsGpu[(j + rowOff) * sequencesCols + colOff];

```

(a) CUDA (b) DPC++

**Fig. 8.** Data accessing in 2D array

After the DPC++ program executed successfully, different tests were performed and their results were verified to ensure that they were equivalent to those of the original CUDA code.

## 4 Experimental Results

### 4.1 Experimental Design

All the tests were carried out using the platforms described in Table 1. The oneAPI and CUDA versions are 2022.0 and 11.5, respectively, and in order to run DPC++ codes on NVIDIA GPU, we built a DPC++ toolchain with support for NVIDIA CUDA, as it is not supported by default on oneAPI<sup>6</sup>. The performance was evaluated by carrying out experiments similar to those in previous works [15, 16], searching 20 query protein sequences against the well-known UniProtKB/Swiss-Prot database (release 2021\_04)<sup>7</sup>, and including the following features:

- The input queries range in length from 144 to 5478, and they were extracted from the Swiss-Prot database (accession numbers: P02232, P05013, P14942, P07327, P01008, P03435, P42357, P21177, Q38941, P27895, P07756, P04775, P19096, P28167, P0C6B8, P20930, P08519, Q7TMA5, P33450, and Q9UKN1).
- The database contains 204173280 amino acid residues in 565928 sequences with a maximum length of 35213.

<sup>6</sup> <https://intel.github.io/llvm-docs/GetStartedGuide.html>.

<sup>7</sup> Swiss-Prot: <https://www.uniprot.org/downloads>.

**Table 1.** Experimental platforms used in the tests

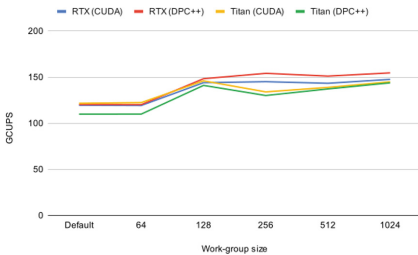
CPU			GPU			
ID	Processor	RAM Memory	ID	Vendor (type)	Model (architecture)	GFLOPS peak (SP)
<i>Core-i5</i>	Intel Core i5-7400	16 GB	<i>Titan</i>	NVIDIA (Discrete)	Titan X (Pascal)	10970
<i>Core-i3</i>	Intel Core i3-4160	8 GB	<i>RTX</i>	NVIDIA (Discrete)	RTX 2070 (Turing)	7465
<i>Core-i9</i>	Intel Core i9-10920X	32 GB	<i>Iris XE</i>	Intel (Discrete)	Iris Xe MAX Graphics (Gen 12.1)	2534
<i>Xeon</i>	Intel Xeon E-2176G	65 GB	<i>P630</i>	Intel (Integrated)	UHD Graphics P630 (Gen 9.5)	441.6

- BLOSUM62 and 10(2) were set as the scoring matrix and gap insertion (extension) penalty, respectively.

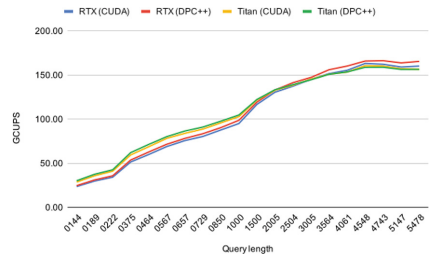
As SW#db is hybrid CPU-GPU software, just a single thread was configured at the CPU level (flag T=1) to minimize its impact on the overall performance. In addition, different work-group<sup>8</sup> sizes were configured for kernel execution. Finally, each test was run twenty times and the performance was calculated as the average in order to avoid variability.

## 4.2 Performance Results

GCUPS (billion cell updates per second) is commonly used as the performance metric in the context of SW [15]. Figure 9 presents the performance of the both CUDA and DPC++ versions on two NVIDIA GPUs when varying work-group size. First, it can be noted that both codes are sensitive to the work-group size. In fact, the best performances are obtained when using work-group sizes that are different to the ones that SW#db set as default. Regarding the performance on each NVIDIA GPU, there are no significant differences between the two codes



**Fig. 9.** Performance of both CUDA and DPC++ versions on the NVIDIA GPUs when varying work-group size.



**Fig. 10.** Performance of both CUDA and DPC++ versions on the NVIDIA GPUs when varying the query length.

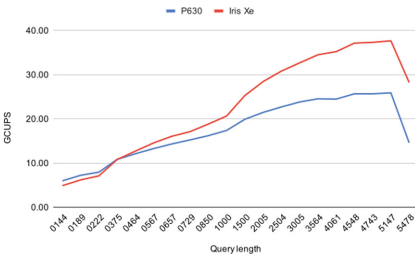
<sup>8</sup> A DPC++ work-group is equivalent to a CUDA block.

on the Titan. However, on the RTX, this situation changes; the DPC++ version actually outperforms its CUDA counterpart (by approximately 5%).

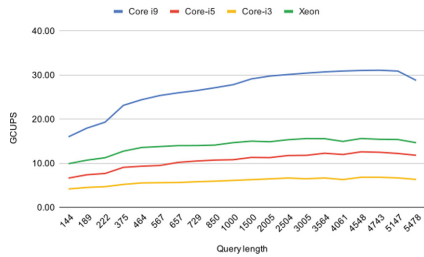
Figure 10 deepens the above analysis by presenting the performance of both the CUDA and DPC++ versions on the NVIDIA GPUs when varying the query length (optimal work-group size was used for each case). It can be noted that all versions benefit from larger workloads. As expected, the CUDA code achieves the same GCUPS as the DPC++ one for all query lengths on the Titan. Both the DPC++ and CUDA versions present practically the same performance on the RTX, with the latter outperforming the former on the largest sequences.

To verify cross-vendor GPU portability, the DPC++ code was executed on two different Intel GPUs, varying the query length (see Fig. 11). Due to the absence of an optimized version for both Intel devices, little can be said about its performance. However, it is important to remark that only two minor changes were necessary to carry out these tests: (1) setting the appropriate work-group size; and (2) setting the corresponding backend. As the ported code was compiled and executed with minimal tuning, there is probably room for further improvement.

Finally, Fig. 12 presents the performance of the DPC++ code on 4 different Intel CPUs, demonstrating its cross-architecture portability. With regards to performance, more GCUPS are achieved as the query length increases. Once again, running the migrated code only required minimal intervention and its performance could be improved through fine tuning.



**Fig. 11.** Performance of DPC++ code on the different Intel GPUs when varying the query length.



**Fig. 12.** Performance of DPC++ code on the different Intel CPUs when varying the query length.

## 5 Conclusions and Future Work

The recently introduced Intel oneAPI ecosystem aims to respond to the programming challenge posed by heterogeneous computing. In this paper, we have presented our experiences when migrating a CUDA-based, biological software tool to DPC++ using the oneAPI framework. The main findings of this research are:



- `dpct` proved to be an effective tool for SW#db code migration to DPC++. While it was not able to translate the complete code, `dpct` did most of the work and gave hints to the programmer on the pending parts.
- The migrated code could be successfully executed on CPUs and also GPUs from different vendors, demonstrating its cross-vendor GPU, cross-architecture portability.
- The performance results showed that the migrated DPC++ code is comparable to the original CUDA one. In fact, DPC++ can even be faster in some cases. As the ported code was compiled and executed with minimal tuning, there is probably room for further improvement.

Future work will focus on:

- Understanding the gap in performance between DPC++ and CUDA code, and optimizing DPC++ code to reach its maximum performance.
- Carrying out more exhaustive experimental work. In particular, by considering other alignment operations, larger workloads, and multi-GPU execution, among other aspects, in order to increase the representativeness of this study.
- Running the DPC++ code on other architectures such as FPGAs, to verify its cross-architecture portability.

**Acknowledgements.** This paper has been supported by the EU (FEDER) and the Spanish MINECO and CM under grants S2018/TCS-4423, RTI2018-093684-B-I00 and PID2021-126576NB-I00.

## References

1. Christgau, S., Steinke, T.: Porting a legacy CUDA stencil code to oneAPI. In: 2020 IEEE IPDPSW, pp. 359–367 (2020). <https://doi.org/10.1109/IPDPSW50202.2020.00070>
2. Costanzo, M., Rucci, E., Sanchez, C.G., Naiouf, M.: Early experiences migrating cuda codes to oneapi. In: Short papers of the 9th Conference on Cloud Computing Conference, Big Data & Emerging Topics. pp. 14–18 (2021). <http://sedici.unlp.edu.ar/handle/10915/125138>
3. De Oliveira Sandes, E.F., Boukerche, A., De Melo, A.C.M.A.: Parallel optimal pairwise biological sequence comparison: algorithms, platforms, and classification. *ACM Comput. Surv.* **48**(4) (2016). <https://doi.org/10.1145/2893488>
4. Hariharan, N., Mallady, R.K., Kapoor, A., O’Leary, K.: Heterogeneous programming using oneapi. *Parallel Universe* **39**, 5–18 (2020)
5. Hasan, L., Al-Ars, Z.: *Computational Biology and Applied Bioinformatics*, chap. 9, pp. 187–202. InTech, September 2011
6. Keryell, R., Yu, L.Y.: Early experiments using SYCL single-source modern C++ on Xilinx FPGA. In: Proceedings of the IWOCCL 2018. ACM, New York (2018). <https://doi.org/10.1145/3204919.3204937>
7. Korpar, M., Sikic, M.: SW# - GPU-enabled exact alignments on genome scale. *Bioinformatics* **29**(19), 2494–2495 (2013). <https://doi.org/10.1093/bioinformatics/btt410>

8. Korpar, M., Sobic, M., Blazeka, D., Sikic, M.: SWdb: GPU-accelerated exact sequence similarity database search. *PLOS ONE* **10**(12), 1–11 (2016). <https://doi.org/10.1371/journal.pone.0145857>
9. Loukatou, S., et al.: Molecular dynamics simulations through GPU video games technologies. *J. Mol. Biochem.* **3**(2), 64 (2014)
10. Marinelli, E., Appuswamy, R.: XJoin: portable, parallel hash join across diverse XPU architectures with OneAPI. *ACM* (2021). <https://doi.org/10.1145/3465998.3466012>
11. Mrozek, D., Brożek, M., Malysiak-Mrozek, B.: Parallel implementation of 3d protein structure similarity searches using a GPU and the CUDA. *J. Mol. Model.* **20**(2), 1–17 (2014)
12. Nobile, M.S., Cazzaniga, P., Tangherloni, A., Besozzi, D.: Graphics processing units in bioinformatics, computational biology and systems biology. *Briefings Bioinform.* **18**(5), 870–885 (2016). <https://doi.org/10.1093/bib/bbw058>
13. Ohue, M., Shimoda, T., Suzuki, S., Matsuzaki, Y., Ishida, T., Akiyama, Y.: Megadock 4.0: an ultra-high-performance protein-protein docking software for heterogeneous supercomputers. *Bioinformatics* **30**(22), 3281–3283 (2014)
14. Robert Dow: GPU shipments increase year-over-year in Q3 (2021). <https://www.jonpeddie.com/press-releases/gpu-shipments-increase-year-over-year-in-q3>
15. Rucci, E., Garcia, C., Botella, G., Giusti, A.E.D., Naiouf, M., Prieto-Matias, M.: Oswald: Opencl smith-waterman on altera’s fpga for large protein databases. *Int. J. High Perform. Comput. Appl.* **32**(3), 337–350 (2018). <https://doi.org/10.1177/1094342016654215>
16. Rucci, E., Sanchez, C.G., Juan, G.B., De Giusti, A., Naiouf, M., Prieto-Matias, M.: Swimm 2.0: enhanced smith-waterman on intel’s multicore and manycore architectures based on avx-512 vector extensions. *Int. J. Parallel Programm.* **47**(2), 296–316 (2019)
17. Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences. *J. Mol. Biol.* **147**(1), 195–197 (1981)
18. The Khronos SYCL Working Group: SYCL Specification (2020). <https://www.khronos.org/registry/SYCL/specs/sycl-2020/pdf/sycl-2020.pdf>
19. Tsai, Y.M., Cojean, T., Anzt, H.: Porting a sparse linear algebra math library to intel gpus (2021)
20. Zahran, M.: Heterogeneous computing: here to stay. *Commun. ACM* **60**(3), 42–45 (2017)

# **Computational Proteomics**



# Fuzzy-Inference System for Isotopic Envelope Identification in Mass Spectrometry Imaging Data

Anna Glodek<sup>(✉)</sup>

The Silesian University of Technology, 44-100 Gliwice, Poland  
anna.glodek@polsl.pl

**Abstract.** Mass spectrometry is one of the widely used techniques in proteome studies, enabling, inter alia, the identification of proteins present in biological samples based on the analysis of unique peptides originating from the proteins of interest. It should be noted, however, that from the point of view of mass spectrometry data pre-processing, the identification of an isotopic envelope of a peptide plays a crucial role in its precise annotation. Nowadays, there are a plethora of algorithms created to reach this goal, nevertheless, they strongly depend on the type of experimental platform (especially the ionization method: MALDI, ESI, etc.) and are usually dedicated to a specific type of molecules (e.g., lipids or peptides). We propose a unique approach that combines information about the spatial distribution of a molecule across a tissue section sample with a fuzzy-inference system. The mass spectrum was considered as a set of peaks. For each peak, an intensity map was constructed, that presents the spatial distribution of peptide abundance across the tissue sample. The obtained intensity map was further analyzed and the outcome of this analysis was applied to the fuzzy-inference system and to the fuzzy C-means image segmentation method.

**Keywords:** Isotopic envelope · Mass spectrometry imaging · Proteomics · MALDI · Fuzzy C-means

## 1 Introduction

Mass Spectrometry Imaging (MSI) is a tool that enables the mapping of the spatial distribution of biomolecules across the tissue of interest [1]. MALDI mass spectrometry imaging methods are widely used in various fields of bioanalysis to analyze proteins, peptides, lipids, or exogenous and endogenous small molecules [2]. Moreover, MALDI MSI has attracted a great deal of interest in the analysis of cancer tissues [3] and it turned out that it is a promising tool for cancer diagnostics [4]. It offers the possibility to generate maps of the spatial distribution of hundreds of molecules across an analyzed tissue section in a single imaging experiment [5]. First, a properly prepared tissue section is introduced into the mass spectrometer, then spectra are acquired at the sample surface (rectangular  $x, y$  grid). As a result, an array of spectra is obtained, where each spectrum is a molecular profile of the area irradiated by the laser [2]. The use of MSI can expand the

amount of information that could be obtained from a tissue – it combines molecular and morphological information [6] since spatially resolved mass spectrometry measurements are taken from a tissue section without destroying it [3].

A mass spectrum is obtained when a beam of ions is separated according to the mass-to-charge ratios ( $m/z$ ) of the ionic species contained within it [7, 8]. A mass spectrum of a protein can be considered as a set of peaks [9]. A peak on a mass spectrum is a signal derived from an ion that was detected in a mass spectrometer. In order to handle such data correctly, various preprocessing methods have been developed, such as baseline removal, smoothing, peak picking, etc. One of them is deisotoping, which is based on the search for the isotopic envelope across the whole spectrum, since some peaks do not originate from different compounds, but are isotopes of one element. An isotope pattern in mass spectrometry is a set of peaks related to ions with the same chemical formula but containing different isotopes [8]. In the case of high-resolution mass spectrometry (which enables the resolution of signals originating from molecules of a given compound differing in isotopic composition) identification of the isotopic envelope (isotope pattern) of a compound is of crucial importance from the correct compound identification point of view. Nowadays, there are several methods that try to deal with the problem of deisotoping, but they are depended on the type of experimental platform and are dedicated to a specific type of molecule (proteins or lipids). Some of the aforementioned algorithms have been tested by the authors and published in [10].

In order to properly identify isotopic envelopes of tryptic peptides, here we have proposed a new approach based on the spatial distribution of peaks and a fuzzy-inference system and fuzzy C-means image segmentation method. The mass spectrum was considered as a set of peaks. For each peak, an intensity map was constructed, that presents the spatial distribution of peptide abundance across the tissue sample. Hence, there is a strong need for defining the way how to choose the criterion that can distinguish intensity maps for peaks that are members of one isotopic envelope from those that are not. To determine whether the image represents peaks that are members of an isotopic envelope, intensity differences were taken into consideration. The outcome of this analysis was applied to the fuzzy-inference system and to the fuzzy C-means image segmentation method. In order to confirm obtained results, an expert in the field of mass spectrometry annotated peaks that were members of an isotopic envelope.

## 2 Materials and Methods

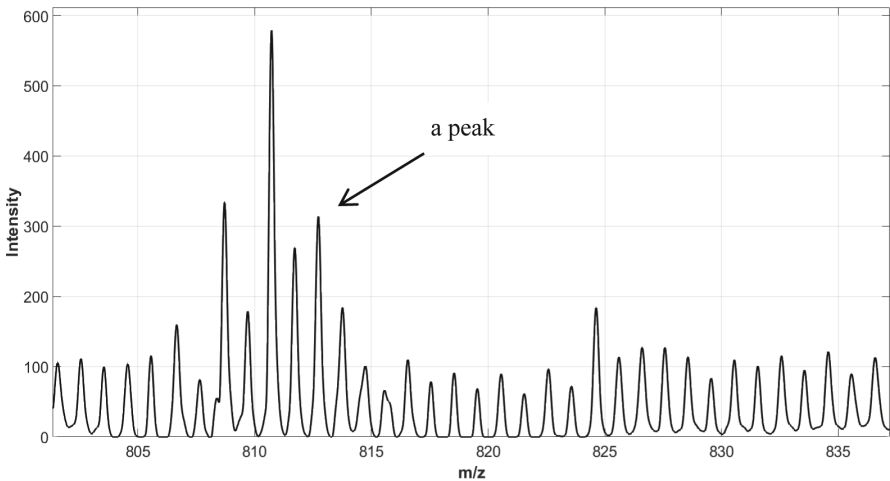
### 2.1 Data Characteristic

Data were provided by Maria Skłodowska-Curie National Research Institute of Oncology in Gliwice (Poland). The material was collected from a patient with oral cavity squamous cell carcinoma, cancer stage T4N2M0 – peptides are taken into consideration in this work. Spectra were acquired in positive reflectron mode in the mass range between 800 and 4000  $m/z$ . The primary dataset consisted of 9,492 averaged spectra with 109,568 mass channels [ $m/z$ ]. Then, the spectra underwent several pre-processing steps: resampling (in order to unify mass channels across the dataset), baseline removal, TIC normalization, and alignment to the average spectrum based on the Fast Fourier Transform [11, 12]. After that, the Gaussian Mixture Model (GMM) approach was applied

to peak detection and spectra modeling [13]. In order to construct the mathematical model of an average spectrum, the GMM approach was applied and it turned out that the complete model consisted of 6,714 Gaussian components. Then, the neighboring components, which model right-skewness of spectral peaks, were merged, and components with relatively low abundance were filtered out. After all, the number of components has been reduced from 6,714 to 2,435 peaks [12]. After these steps, the peaks that were considered to belong to an isotopic envelope were annotated by an expert in the field of mass spectrometry.

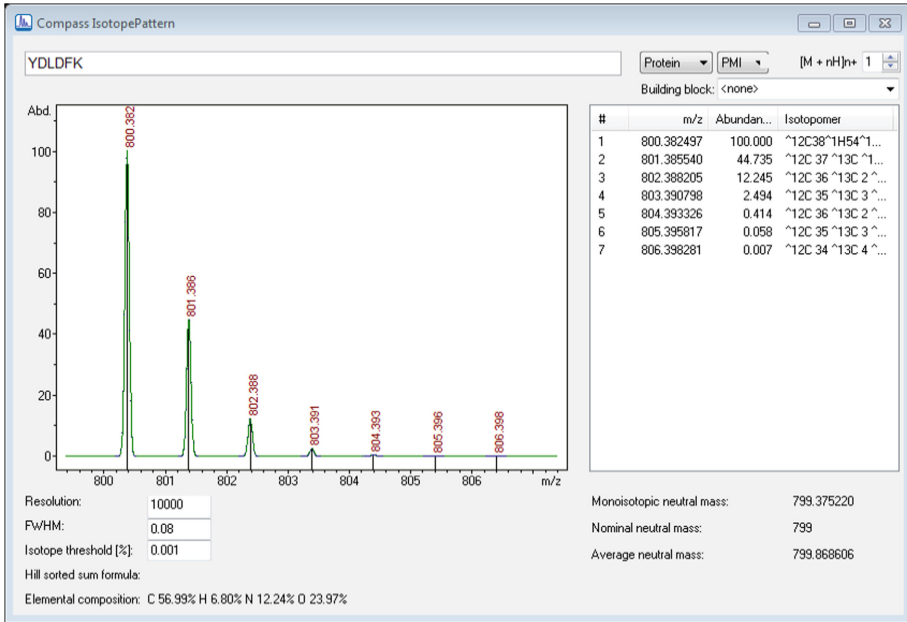
## 2.2 Idea Explanation

A spectral fragment is represented by one datum – peaks [14] (Fig. 1). For further analysis, the spectrum is to be considered as a set of peaks.



**Fig. 1.** Peptides spectrum fragment.

An isotopic envelope consists of the isotopes of one compound. An exemplary isotopic envelope for a peptide is shown in Fig. 2. It is the isotopic envelope of peptide, calculated with the Isotope Pattern Calculator, developed by Bruker®.



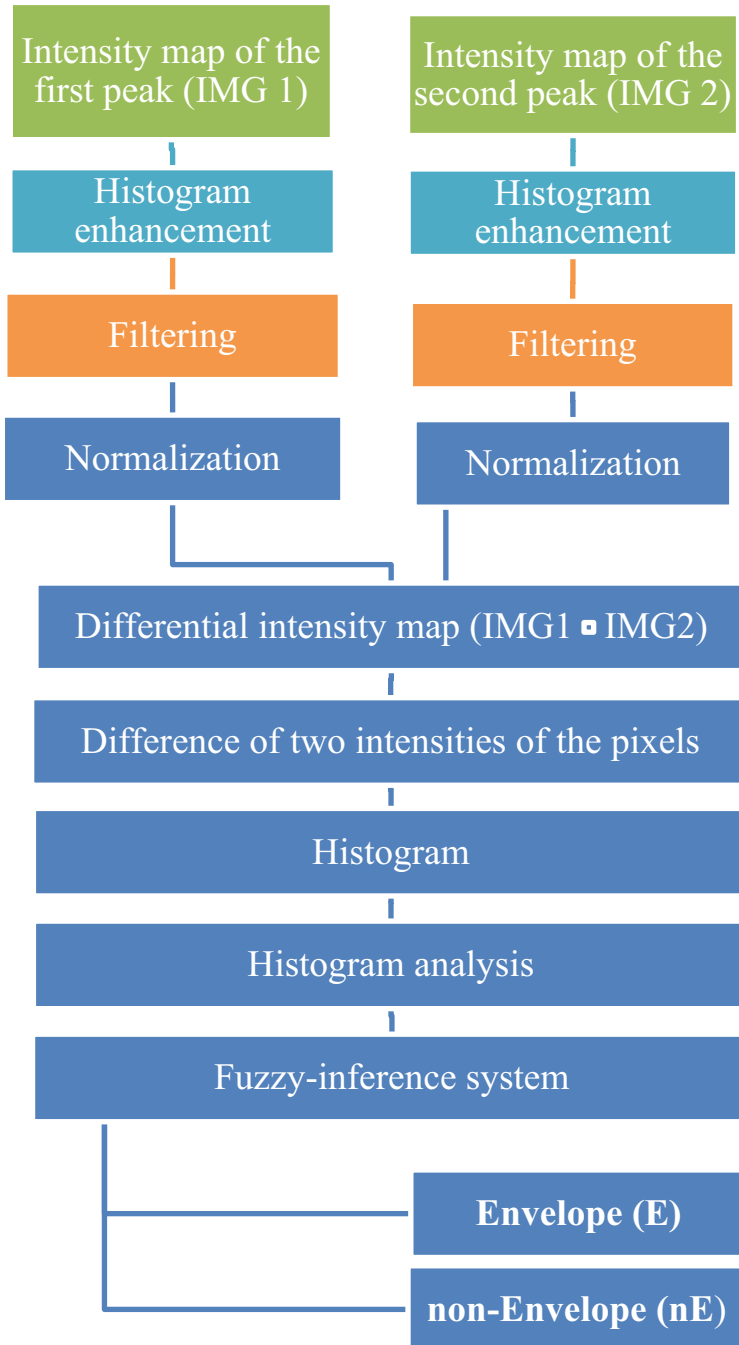
**Fig. 2.** An exemplary isotopic envelope for peptide YDLDFK.

The first step of analysis is based on Mamdani-Assilan fuzzy-inference system, presented in detail in [10]. The system is based on the following IF-THEN rules, which define if a peak is a member of an isotopic envelope [10]:

- “distance between two neighboring peaks is approximately equal to 1 Dalton
- the variance ratio of two neighboring peaks is approximately equal to 1
- amplitude ratio between two neighboring peaks is decreasing”.

To combine the rule outputs, the maximum-based aggregation method was applied [10]. “For defuzzification – the center of gravity method was used” [10].

After identifying candidate peaks that were prior identified with the fuzzy algorithm based on the Mamdani-Assilan fuzzy-inference system, the following algorithm for each peak pair is performed – Fig. 3.



**Fig. 3.** Algorithm pipeline.

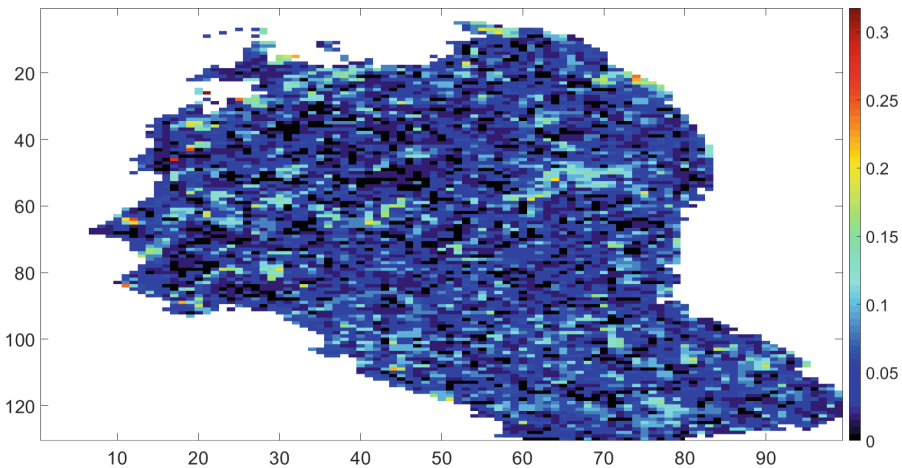


To annotate peaks that are included in an isotopic envelope, we propose the following algorithm:

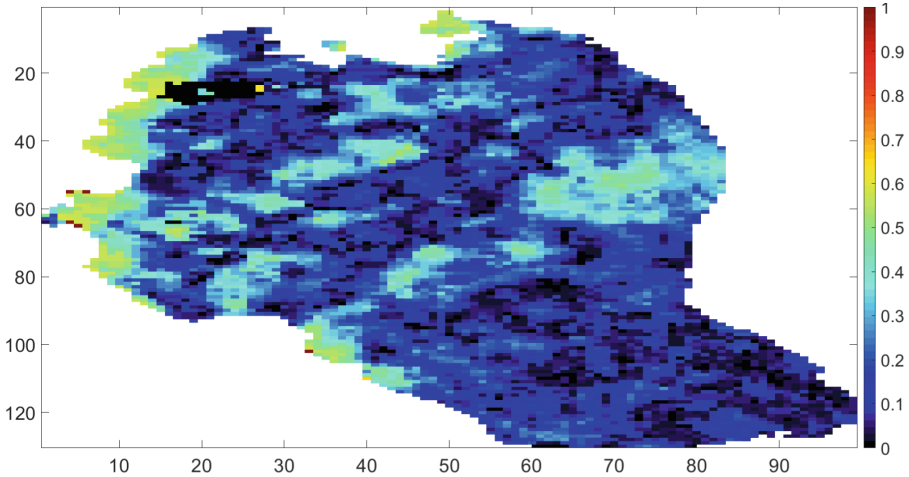
- **Peak visualization**

A separate spectrum is acquired for every tissue coordinate. The lateral resolution (raster width) of our MS images is 100  $\mu\text{m}$ . To visualize the spatial distribution of the peaks, each peak from a spectrum with a given  $m/z$  value is visualized as an image (map of intensities). It represents the intensities of peaks registered for specific  $m/z$  values throughout the whole tissue section. The image is constructed in the following way: intensities of the peaks are represented by different colors across the entire sample, in the original coordinate system. Then, in order to find the isotopic envelopes, for each peak of the pair, the aforementioned intensity map is constructed, that presents the spatial distribution of peptide abundance across the tissue sample.

After all peaks in the spectrum were visualized as images (intensity maps), pairwise differential images (Fig. 4, Fig. 5) have been created for each pair of peaks by subtracting normalized images.



**Fig. 4.** Differential image of peaks that are members of one isotopic envelope.

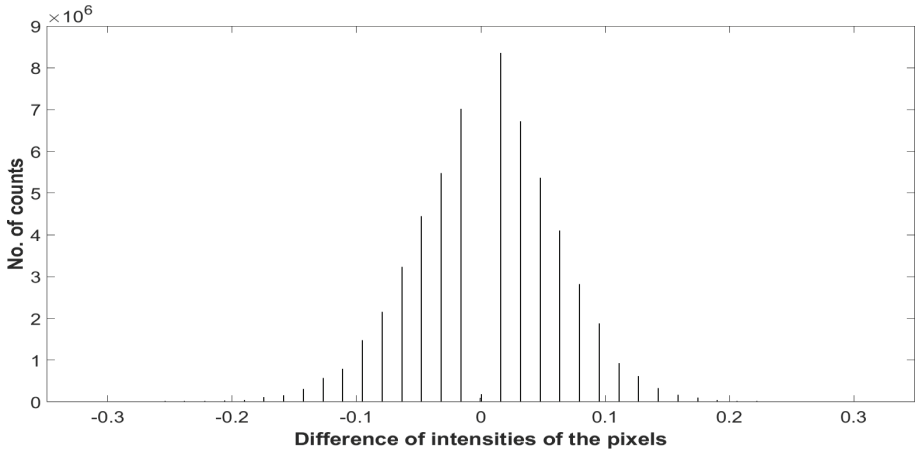


**Fig. 5.** Differential image of peaks that are not the members of one isotopic envelope.

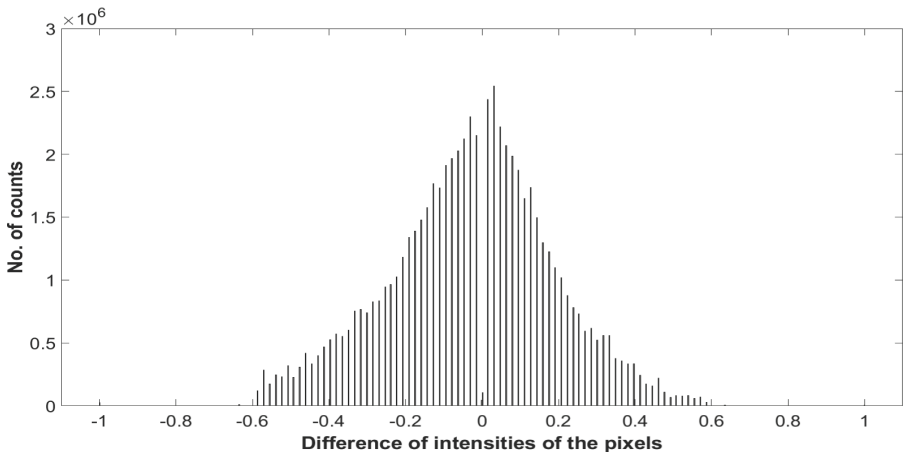
For each differential image, a comprehensive analysis of its intensity histogram and signal spatial distribution was performed, as shown in Fig. 3. We claim that if two peaks belong to the one isotopic envelope, their spatial distribution should be similar, and the obtained differential image should not have any internal structure. The above hypothesis could be verified by applying histogram analysis. The uniformity of the signal abundance means that we do not expect any particular structure in the image. The ‘salt and pepper’ structure is the most required one. Salt and pepper noise can be compared to sprinkling white and black dots on the image – it is a well-known process of image degradation [15]. To sum up, we assume that the peaks which are members of the particular isotopic envelope have the same spatial distribution, so that no structure should be visible in a differential image (Fig. 4). Therefore, we expect that for peaks that are not members of an isotopic envelope, a structure is visible in a differential image (Fig. 5).

### **Contribution of Peaks Intensities in Isotopic Envelope Defining**

The difference in intensities of the pixels was pairwise calculated and as a result, histograms of that difference were created (Fig. 6, Fig. 7).

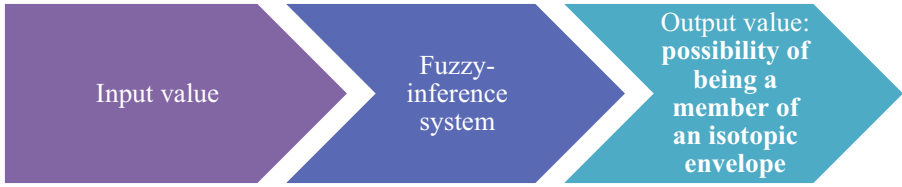


**Fig. 6.** A histogram presenting the difference in intensities of the pixels for peaks that are members of an isotopic envelope.

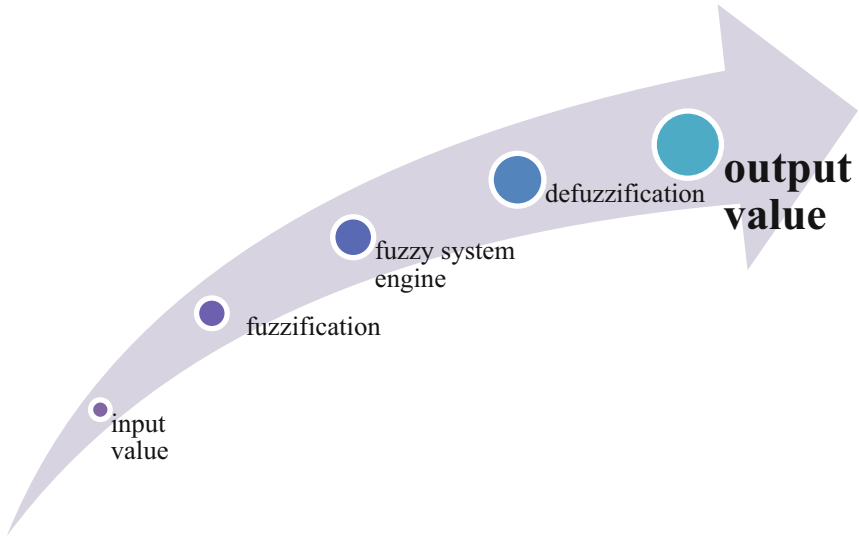


**Fig. 7.** A histogram presenting the difference in intensities of the pixels for peaks that are not the members of an isotopic envelope.

As it can be observed, a standard deviation of envelope peaks is lower than the standard deviation of peaks that are not members of an isotopic envelope. According to this, that feature was taken into consideration for further analysis. Based on that, the number of peaks included in the range  $\langle -0.2; 0.2 \rangle$  were calculated. Finally, the value [%] is an input for the fuzzy-inference system (Fig. 8).



**Fig. 8.** Decision making process pipeline.



**Fig. 9.** Fuzzy-inference system [16].

Then, the Sugeno fuzzy-inference system [17] was created, in Fig. 9 the process of calculating the output value was presented. During the fuzzification process, numerical input values are converted to fuzzy input values by the Gaussian combination membership function (Eq. 1, Fig. 10). The defuzzification method is based on a weighted average of data points [16, 17].

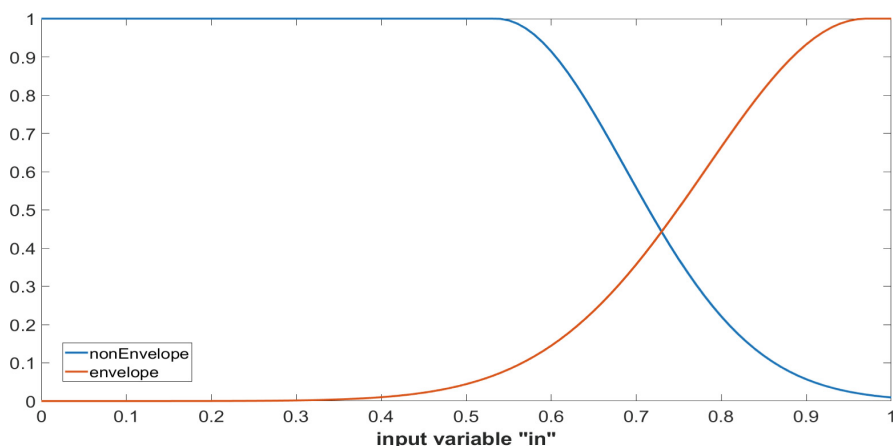
$$f(x; \sigma, c) = e^{-\frac{(x-c)^2}{2\sigma^2}}, \quad (1)$$

where:

- $\sigma$  – standard deviation,
- $c$  – mean for each Gaussian function.

Such a system provides the percentage possibility of being a member of an isotopic envelope.

Another approach for isotopic envelope identification is based on the fuzzy C-means clustering approach (FCM). In this method, a piece of data can belong to two or more clusters [18]. The fuzzy C-means segmentation of the differential image was performed



**Fig. 10.** Gaussian membership function.

by converting an input differential image into two segments by the fuzzy C-means algorithm [19].

### 3 Results

#### *Fuzzy-Inference System*

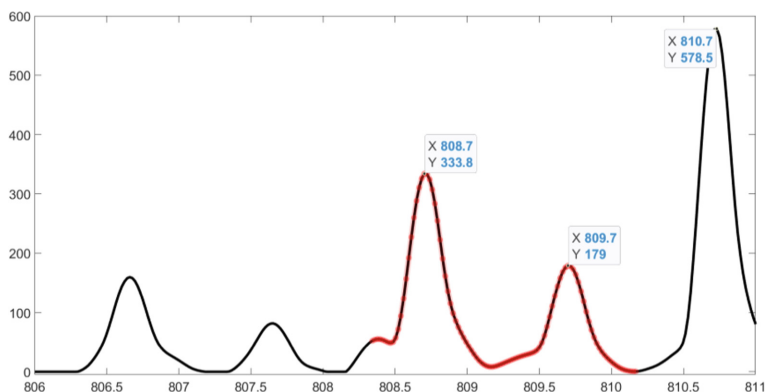
In Table 1 exemplary results for peptides' mass spectrum are presented. An output of the fuzzy-inference system is a possibility of isotopic envelope membership – it indicates whether a peak is a member of an isotopic envelope or not. As it can be observed, peaks that are members of an isotopic envelope are characterized by possibility values bigger than 50%, whereas the non-envelope ones have possibility values lower than 50%.

**Table 1.** Exemplary results for peptides' mass spectrum.

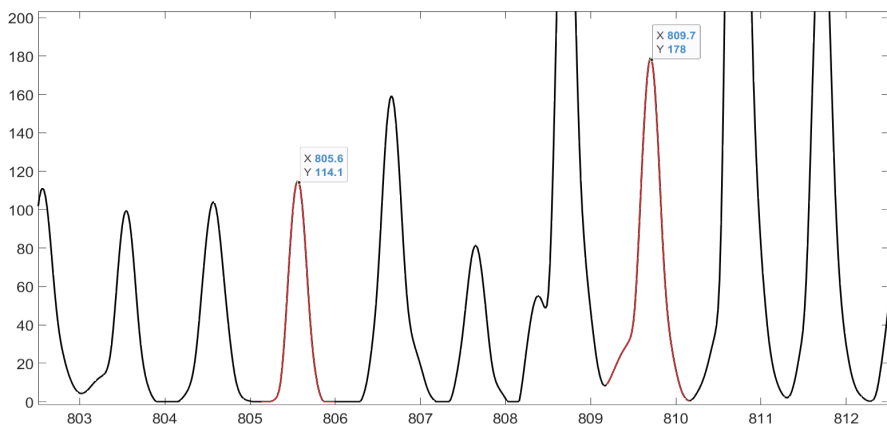
$m/z_1$	$m/z_2$	Possibility of isotopic envelope membership [%]
805.6	809.7	46 ( <b>Non-envelope</b> )
808.7	809.7	74.7 ( <b>Envelope</b> )
810.7	811.7	98.1 ( <b>Envelope</b> )
810.8	897.6	15.3 ( <b>Non-envelope</b> )
812.7	813.7	98.7 ( <b>Envelope</b> )
812.7	897.6	25.1 ( <b>Non-envelope</b> )
843.7	844.7	99 ( <b>Envelope</b> )

Obtained results have been checked on the mass spectrum, in Fig. 11 peaks included in the same isotopic envelope are marked in red, whereas in Fig. 12 peaks that are not

included in the same isotopic envelope were marked in red. Isotopic envelope members are characterized by the lower number of peaks within the range  $\langle -0.2; 0.2 \rangle$ . The reason is that peaks of one isotopic envelope in such a range of  $m/z$  values ( $\sim 800 - \sim 1000$  Da) follow such a pattern: the first peak has the highest intensity (monoisotopic peak), whereas the successive peaks represent  $\sim 45\%$  and  $\sim 12\%$  of the intensity of the first peak, respectively. According to that, the intensity histogram of peaks included in one envelope is denser within the range  $\langle -0.2; 0.2 \rangle$ .



**Fig. 11.** Peaks included in the same isotopic envelope.



**Fig. 12.** Peaks that are not members of the same isotopic envelope.

The obtained results were compared to results of an analysis of an average MSI spectrum performed by an experienced mass spectrometrists, who assessed whether a particular isotopic peak belonged to a given isotopic envelope based on the theoretical isotope pattern for a peptide with a given mass. The theoretical isotopic pattern for a peptide was obtained using the *Compass IsotopePattern Calculator* (Bruker®) taking into account the peptide sequence obtained in an LC-MALDI MSMS analysis of the tissue protein extract.

Fuzzy C-Means Differential Image Segmentation

In Fig. 13 and Fig. 14 exemplary results of the differential image segmentation based on the fuzzy C-means algorithm for the envelope and non-envelope members are presented, respectively. As it can be observed, in the envelope image there is no structure visible, whilst in the non-envelope image, the structure is clearly visible.

Envelope



Non-envelope



**Fig. 13.** Final segmentation after fuzzy C-means clustering.

**Fig. 14.** Final segmentation after fuzzy C-means clustering.

**Table 2.** Exemplary results of the differential image segmentation based on the fuzzy C-means clustering.

Envelope		Non-envelope	
Cluster center 1	Cluster center 2	Cluster center 1	Cluster center 2
2.6	32.4	7.0	71.7
2.1	23.9	7.9	86.1
2.2	23.6	7.7	81.0

In Table 2 exemplary results of the differential image segmentation based on fuzzy c-means clustering are presented. The cluster center is an arithmetic mean of all the data points that belong to the specific cluster. It can be observed that for the envelope, the cluster centers are characterized by significantly lower values in comparison to the non-envelope ones.

## 4 Conclusions

The proposed method is based on mass spectrum analysis from the spatial distribution point of view. Other aspects of the mass spectrum are not taken into consideration. According to that, the presented method can be used for any mass spectrum, no matter what type of mass spectrometry experiment it comes from. There are several existing methods for deisotoping, but they are usually dedicated to a specific type of experimental platform (for instance MS-Deconv [18], BPDA [19]) or type of molecule (lipids or peptides), for example, BPDA [19], YADA [20].

Nevertheless, there are some limitations to this work, since the proposed method is dedicated only to molecular imaging techniques and cannot be used in other proteome studies.

Finally, we conclude that the proposed algorithm for the automatic identification of the isotopic envelope is independent of the type of mass spectrometry experiment (MALDI, ESI, etc.) and of the type of molecules to be analyzed due to the fact that the only feature that is considered is a spatial distribution of the peaks from the mass spectrum. The basic idea that a proper analysis of the spatial distribution of the molecular abundance in the tissue sample allows distinguishing between the envelope and non-envelope peaks has been proven, and the obtained results seem to be promising.

**Acknowledgements.** We thank Prof. Jacek Łęski for comprehensive verification of fuzzy-inference system and Marta Gawin for sharing the expert knowledge in the field of mass spectrometry. This work was co-financed by European Union grant under the European Social Fund, project no. POWR.03.02.00-00-1029.

## References

1. Piehowski, P.D., Zhu, Y., Bramer, L.M., et al.: Automated mass spectrometry imaging of over 2000 proteins from tissue sections at 100- $\mu$ m spatial resolution. *Nat. Commun.* **11**(8) (2020)
2. Cornett, D.S., Reyzer, M.L., Chaurand, P., Caprioli, R.M.: MALDI imaging mass spectrometry: molecular snapshots of biochemical systems. *Nat. Methods* **4**, 828–833 (2007)
3. Schöne, C., Höfler, H., Walch, A.: MALDI imaging mass spectrometry in cancer research: combining proteomic profiling and histological evaluation. *Clin. Biochem.* **46**, 539–545 (2013)
4. Kriegsmann, J., Kriegsmann, M., Casadonte, R.: MALDI TOF imaging mass spectrometry in clinical pathology: a valuable tool for cancer diagnostics (Review). *Int. J. Oncol.* **46**(3), 893–906 (2015)
5. Caprioli, R.M., Farmer, T.B., Gile, J.: Molecular imaging of biological samples: localization of peptides and proteins using MALDI-TOF MS. *Anal. Chem.* **69**(23), 4751–4760 (1997)
6. Pietrowska, M., et al.: Molecular profiles of thyroid cancer subtypes: classification based on features of tissue revealed by mass spectrometry imaging. *Biochim. Biophys. Acta. – Proteins Proteom.* **1865**(7), 837–845 (2017)
7. Tuma, R.S.: MALDI-TOF mass spectrometry: getting a feel for how it works. *Oncol. Times* **25**(19), 26 (2003)
8. IUPAC.: *Compendium of Chemical Terminology*, 2nd edn. (the “Gold Book”). Compiled by A. D. McNaught and A. Wilkinson. Blackwell Scientific Publications, Oxford (1997). Online version (2019) created by S. J. Chalk. ISBN 0-9678550-9-8



9. Eidhammer, I., Flikka, K., Martens, L., Mikalsen, S.O.: *Computational Methods for Mass Spectrometry Proteomics*. Wiley (2007)
10. Glodek, A., Polańska, J.: Method for mass spectrometry spectrum deisotoping based on fuzzy inference systems. *Math. Appl.* **46**(1), 77–86 (2018)
11. Pietrowska, M., et al.: Comparison of peptide cancer signatures identified by mass spectrometry in serum of patients with head and neck, lung and colorectal cancers: association with tumor progression. *Int. J. Oncol.* **40**, 148–156 (2012)
12. Mrukwa, G., Drażek, G., Pietrowska, M., Widłak, P., Polańska, J.: A novel divisive iK-means algorithm with region-driven feature selection as a tool for automated detection of tumour heterogeneity in MALDI IMS experiments. In: *4th International Work-Conference on Bioinformatics and Biomedical Engineering, IWBBIO 2016, Bioinformatics and Biomedical Engineering LNCS*, vol. 9656, pp. 113–124. Springer, Heidelberg (2016)
13. Polański, A., Marczyk, M., Pietrowska, M., Widłak, P., Polańska, J.: Signal partitioning algorithm for highly efficient Gaussian mixture modeling in mass spectrometry. *PLoS ONE* **10**, e0134256 (2015)
14. Eidhammer, I., Flikka, K., Martens, L., Mikalsen, S.O.: *Computational Methods for Mass Spectrometry Proteomics*. Wiley (2007)
15. Bovik, A.: *Handbook of Image and Video Processing*, 2nd edn. Academic Press (2005)
16. Ross, T.J.: *Fuzzy Logic with Engineering Applications*. Wiley (2017)
17. Czogała, E., Łęski, J.: *Fuzzy and neuro-fuzzy intelligent systems*. Physica-Verlag, Heidelberg (2000)
18. Christ, M.C.J., Parvathi, R.M.S.: Fuzzy c-means algorithm for medical image segmentation. In: *2011 3<sup>rd</sup> International Conference on Electronics Computer Technology*, pp. 33–36 (2011)
19. Camilus, S.: *Fuzzy c-Means Segmentation*. MATLAB Central File Exchange (2022)



# Receptor Tyrosine Kinase KIT: A New Look for an Old Receptor

Julie Ledoux<sup>(✉)</sup> and Luba Tchertanov

Université Paris-Saclay, ENS Paris-Saclay, Centre Borelli, 4 Avenue des Sciences,  
91190 Gif-sur-Yvette, France  
[julie.ledoux@ens-paris-saclay.fr](mailto:julie.ledoux@ens-paris-saclay.fr)

**Abstract.** RTK KIT regulates a variety of crucial cellular processes via its cytoplasmic (CD) domain composed of the tyrosine kinase domain crowned by highly flexible domains - juxtamembrane region, kinase insert domain and C-tail, key recruitment regions for downstream signalling proteins. We generated 3D models of the full-length CD attached to the transmembrane helix to prepare a structural basis for characterization of interactions of native KIT and its oncogenic mutant D816V with signalling proteins (KIT INTERACTOME). Generic models of native KIT in inactive state and constitutively activated KIT mutant D816V were studied by molecular dynamics simulation under conditions mimicking the natural environment of KIT. With the accurate atomistic description of the multidomain KIT dynamics, we explained the role of somatic mutation D816V on structural and dynamical properties of RTK KIT focusing on its intrinsic (intra-domain) and extrinsic (inter-domain) disorder. Conformational ensembles of native and mutated KIT were represented through free energy landscapes. Strongly coupled movements within each domain and between distant domains of KIT prove the functional interdependence of these regions, described as allosteric regulation, a phenomenon widely observed in many proteins.

**Keywords:** Receptor tyrosine kinase · RTK · Full-length KIT cytoplasmic region · Intrinsically disordered regions · Somatic mutation d816v · Phosphotyrosine · Modelling · Molecular dynamics · Conformational transition · Allosteric regulation and deregulations · Free energy landscape

Receptor tyrosine kinases (RTKs) control various signalling pathways in cells. Their remarkable conformational plasticity enables the specific recognition of many molecules such as ligands, substrates or proteins. In solution, RTKs are at equilibrium between different conformations ranging from an inactive auto-inhibited state to a fully active state. Ligand-induced activation of RTKs leads to recruitment and activation of multiple downstream signalling proteins which alter the expression of genes governing cell physiology [10]. Explicit elucidation of signalling events is an important and unresolved problem in cell biology. The initiation of these cascade-like processes involves different domains of

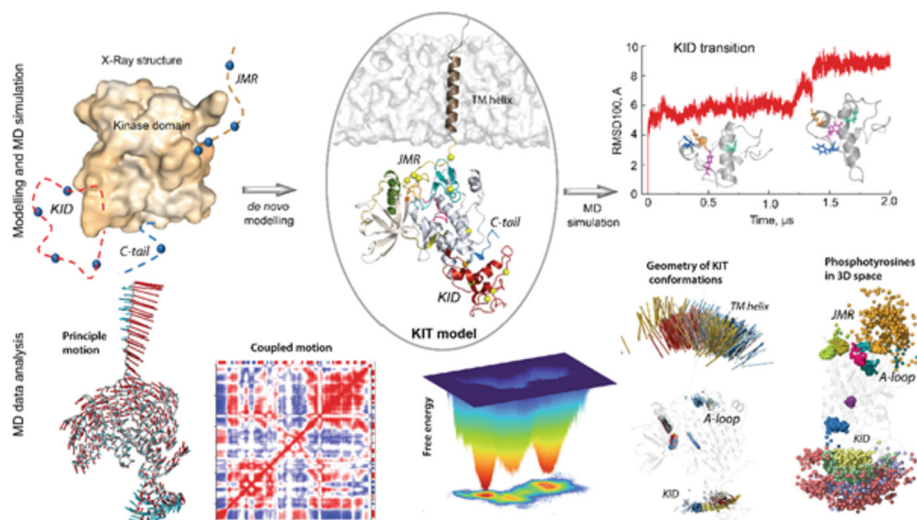
RTK, each performs specific actions, finely concerted by a regulated allosteric mechanism controlling all functional biological processes [4, 12]. Dysregulation of RTK-controlled signalling pathways, prompted generally by gain-of-function mutations, underlies abnormal cell development leading to tumorigenesis.

Focusing on the RTK KIT, an important target in oncology [3, 11], we will discuss this RTK as a key regulator of intracellular signalling mediated by regions possessing multiple phosphorylation sites: juxtamembrane region (JMR), kinase insert domain (KID), activation loop (A-loop) and C-terminal tail. Since these regions are very flexible or disordered, their properties are not yet well understood. As KIT is a multi-functional protein, its different regions regulate catalytic processes and/or events that activate and control the signalling cascade. To complete the functions required in more than one region, these regions should be directly or collaterally coupled.

We generated the 3D model of the inactive full-length native KIT (KIT WT) attached to the transmembrane helix (Fig. 1), to (i) prepare a structural basis for the characterisation of interconnections between functional regions - JMR, tyrosine kinase (TK) domain, KID and C-tail -, and (ii) establish the interactions of KIT with its signalling proteins. Then, we investigated this model by molecular dynamics (MD) simulation in conditions mimicking its natural environment. We suggested that such atomistic description of KIT will fully elucidate structural and dynamical properties of its different functional regions. To the best of our knowledge, we have presented for the first time a model of a full-length cytoplasmic region of an RTK KIT attached to a transmembrane helix and its molecular dynamics simulations under conditions that mimic its natural environment [9].

Analysis of the simulation data (three 2- $\mu$ s MD trajectories) put in evidence that the multidomain RTK KIT is a modular protein consisting of a quasi-stable TK domain crowned by at least four intrinsically disordered (ID) regions - JMR, KID, A-loop and C-tail. These ID regions belong to two types - the very elongated (extended) and poorly folded regions (JMR, A-loop and C-tail), and the globule-like (collapsed) KID having a high level of the helical structures. KIT ID regions contain transient structures (helices and  $\beta$ -strands) and their local architecture displays various degrees of compaction and elongation. Therefore, the structure of each ID domain of KIT represents a very complex mixture of a broad variety of differently folded conformations which describe a reversible folding-unfolding process, specific for each ID domain. In particular, the KID, composed of transient helices linked by coils, is the most disordered domain in respect to other KIT domains, but shows a globule-like shape stabilised by non-covalent interactions [5, 7].

Also, the inherently disordered KID shows different positions derived from two types of motions - linear (translation) and angular (rotation) displacement - regarding the stable TK domain. The elongated regions (JMR, A-loop and C-tail) show rather local disorder as evidenced by alternating positions of their short segments relative to the stable TK domain.



**Fig. 1. The 3D model of full-length KIT in native and constitutively active states attached to the transmembrane helix.** Modelling and study by MD simulations (top panel) analysed to (i) characterise the global motions and coupling, (ii) to represent the MD conformations as the free energy landscape, (iii) to describe the relative positions of the KIT functional regions and the position of phosphotyrosine residues.

The two-level disorder (intrinsic and extrinsic) provides high conformational variability of KIT and supplies the high adaptability of JMR, KID and C-tail required for the scaffolding (docking sites) and recruitment of different protein partners of KIT and accomplishes the tight regulation of cellular processes. Consequently, the overall structure of KIT represents a continuous spectrum of conformations with a different degree and depth of disorder, thereby generating a complex protein structural space. It is partially reflected by free energy landscapes (FEL) lacking a unique global deep minimum as typically observed in ordered proteins. Such energy landscapes, with two local minima joined by a ‘flattened plateau’ containing the intermediate conformations, show that KIT is extremely sensitive to different environmental changes (e.g. phosphorylation) that can alter its FEL in different ways. We suggest that JMR, displaced from its packed auto-inhibited position upon the SCF-induced activation of KIT, will achieve higher levels of disorder, and therefore a higher level of adaptability for the recruitment of signalling proteins. We suggest that JMR, displaced from its packed auto-inhibited position upon the SCF-induced activation of KIT, will achieve higher levels of disorder, and therefore a higher level of adaptability for the recruitment of signalling proteins.

Since ID domains are multiple in RTK KIT, does the disorder/order of one domain depend on the disorder/order of other remote regions? As evidenced by the cross-correlations, the highly coupled motions of distant regions of KIT suggest their functional dependence, classified as allosteric regulation, phenomenon

largely observed in many proteins. In particular, the coupling motions within the TK domain reflect the allosteric regulation of kinase function which is well-described for different non-receptor and receptor tyrosine kinases [1]. The coupled/uncoupled motions of A-loop and JMR were described through their allosteric communication in the wild-type KIT and oncogenic mutants [2,6], although this characterization was made using limited structural data with only partially resolved JMR.

To characterise function-related mutation-induced effects in KIT, we studied KID D816V mutant represented by the most realistic the full-length model [8]. Even through KID D816V structure is highly similar to native KIT, motions of these two proteins are quite different. Indeed, KID D816V mutant shows significantly increasing coupling of intra- and inter-domain motions. Moreover, KIT communication pathways network is strongly different in two proteins. Such multipanel characterisation has explained more explicitly the role of each region in maintaining KIT inactive and constitutively active state and/or as a signalling support for the phosphotyrosines-containing regions, and has established relationships between them.

This dynamic model of allosterically regulated KIT in two states is the first step for the reconstruction of its (i) INTERACTOME, composed of a set of KIT complexes with its signalling proteins, and its (ii) DYNASOME, constituting of an ensemble of KIT intermediate conformations before, over and after post-transduction processes upon its physiological and pathological context. We also suggested that KIT in inactive state encodes all properties of the active protein and post-transduction events. Such hypothesis echoes high dependences of the INTERACTOME to the DYNASOME.

## References

1. Ahuja, L.G., Aoto, P.C., Kornev, A.P., Veglia, G., Taylor, S.S.: Dynamic allostery-based molecular workings of kinase: peptide complexes. *Proc. Natl. Acad. Sci.* **116**(30), 15052–15061 (2019). <https://doi.org/10.1073/pnas.1900163116>. <http://www.pnas.org/lookup/doi/10.1073/pnas.1900163116>
2. Allain, A., Chauvot de Beauchêne, I., Langenfeld, F., Guarracino, Y., Laine, E., Tchertanov, L.: Allosteric pathway identification through network analysis: from molecular dynamics simulations to interactive 2D and 3D graphs. *Faraday Discuss.* **169**, 303–321 (2014). <https://doi.org/10.1039/C4FD00024B>. <http://xlink.rsc.org/?DOI=C4FD00024B>
3. Gilreath, J., Tchertanov, L., Deininger, M.: Novel approaches to treating advanced systemic mastocytosis. *Clin. Pharmacol. Adv. Appl.* **11**, 77–92 (2019). <https://doi.org/10.2147/CPAA.S206615>. <https://www.dovepress.com/novel-approaches-to-treating-advanced-systemic-mastocytosis-peer-reviewed-article-CPAA>
4. Gunasekaran, K., Ma, B., Nussinov, R.: Is allostery an intrinsic property of all dynamic proteins? *Proteins Struct. Function Bioinform.* **57**(3), 433–443 (2004). <https://doi.org/10.1002/prot.20232>. <https://onlinelibrary.wiley.com/doi/10.1002/prot.20232>

5. Inizan, F., Hanna, M., Stolyarchuk, M., Chauvot de Beauchêne, I., Tchertanov, L.: The first 3D model of the full-length KIT cytoplasmic domain reveals a new look for an old receptor. *Sci. Rep.* **10**(1), 5401 (2020). <https://doi.org/10.1038/s41598-020-62460-7>. <http://www.nature.com/articles/s41598-020-62460-7>
6. Laine, E., Auclair, C., Tchertanov, L.: Allosteric communication across the native and mutated KIT receptor tyrosine kinase. *PLoS Comput. Biol.* **8**(8), e1002661 (2012). <https://doi.org/10.1371/journal.pcbi.1002661>. <https://dx.plos.org/10.1371/journal.pcbi.1002661>
7. Ledoux, J., Trouvé, A., Tchertanov, L.: Folding and intrinsic disorder of the receptor tyrosine kinase KIT insert domain seen by conventional molecular dynamics simulations. *Int. J. Mol. Sci.* **22**(14), 7375 (2021). <https://doi.org/10.3390/ijms22147375>. <https://www.mdpi.com/1422-0067/22/14/7375>
8. Ledoux, J., Trouvé, A., Tchertanov, L.: Does cancerogenic mutation D816V influence modular receptor tyrosine kinase KIT intrinsic disorder? Paper in preparation (2022)
9. Ledoux, J., Trouvé, A., Tchertanov, L.: The inherent coupling of intrinsically disordered regions in the multidomain receptor tyrosine kinase KIT. *Int. J. Mol. Sci.* **23**(3), 1589 (2022). <https://doi.org/10.3390/ijms23031589>. <https://www.mdpi.com/1422-0067/23/3/1589>
10. Lemmon, M.A., Schlessinger, J.: Cell signaling by receptor tyrosine kinases. *Cell* **141**(7), 1117–1134 (2010). <https://doi.org/10.1016/j.cell.2010.06.011>. <https://linkinghub.elsevier.com/retrieve/pii/S0092867410006653>
11. Pham, D.D.M., Guhan, S., Tsao, H.: KIT and melanoma: biological insights and clinical implications. *Yonsei Med. J.* **61**(7), 562 (2020). <https://doi.org/10.3349/ymj.2020.61.7.562>. <https://eymj.org/DOIx.php?id=10.3349/ymj.2020.61.7.562>
12. Wodak, S.J., et al.: Allostery in its many disguises: from theory to applications. *Structure* **27**(4), 566–578 (2019). <https://doi.org/10.1016/j.str.2019.01.003>. <https://linkinghub.elsevier.com/retrieve/pii/S0969212619300036>



# Human Vitamin K Epoxide Reductase as a Target of Its Redox Protein

Julie Ledoux<sup>(✉)</sup>, Maxim Stolyarchuk, and Luba Tchertanov

Université Paris-Saclay, ENS Paris-Saclay, Centre Borelli, 4 Avenue des Sciences,  
91190 Gif-sur-Yvette, France  
[julie.ledoux@ens-paris-saclay.fr](mailto:julie.ledoux@ens-paris-saclay.fr)

**Abstract.** Human Vitamin K epoxide reductase (hVKORC1) is a key enzyme to reduce vitamin K. Such function requires activation of the enzyme by a redox partner delivering reducing equivalents through thiol-disulphide exchange reactions. The activation process represents a first and less studied step in hVKORC1 vital cycle, involving the oxidised luminal loop (L-loop) and a reduced thioredoxin protein (Trx), which is yet undefined for hVKORC1. A careful *in silico* study, based on molecular dynamic (MD) simulations of hVKORC1 in oxidised state, and a comparative analysis of four Trx proteins - protein disulphide isomerase (PDI), endoplasmic reticulum oxidoreductase (ERp18), thioredoxin-related transmembrane protein 1 (Tmx1) and thioredoxin-related transmembrane protein 4 (Tmx4)), viewed as the most probable reducers of hVKORC1 - in their sequence, secondary and tertiary structure, dynamics, intra-protein interactions and composition of the surface exposed to the target - provided the identification of putative recognition/binding sites on each isolated protein. PDI was suggested as the most probable hVKORC1 partner. By probing the alternative orientation of PDI with respect to hVKORC1, two PDI-VKOR models were proposed and one of them considered as precursor for thiol-disulphide exchange reactions.

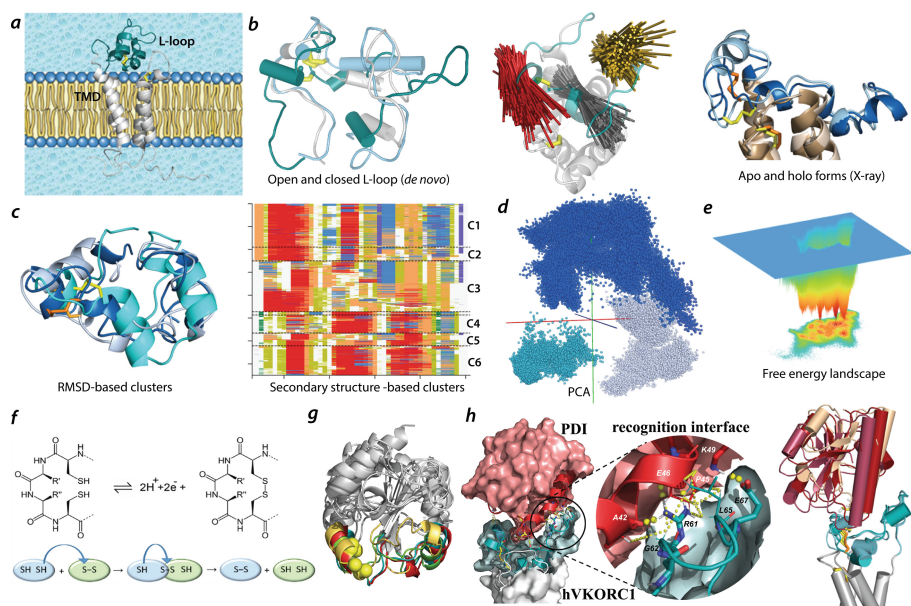
**Keywords:** hVKORC1 · Trx-fold redox proteins · Protein folding · Intrinsic disorder · Modular protein · Molecular recognition · Thiol-disulphide exchange · Protein-protein interactions · PDI-hVKORC1 complex · 3D modelling · Molecular dynamics

The human vitamin K epoxide reductase (hVKOR) hVKORC1 is an endoplasmic reticulum (ER)-resident transmembrane protein reducing vitamin K inside a membrane-embedded cysteine-containing redox centre [9]. Such activity requires the cooperation of VKOR with a redox partner delivering reducing equivalents through thiol-disulphide exchange reactions, involving a disulphide bridge from the extended luminal loop (L-loop) of VKOR [6]. The activation process represents a first and less studied step in VKOR vital cycle. The physiological redox partner of hVKORC1 remains uncertain. Four human redoxin proteins (Trx) - protein disulphide isomerase (PDI), endoplasmic reticulum oxidoreductase



(ERp18), thioredoxin-related transmembrane protein 1 (Tmx1) and thioredoxin-related transmembrane protein 4 (Tmx4) - were suggested as the most probable H-donors of hVKORC1 [7]. In addition, the structure of hVKORC1 L-loop is not credibly characterised. Consequently, deciphering the molecular origins of hVKORC1 recognition by an unknown redox protein is not a trivial task.

We suggested that an accurate *in silico* study of Trxs and hVKORC1 as isolated proteins would provide useful information for the development of putative Trx-VKOR complexes. Quantitative metrics and qualitative estimations can shed new light on the target (hVKORC1) features and peculiarities of redox proteins (Trx). Such information may help in predicting (i) the protein fragments participating in VKORC1 recognition by a Trx and (ii) the most probable partner of VKORC1.



**Fig. 1. 3D model of VKORC1 and its complex with redox protein (a)** VKORC1 inserted into membrane and surrounded by water molecules. The helically folded L-loop shows structural and conformational disorder in *de novo* model (left and middle) and in X-ray structures. (b) Clustering of L-loop conformations using RMSD and secondary structures. (d) Projection of MD conformations on the three principal components determined by principal component analysis (PCA). (e) Free energy landscape of L-loop conformations defined on two primary reaction coordinates, RMSD and radius of gyration (Rg). (f) Mechanism of disulphide exchange between Trx and a target. (g) Superimposed 3D structures of Trx-fold proteins with cysteine residue as yellow balls. Regions that are potentially involved in target recognition and/or electron transfer reaction are differentiated by colour. (h) Modelling of human PDI-hVKORC1 complex (left) and reproduced results by protein-protein docking (HADDOCK) (right).



First, conformational features of hVKORC1 and L-loop, the principal platform of hVKORC1 for Trx recognition, scaffolding and intermolecular thiol-disulphide exchange reactions, were characterised by extended molecular dynamics simulations (MD) of a de novo model [1,8] and crystallographic structures [4] of the enzyme in oxidized state. This study clearly showed that (i) L-loop is an intrinsically disordered region, and (ii) hVKORC1 is a modular protein composed of the structurally stable transmembrane domain (TMD) crowned by the disordered L-loop [2] (Fig. 1a–e). Indeed, the structurally well conserved TM helices, varied slightly only at their ends, show a cooperative drift typical for transmembrane domains rigidified by the stable non-covalent interactions. In contrast, L-loop exhibits an unstable helical fold represented by reversible transient - and 310-helices linked by flexible coils, offering L-loop a great conformational diversity, from compact ‘globule-like’ shape (closed form) to extended (open form) (Fig. 1b–c).

Such modular architecture of hVKORC1 provides (i) excellent conformational plasticity required for specific adaptation over recognition by redox protein, activation process and catalysis, and (ii) easy and exact reproducibility of hVKORC1 metastable intermediates during repeated enzymatic cycles. Those qualities are strictly required for hVKORC1 activities [5]. Structure and conformations of the disordered L-loop are better described in terms of free energy than conventional methods such as clustering.

Secondly, focusing on Trx-fold proteins, probable hVKORC1 redox partners, we found that, despite similar architecture, each redox partner has its own sequence-dependent dynamical features. Further analysis identified PDI as the most probable redox partner of hVKORC1 (Fig. 1f–h) [8]. By probing PDI alternative orientations with respect to hVKORC1, two models of noncovalent complex were proposed. One of them was considered as functionally related model and postulated as the first precursor to probe thiol-disulphide exchange reaction. This predicted complex, formed by hVKORC1 and PDI, was further reproduced by docking trials (protein-protein docking with HADDOCK) [2].

Third, results obtained for hVKORC1 simulated in different environment (water/membrane) and simulation methods (conventional and accelerated) indicated that, for *in silico* study of hVKORC1 and its complexes, the membrane is probably not necessary, and the cleaved L-loop, simulated as isolated polypeptide, reflects its properties when fused to the transmembrane domain. Therefore, it may be used to study hVKORC1 recognition by its redox protein. Extension of these conclusions for experimental studies of hVKORC1 requires their empirical validation.

Finally, all these findings lead to modelling of PDI-hVKORC1 complexes assembled during a biomolecular proton-electron transfer reaction between PDI and hVKORC1 - a hydrogen-bonded ‘precursor complex’ formed prior to proton-electron (hydrogen-atom) transfer, intermediate covalent complex characterising transient state, and ‘successor complex’, corresponding to post proton-electron (hydrogen-atom) transfer [3]. Such 3D models of PDI-hVKORC1 complexes will lay the structural basis for comprehensive description of hVKORC1 activation mechanisms.

## References

1. Chatron, N., et al.: Identification of the functional states of human vitamin K epoxide reductase from molecular dynamics simulations. *RSC Adv.* **7**(82), 52071–52090 (2017). <https://doi.org/10.1039/C7RA07463H>. <http://xlink.rsc.org/?DOI=C7RA07463H>
2. Ledoux, J., Stolyarchuk, M., Bachelier, E., Trouvé, A., Tchertanov, L.: Human vitamin K epoxide reductase as a target of its redox protein. *Int. J. Mol. Sci.* **23**(7), 3899 (2022). <https://doi.org/10.3390/ijms23073899>. <https://www.mdpi.com/1422-0067/23/7/3899>
3. Ledoux, J., Stolyarchuk, M., Tchertanov, L.: Modelling of precursor and successor complexes and transition state assembled in a biomolecular proton-electron transfer reaction between PDI and hVKORC1. Paper in preparation (2022)
4. Liu, S., Li, S., Shen, G., Sukumar, N., Krezel, A.M., Li, W.: Structural basis of antagonizing the vitamin K catalytic cycle for anticoagulation. *Science* (New York N.Y.) **371**(6524), eabc5667 (2021). <https://doi.org/10.1126/science.abc5667>
5. Ma, B., Kumar, S., Tsai, C.J., Hu, Z., Nussinov, R.: Transition-state ensemble in enzyme catalysis: possibility, reality, or necessity? *J. Theor. Biol.* **203**(4), 383–397 (2000). <https://doi.org/10.1006/jtbi.2000.1097>. <https://linkinghub.elsevier.com/retrieve/pii/S002251930091097X>
6. Rishavy, M.A., Usabalieva, A., Hallgren, K.W., Berkner, K.L.: Novel insight into the mechanism of the vitamin K oxidoreductase (VKOR). *J. Biol. Chem.* **286**(9), 7267–7278 (2011). <https://doi.org/10.1074/jbc.M110.172213>. <https://linkinghub.elsevier.com/retrieve/pii/S0021925820519212>
7. Schulman, S., Wang, B., Li, W., Rapoport, T.A.: Vitamin K epoxide reductase prefers ER membrane-anchored thioredoxin-like redox partners. *Proc. Natl. Acad. Sci.* **107**(34), 15027–15032 (2010). <https://doi.org/10.1073/pnas.1009972107>. <http://www.pnas.org/cgi/doi/10.1073/pnas.1009972107>
8. Stolyarchuk, M., Ledoux, J., Maignant, E., Trouvé, A., Tchertanov, L.: Identification of the primary factors determining the specificity of human VKORC1 recognition by thioredoxin-fold proteins. *Int. J. Mol. Sci.* **22**(2), 802 (2021). <https://doi.org/10.3390/ijms22020802>. <https://www.mdpi.com/1422-0067/22/2/802>
9. Tie, J.K., Stafford, D.W.: Structure and function of vitamin K epoxide reductase. *Vitam. Horm.* **78**, 103–130 (2008). [https://doi.org/10.1016/S0083-6729\(07\)00006-4](https://doi.org/10.1016/S0083-6729(07)00006-4)



# A Distance Geometry Procedure Using the Levenberg-Marquardt Algorithm and with Applications in Biology but Not only

Douglas S. Gonçalves<sup>1</sup>(✉) and Antonio Mucherino<sup>2</sup>

<sup>1</sup> Department of Mathematics, UFSC, Florianópolis, SC, Brazil  
douglas.goncalves@ufsc.br

<sup>2</sup> IRISA, University of Rennes 1, Rennes, France  
antonio.mucherino@irisa.fr

**Abstract.** We revisit a simple, yet capable to provide good solutions, procedure for solving the Distance Geometry Problem (DGP). This procedure combines two main components: the first one identifying an initial approximated solution via semidefinite programming, which is thereafter projected to the target dimension via PCA; and another component where this initial solution is refined by locally minimizing the Smooth STRESS function. In this work, we propose the use of the projected Levenberg-Marquardt algorithm for this second step. In spite of the simplicity, as well as of its heuristic character, our experiments show that this procedure is able to exhibit good performances in terms of quality of the solutions for most of the instances we have selected for our experiments. Moreover, it seems to be promising not only for the DGP application arising in structural biology, which we considered in our computational experiments, but also in other ongoing studies related to the DGP and its applications: we finally provide a general discussion on how extending the presented ideas to other applications.

## 1 Introduction

Let  $G = (V, E, d)$  be a simple weighted undirected graph, where the weight function  $d$  maps every edge of the graph to a given distance value. We suppose that a unique numerical label  $i \in \{1, 2, \dots, |V|\}$  is associated to every vertex in  $V$ , so that a vertex ordering is implicitly given. The focus of this article is a geometric problem having several real-life applications [12, 15]:

**Definition 1.1.** *Given the graph  $G$  and a dimension  $K > 0$ , the Distance Geometry Problem (DGP) asks whether there exists any realization  $x : V \rightarrow \mathbb{R}^K$  such that the following distance constraints are satisfied:*

$$\forall \{i, j\} \in E, \quad \|x_i - x_j\| = d_{ij}, \quad (1)$$

where  $\|\cdot\|$  is the Euclidean norm and  $x_i = x(i)$ .

Throughout this article, we will suppose that the considered problem instances admit at least one solution, which we will refer to as “valid realizations”.

We remark that, in several applications, such as the one arising in the context of structural biology [5], sensor network localization [4], or even in computer graphics [11], the distance information cannot be provided with high precision. Most likely, instead of having one precise distance value  $d_{ij}$ , approximated lower and upper bounds are actually provided for most of the involved distances. Let us suppose therefore that our weight function  $d$  in  $G$  does not provide a single real number, but rather a pair of real numbers,  $\underline{d}_{ij}$  and  $\bar{d}_{ij}$  for every  $\{i, j\} \in E$ , such that  $\underline{d}_{ij} < \bar{d}_{ij}$ . In order to take these interval distances into consideration, we introduce new variables  $y$  indexed on the edge set  $E$ , and modify the problem in Eq. (1) as follows:

$$\forall \{i, j\} \in E, \quad \begin{cases} \|x_i - x_j\|^2 - y_{ij} = 0, \\ \underline{d}_{ij}^2 \leq y_{ij} \leq \bar{d}_{ij}^2. \end{cases} \quad (2)$$

In spite of the current large efforts of the research community in finding new and efficient solution methods to the DGP, a general method has not been devised yet. In this work, we focus our attention on a rather simple procedure, which is basically composed by two main components: (i) the generation of an *initial* realization that we can expect to be in a relatively small neighborhood of a DGP solution; (ii) a refinement step: from the found initial realization, we locally minimize the sum of squared constraint violations, with the aim of identifying a better approximation of the DGP solution. In particular, to tackle part (i), we solve a semidefinite programming relaxation [4] of the original DGP and project the obtained high-dimensional realization in  $\mathbb{R}^K$ ; then, from the obtained initial realization, we run the projected Levenberg-Marquardt algorithm [8] to tackle the part (ii) of our procedure.

We point out that the general structure of our procedure is not new. One example can be found in [1], where semidefinite programming also comes to play; another example can be found in [6]. To the best of our knowledge, however, the procedure used in our article is the first one that employs the Levenberg-Marquardt algorithm. The main motivation to use this algorithm is that it provides better convergence results when compared to other methods (such as gradient descent methods), as our computational experiments will show. As a result, despite the simplicity of our procedure, we can report successful computational experiments on relatively small-sized instances (but not really *tiny* instances, as in the experiments presented in other works). The use of our procedure appears therefore to be promising for future studies in the context of the DGP.

The rest of the paper is organized as follows. The DGP procedure main structure will be briefly introduced in Sect. 2. This short section will then contain two subparts, one (Sect. 2.1) focusing on a semidefinite programming relaxation of our target problem, and another (Sect. 2.2) describing the Projected Levenberg-Marquardt (PLM) algorithm. Computational experiments on a set of protein-like instances will be reported in Sect. 3. Finally, we will conclude the paper in Sect. 4

with an extensive discussion on the possibilities of use for the described procedure, as well as on the use of its components in other algorithmic frameworks. A particular emphasis will be given to the impact of the uncertainty on the distances on sets of DGP solutions.

## 2 Our DGP Procedure

Given a pair  $(G, K)$  representing a DGP instance, our procedure for its solution can be simply summarized in the following two steps:

**Step A.** Find a realization via a Semidefinite Programming (SDP) relaxation, following by a Principal Component Analysis (PCA) projection. The realization this way obtained is our “initial realization” (see Sect. 2.1);

**Step B.** Improve the quality of the initial realization found at the previous step by running the Projected Levenberg-Marquardt (PLM) algorithm (see Sect. 2.2).

Notice that, from now on, we will be using the acronyms SDP, PCA and PLM for referring to the methods mentioned above.

### 2.1 Semidefinite Programming Relaxation

Let  $X \in \mathbb{R}^{K \times n}$  be a matrix with the vectors  $x_i \in \mathbb{R}^K$  as its columns. We have:

$$\|x_i - x_j\|^2 = (e_i - e_j)^\top X^\top X (e_i - e_j) =: (e_i - e_j)^\top Y (e_i - e_j),$$

where  $e_i$  stands for the  $i^{\text{th}}$  canonical vector of  $\mathbb{R}^n$ . As a consequence, problem (2) is equivalent to find a positive semidefinite matrix  $Y$  of rank  $K$  such that

$$\underline{d}_{ij}^2 \leq \langle E_{ij}, Y \rangle \leq \bar{d}_{ij}^2, \quad \forall \{i, j\} \in E,$$

where  $\langle A, B \rangle := \text{trace}(A^\top B)$  is the trace inner product and  $E_{ij} := (e_i - e_j)(e_i - e_j)^\top$ . This reformulation could be cast as a linear SDP except for the (nonconvex) rank constraint. Following [1, 4], we suppress the rank constraint and consider the following SDP relaxation:

$$\begin{aligned} \min_{Y=Y^\top} \quad & -\gamma \langle I, Y \rangle \\ \text{s.t.} \quad & \underline{d}_{ij}^2 \leq \langle E_{ij}, Y \rangle \leq \bar{d}_{ij}^2, \quad \forall \{i, j\} \in E \\ & Y\mathbf{1} = 0, \quad Y \succeq 0, \end{aligned} \tag{3}$$

where  $\mathbf{1} \in \mathbb{R}^n$  is a vector of ones,  $\gamma > 0$  is a regularization parameter and  $Y \succeq 0$  means that  $Y$  must be positive semidefinite. The term  $-\langle I, Y \rangle$  in the objective function corresponds to a rank reduction heuristic [4]. The reasoning behind it is that, under  $Y\mathbf{1} = 0$ , we have that

$$\langle I, Y \rangle = \text{trace}(Y) = \text{trace}(Y) - \frac{1}{n} \mathbf{1}^\top Y \mathbf{1} = \frac{1}{2n} \sum_i \sum_j \|x_i - x_j\|^2,$$

and by maximizing this quantity we force the corresponding realization to be “more flat” and hence belonging (hopefully) to a lower dimensional space.

Let  $Y$  be a solution to problem (3). Since  $Y \succeq 0$ , we have that  $Y = X^\top X$ , where  $X \in \mathbb{R}^{r \times n}$ , with  $r = \text{rank}(Y)$ . Although  $X$  satisfies all distance constraints, it may happen that the rank  $r$  is strictly larger than the desired dimension  $K$ . This is the reason why it is necessary to project  $X$  onto  $\mathbb{R}^K$ : we perform this projection by PCA. Let  $Y = Q\Lambda Q^\top$  be the eigendecomposition of  $Y$  and assume the eigenvalues are ordered in non-increasing order  $\lambda_1 \geq \dots \lambda_n \geq 0$ . If  $\Lambda_K$  denotes the principal submatrix of  $\Lambda$  containing the  $K$  largest eigenvalues of  $Y$  and  $Q_K$  contains the  $K$  corresponding eigenvectors in its columns, then

$$X_0 = \sqrt{\Lambda_K} Q_K^\top$$

gives us the sought “projection”, which is an approximate realization in  $\mathbb{R}^K$ . We say *approximate* because after the projection,  $X_0$  may no longer satisfy some distance constraints.

In order to recover the feasibility of the violated constraints, we consider a refinement step which consists in an iterative method for solving problem (2) using the columns of  $X_0$  as starting point for the vectors  $x_i$  (the additional variables  $y_{ij}$  are initialized to the values  $(\bar{d}_{ij}^2 + \bar{d}_{ij}^2)/2$ ). For this refinement step, we consider the PLM algorithm [8], which is briefly reviewed in the next subsection.

## 2.2 Projected Levenberg-Marquardt Algorithm

Consider the following constrained system of nonlinear equations:

$$\begin{cases} F(z) = 0, \\ z \in C, \end{cases} \tag{4}$$

where  $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$  is continuously differentiable and  $C$  is a convex compact set. Let  $J(z)$  denote the Jacobian of  $F$  at  $z$  and  $P_C(u)$  the Euclidean projection of  $u$  onto  $C$ . Moreover, let us define the following function:

$$f(z) = \frac{1}{2} \|F(z)\|^2.$$

Following [8], we consider the PLM algorithm for solving eq. (4), summarized below.

### The Projected Levenberg-Marquardt (PLM)

Given  $z_0 \in C$ ,  $\sigma, \eta_1 \in (0, 1)$ ,  $M \in \mathbb{Z}_{++}$ ,  $\eta_2 > 0$ , set  $k = 0$ .

**Step 1.** Set  $\mu_k = \|F(z_k)\|^2$  and solve  $(J(z_k)^\top J(z_k) + \mu_k I) d_k^U = -J(z_k)^\top F(z_k)$

**Step 2.** Set  $d_k^C = P_C(z_k + d_k^U) - z_k$ . If  $d_k^C$  satisfies

$$F(z_k)^\top J(z_k) d_k^C \leq -\eta_1 \|d_k^C\|^2 \tag{5}$$

$$\|d_k^C\| \leq \eta_2 \|J(z_k)^\top F(z_k)\| \tag{6}$$

then  $d_k = d_k^C$ , else  $d_k = P_C(z_k - J(z_k)^\top F(z_k)) - z_k$ .

**Step 3.** Set  $\alpha = 1$ . While  $f(z_k + \alpha d_k) > \max_{0 \leq j \leq \min\{M, k\}} f(z_{k-j}) + \sigma \alpha d_k^\top J(z_k)^\top F(z_k)$ , update  $\alpha = \alpha/2$ .

**Step 4.** Set  $\alpha_k = \alpha$  and update  $z_{k+1} = z_k + \alpha_k d_k$ . Go to Step 1.

Given  $z_k \in C$ , the unconstrained LM direction  $d_k^U$  is computed at Step 1. In Step 2, the feasible direction  $d_k^C$ , based on the projection of  $z_k + d_k^U$  onto  $C$ , is computed and it is chosen as search direction if it satisfies the descent conditions in Eq. (5) and (6). Otherwise, the projected gradient direction is taken. A step-size  $\alpha > 0$  verifying a non-monotone Armijo-like condition [9] is determined in Step 3 by a backtracking process.

For more details about this algorithm, the reader is referred to [8], where a detailed convergence analysis of the algorithm can also be found. It was proved, in fact, that every limit point of the sequence  $\{z_k\}$  is stationary for the problem of minimizing  $f(z)$  subject to  $z \in C$ . Furthermore, under an error bound condition, a local superlinear convergence was established.

We point out that problem (2) corresponds to problem (4) with  $z = (X, y) \in \mathbb{R}^{K \times n} \times \mathbb{R}^{|E|}$ ,  $F : \mathbb{R}^{K \times n} \times \mathbb{R}^{|E|} \rightarrow \mathbb{R}^{|E|}$  with

$$[F(X, y)]_{ij} = \|x_i - x_j\|^2 - y_{ij}, \quad \forall \{i, j\} \in E,$$

and

$$C = \{(X, y) \in \mathbb{R}^{K \times n} \times \mathbb{R}^{|E|} \mid \underline{d}_{ij}^2 \leq y_{ij} \leq \bar{d}_{ij}^2, \forall \{i, j\} \in E\}.$$

In this case, the least-squares function  $f$  takes the form

$$f(X, y) = \frac{1}{2} \|F(X, y)\|^2 = \sum_{\{i, j\} \in E} (\|x_i - x_j\|^2 - y_{ij})^2, \quad (7)$$

which corresponds to the Smooth STRESS function [18], with  $\underline{d}_{ij}^2 \leq y_{ij} \leq \bar{d}_{ij}^2$ .

### 3 Computational Experiments

We propose in this section some initial computational experiments with the DGP procedure sketched in Sect. 2. All experiments were carried out on Matlab R2018b running MacOS X 10.13.6 (personal laptop).

We consider two sets of instances, both related to protein conformations. However, in the first set that we consider, the instances will only *resemble* to typical protein instances, because we will not include any additional distance information that is likely to help DGP solvers to find solutions. This “extra” and helpful distance information would include, for example, the length of chemical bonds, as well as the angles formed by triplets of consecutively bonded atoms. Thus, we decide to take, in our instance generation procedure, no advantage from the typical chemical structure of protein conformations. We make this choice for

**Table 1.** Some experiments showing the effectiveness of our DGP procedure on the two sets of instances. In the upper row block, we consider the instances generated in this work where no extra information about the nature of the distances is exploited; in the lower row block, we present the experiments on the protein instances previously used in [7].

pdb-id	V	E	$\gamma$	$f_0$	PLM			SPG		
					$k$	$f$	RMSD	$k$	$f$	RMSD
2JMY	45	432	1	9.16E+01	93	4.26E-09	0.08	322	4.53E-02	0.08
2LR9	57	505	1	3.72E+03	555	4.90E-09	1.16	801	1.73E-01	1.36
1HJO	123	1210	1	8.40E+03	73	4.20E+01	3.86	1602	4.22E+01	3.83
1HJO	123	1210	10	8.37E+03	778	4.94E-09	2.13	1498	5.23E-01	2.57
2KSL	153	1398	1	4.56E+04	250	2.63E+02	7.59	1010	6.58E+02	10.23
2KXA [7]	177	973	1	9.43E+02	172	4.68E-09	0.45	686	2.79E-02	0.62
1DSK [7]	222	1210	10	6.29E+03	268	4.99E-09	2.51	578	1.79E-02	2.44
2ERL [7]	323	1789	1	1.99E+03	245	4.80E-09	0.41	704	2.66E-02	0.41
2JWU [7]	447	2413	1	6.02E+03	219	4.69E-09	1.03	824	4.26E-02	0.99

the generation of our first set of instances with the aim of testing the effectiveness of the procedure for larger classes of DGP instances, which may be related to different applications.

Instead, the second set of instances that we consider in our experiments will include this additional information. For lack of space, we focus our attention on the main steps for generating our new instances of the first set, while the reader is referred to [7] for details on how the 4 instances of the second set were generated.

In order to generate the first set of instances, we consider models of protein conformations obtained from the Protein Data Bank (PDB) [2]. From one selected PDB model (when more than one model is available in the same PDB file, we simply pick the first one), we extract the backbone atoms N,  $C_\alpha$  and C, and we generate the corresponding instance by measuring all possible distances between pairs of such backbone atoms, and by keeping only the distances shorter than  $6\text{\AA}$ . Noise is thereafter added to the distances by creating an interval  $[\underline{d}, \bar{d}]$  of range  $0.2\text{\AA}$ , where the computed distance is randomly placed. The procedure outputs a simple weighed graph  $G$  that represents an instance of the DGP. We point out that our procedure introduces the same level of noise in all the distances, without distinguishing between distances between bonded atoms or other.

To assess the performance of PLM as a refinement tool, we compare it with the Spectral Projected Gradient (SPG) algorithm [3] for minimizing the function (7) over  $C$ . Notice that SPG was already successfully used in previous works as a local solver for DGP [17].

In all experiments, the SDP relaxation in Eq. (3) was solved using SDPT3 solver [19] with standard parameters and tolerances. In PLM, we consider the parameters  $\eta_1 = 10^{-4}$ ,  $\eta_2 = 10^4$ ,  $\sigma = 10^{-3}$ ,  $M = 10$  and stop the iterations when



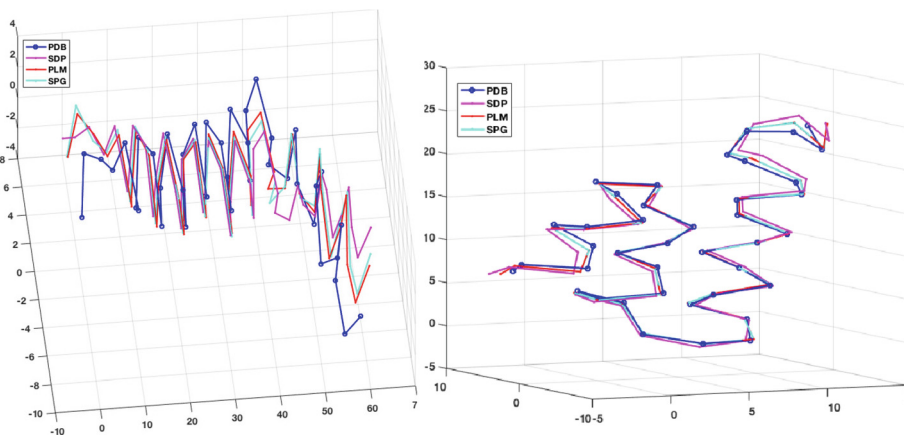
either  $\|F(X_k, y_k)\| \leq \varepsilon = 10^{-4}$  or  $\|z_k - P_C(z_k - \nabla f(z_k))\| \leq \varepsilon$ . For SPG, we used the same parameters as in [17], and stopped the iterations when  $\|d_k\| \leq \varepsilon$ . The maximum number of iterations was set to 2,000 for both SPG and PLM.

Table 1 summarizes the performed computational experiments. For every instance, we report the original PDB identifier of the protein in the PDB, together with the total number of vertices and the total number of edges in the generated graph  $G$ . The parameter  $\gamma$  is the one involved in SDP, while  $f_0$  is set to  $f(X_0, y_0)$ , which corresponds to the value of eq. (7) evaluated at the solution  $X_0$  in **Step A**, where  $y_0 = (\underline{d}^2 + \bar{d}^2)/2$ . For both PLM and SPG, we report the number of iterations  $k$ , the final objective function value for  $f(X, y)$  (denoted  $f$  in the table), and the RMSD with respect to the first model of the PDB file. Notice that only the  $C_\alpha$  atoms in the solution found for our second set of instances were taken into consideration when computing the RMSD (for example, for the 2ERL instance, only 40 atoms out of 323 were selected).

The experiments show that, although after the execution of our **Step A** the value of Eq. (7) for our initial realization is relatively large, such a starting point is nevertheless close enough to one of the instance solutions. In fact, the **Step B** in our procedure, when performed by running the PLM algorithm, is able to decrease the value of the Smooth STRESS function to a magnitude of  $10^{-9}$  for the instances 2JMY, 2LR9, 2KXA and 2ERL belonging to our first instance set. This indicates that all involved distances are satisfied (i.e.,  $\|x_i - x_j\| \in [\underline{d}_{ij}, \bar{d}_{ij}]$ ), or close to be satisfied (i.e.,  $\|x_i - x_j\| \notin [\underline{d}_{ij}, \bar{d}_{ij}]$ , but  $\|x_i - x_j\|$  is close to one of the two bounds). Notice that we can state a similar remark for SPG, but with a final value for the STRESS function that is about six orders of magnitude larger. Even if the corresponding RMSD values are similar to those obtained by PLM, we can remark therefore that the PLM provides solutions in general capable to better satisfy the available distances.

Concerning the number of iterations of PLM and SPG, we can observe that, although the former requires fewer iterations, it is important to mention that its iterations are more expensive because requiring the solution of a positive definite linear system.

Two times the instance 1HJ0 is reported in the upper row block of Table 1. In fact, the former of the two experiments shows that the initial realization from **Step A** was not close enough to one of the instance solutions: both PLM and SPG have most certainly converged towards a local minimizer or a stationary point of (7). In the latter experiment concerning 1HJ0, however, where the value of the  $\gamma$  parameter was changed from 1 to 10, we can observe a performance for our procedure which is close to the other experiments, where the initial realization is actually a good starting point for the refinement step (with PLM still beating SPG on the STRESS function value). This example is particularly interesting because, even though the use of a different value for  $\gamma$  leads to a STRESS function value approaching zero, the final RMSD value in the found solutions does not change much. This indicates that the instance we have generated by using the first model in the PDB file does not have only that model in its solution set. The left side of Fig. 1 shows the model we have obtained with  $\gamma = 10$ .



**Fig. 1.** A comparison among some obtained solutions and the original PDB model used to generate the instances: 1HJ0 ( $\gamma = 10$  version, on the left-hand side), and 2ERL (only  $C_\alpha$  atoms, on the right-hand side). Axis units in Angstroms.

The last line of the upper row block in Table 1 shows that, for one of the instances of our first set, we could not find any satisfactory solutions. Trying to use alternative values for the  $\gamma$  parameter did improve the results in this case; the use of small values (e.g. 0.01, 0.1) or even larger than 10, did not allow us to get close enough to one of the solutions for having either SPG or PLM converge to a global minimizer. This is certainly not the only case where our procedure can fail, because of its simplicity.

Finally, the lower row block of experiments in Table 1 shows the performances of our procedure on 4 of the instances already used in [7]. As remarked above, these instances exploit some additional distance information that can help the solvers identifying the solutions, for example by fixing some of the distances to some given precise values. Our procedure seems to provide similar performances on this second set of instances, and the comparison between PLM and SPG remains the same as well. The right side of Fig. 1 shows the solution found for the instance 2ERL: since these instances contain more atoms from the proteins (not only its backbone atoms), for clarity we only consider in the figure its  $C_\alpha$  trace, which is the same considered in the computation of the RMSD.

## 4 Discussion and Conclusions

We have presented and tested a simple procedure for the solution of DGPs where the value of the distances is uncertain and generally represented by a real-valued interval. As pointed out in the Introduction, the general structure of our simple procedure is not new, as it was already used in previous works, but, to the best of our knowledge, this is the first time that the projected Levenberg-Marquardt is employed in this context.

Even if it cannot be considered as a general solver for the DGP, our computational experiments have shown the effectiveness of our procedure on a set of artificially generated instances related to a typical biological application. The experiments show in fact that, when the first step of the procedure is able to identify an initial realization that is close enough to a valid realization (a solution for problem (2)), then its second step is able to localize that solution in the search domain.

These results open the doors for other possible uses of this procedure (or of one of its components) in more general solution methods for the DGP. For example, MDJEEP<sup>1</sup> is a solver for DGPs for which the discretization of the search space can be performed, by transforming the problem in a combinatorial problem [16]. When the value on the distances is uncertain, however, some nodes of the search domain cannot be associated to singletons, but rather to relatively small portions of the original continuous search domain. This is the reason why, in MDJEEP, the combinatorics is coupled with a refinement step consisting in locally exploring all those small domain portions in the attempt to improve the overall solution quality [14]. The impact of the current work on the future developments of MDJEEP can be two-fold. Firstly, the first step of our procedure may be used to guess the most promising parts of the discretized search domain to enhance its performance in terms of time. The idea is only to give higher priority to the identified parts of the search domain, and to remove, a priori, none of them, so that the entire search domain may, potentially, still be explored. Secondly, since the current version of MDJEEP, the version 0.3.2 at the moment we are writing this article, uses SPG for performing its refinement step, another possible improvement may be to replace SPG with PLM.

Another interesting application falls in the context of motion adaptation [11]. Here, a skeletal structure (representing for example a human character) performs a given motion over time, and the problem of embedding the same motion (or a motion as close as possible to the original one) on another skeletal structure is considered. One of the main difficulties in solving such a problem is related to the fact that distorted self-contacts, which may be either artificially created in the new motion, or rather omitted from the original one, are likely to make the viewer perceive the motion as different when compared to the original. Self-contact is synonym of high proximity, and hence of near distances. The *dynamical* DGP (dynDGP) was introduced in [13] to tackle this class of problems, and more recently it was applied to motion adaptation in [10]. In this application, every frame of the motion can be considered as a separated (and static) DGP, where every new frame belongs to a small neighborhood, in its search domain, of the previous frame. SPG was exploited in previous works on distance-based motion adaptation: the animations created in [10] were generated by SPG for example. Again, we propose the use of PLM as a replacement for SPG, which is likely to provide interesting results in this application context as well.

We remark that, in the two cases where we propose to use PLM as a tool for refinement, the DGP instances to be solved (in both cases, these are actually sub-

---

<sup>1</sup> <https://github.com/mucherino/mdjeep>.

instances of the original problems) are rather simple if compared to the ones we used in the computational experiments in this work. When the refinement step is performed in MDJEEP, in fact, only a subset of vertices of the original instance is considered, and the local solver can benefit of a starting point where (in most of the cases) only a few distances are not satisfied (the best-case scenario being the one where only one distance is violated). In the case of motion adaptation, since the starting point is always the solution obtained at the previous frame of the motion, it is expected the new local solution to be very near the available starting point. Future works will be devoted to these very promising research directions.

**Acknowledgments.** The authors are very grateful to CAPES/Brazil for the CAPES-PRINT project, process number 88887.578009/2020-00, allowing AM to visit DG at UFSC, Florianópolis (SC, Brazil) for a 2-week time in December 2021. Most of the presented work was performed during such a visit. AM is also thankful to the ANR for the support on the international France-Taiwan project MULTIBIOSTRUCT (ANR-19-CE45-0019). DG thanks the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Grant 305213/2021-0.

## References

1. Alipanahi, B., Krislock, N., Ghodsi, A., Wolkowicz, H., Donaldson, L., Li, M.: Determining protein structures from NOESY distance constraints by semidefinite programming. *J. Comput. Biol.* **20**(4), 296–310 (2013)
2. Berman, H., et al.: Protein data bank. *Nucleic Acids Res.* **28**, 235–242 (2000)
3. Birgin, E.G., Martínez, J.M., Raydan, M.: Spectral projected gradient methods: review and perspectives. *J. Stat. Softw.* **60**(i03), 21 (2014)
4. Biswas, P., Liang, T., Wang, T., Ye, Y.: Semidefinite programming based algorithms for sensor network localization. *ACM Trans. Sens. Netw.* **2**(2), 188–220 (2006)
5. Crippen, G.M., Havel, T.F.: *Distance Geometry and Molecular Conformation*. John Wiley & Sons (1988)
6. D’Ambrosio, C., Vu, K., Lavor, C., Liberti, L., Maculan, N.: New error measures and methods for realizing protein graphs from distance data. *Discrete Comput. Geom.* **57**(2), 371–418 (2017). <https://doi.org/10.1007/s00454-016-9846-7>
7. Gonçalves, D.S., Mucherino, A., Lavor, C., Liberti, L.: Recent advances on the interval distance geometry problem. *J. Glob. Optim.* **69**(3), 525–545 (2017)
8. Gonçalves, D.S., Gonçalves, M.L.N., Oliveira, F.R.: An inexact projected LM type algorithm for solving convex constrained nonlinear equations. *J. Comput. Appl. Math.* **391**(1), 113421 (2021)
9. Grippo, L., Lampariello, F., Lucidi, S.: A truncated newton method with non-monotone line search for unconstrained optimization. *J. Optim. Theory Appl.* **60**, 401–419 (1989)
10. Hengeveld, S.B., Mucherino, A.: On the representation of human motions and distance-based retargeting. In: *IEEE Conference Proceedings, Federated Conference on Computer Science and Information Systems (FedCSIS21), Workshop on Computational Optimization (WCO21)*, pp. 181–189. Sofia, Bulgaria (2021)

11. Ho, E.S.L., Komura, T., Tai, C-L.: Spatial relationship preserving character motion adaptation. In: Proceedings of the 37<sup>th</sup> International Conference and Exhibition on Computer Graphics and Interactive Techniques. ACM Transactions on Graphics, vol. 29, no. 4, p. 8 (2010)
12. Liberti, L., Lavor, C., Maculan, N., Mucherino, A.: Euclidean distance geometry and applications. *SIAM Rev.* **56**(1), 3–69 (2014)
13. Mucherino, A., Gonçalves, D.S.: An approach to dynamical distance geometry. In: Nielsen, F., Barbaresco, F. (eds.) GSI 2017. LNCS, vol. 10589, pp. 821–829. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-68445-1\\_94](https://doi.org/10.1007/978-3-319-68445-1_94)
14. Mucherino, A., Gonçalves, D.S., Liberti, L., Lin, J-H., Lavor, C., Maculan, N.: MD-jeep: a new release for discretizable distance geometry problems with interval data. In: IEEE Conference Proceedings, Federated Conference on Computer Science and Information Systems (FedCSIS20), Workshop on Computational Optimization (WCO20), pp. 289–294. Sofia, Bulgaria (2020)
15. Mucherino, A., Lavor, C., Liberti, L., Maculan, N. (eds.), *Distance Geometry: Theory, Methods and Applications*, p. 410. Springer (2013)
16. Mucherino, A., Liberti, L., Lavor, C.: An implementation of a branch and prune algorithm for distance geometry problems. In: Fukuda, K., Hoeven, J., Joswig, M., Takayama, N. (eds.) ICMS 2010. LNCS, vol. 6327, pp. 186–197. Springer, Heidelberg (2010). [https://doi.org/10.1007/978-3-642-15582-6\\_34](https://doi.org/10.1007/978-3-642-15582-6_34)
17. Mucherino, A., Lin, J-H.: An efficient exhaustive search for the discretizable distance geometry problem with interval Data. In: IEEE Conference Proceedings, Federated Conference on Computer Science and Information Systems (FedCSIS19), Workshop on Computational Optimization (WCO19), Leipzig, Germany, pp. 135–141 (2019)
18. Takane, Y., Young, F.W., de Leeuw, J.: Nonmetric individual multidimensional scaling: an alternating least squares method with optimal scaling features. *Psychometrika* **42**(1), 7–67 (1977)
19. Toh, K.C., Todd, M.J., Tütüncü, R.H.: SDPT3 - a matlab software package for semidefinite programming. *Optim. Meth. Softw.* **11**, 545–581 (1999)



# A Semi-supervised Graph Deep Neural Network for Automatic Protein Function Annotation

Akrem Sellami<sup>1</sup>✉, Bishnu Sarker<sup>2</sup>, Salvatore Tabbone<sup>1</sup>,  
Marie-Dominique Devignes<sup>1</sup>, and Sabeur Aridhi<sup>1</sup>

<sup>1</sup> Université de Lorraine, CNRS, Inria, LORIA, Nancy, France  
{akrem.sellami,marie-dominique.devignes,sabeur.aridhi}@loria.fr,  
salvatore.tabbone@univ-lorraine.fr

<sup>2</sup> Meharry Medical College, Nashville, TN, USA  
bsarker@mmc.edu

**Abstract.** The protein function annotation based on functional properties like the Enzyme Commission (EC) numbers is a very challenging task that aims to understand life at the molecular level. Especially, the size of features for each protein is very huge and the number of labeled samples is limited, which can significantly affect the annotation accuracy. To address these issues, we propose a novel semi-supervised graph deep learning model that aims to learn better latent representations for each protein/node by taking into account the neighborhood information in order to improve the annotation. Firstly, we extract a set of features from raw protein data. Each protein is associated with a 1-D feature vector that represents its InterPro domain composition. As  $D$ , the number of possible interPro domains, is very high ( $>11,000$ ), we design a deep autoencoder model (DAE) that seeks to find an efficient representation of the domain composition of proteins in a lower dimensional latent space. Then, we construct a protein graph where each node is a protein associated with its latent representation vector and each edge is weighted by the Euclidean distance between the two nodes it connects. Finally, we train a semi-supervised graph neural network (SGNN) for the automatic protein function annotation using the constructed protein graph. Experiments are conducted on four reference proteomes in UniProtKB/SwissProt, including Human, Arabidopsis Thaliana, Mouse, and Rat. Experimental results show that the proposed model is competitive for protein function annotation compared to existing methods.

**Keywords:** Deep autoencoder · Feature extraction · Protein function annotation · Graph representation learning

## 1 Introduction

Nowadays, protein sequence analysis plays a very important role in understanding disease processes and drug discovery. Thanks to the remarkable scientific

progress in this field, it is possible to collect large numbers of protein sequences that are now available in public databases like UniProtKB [3]. However, this huge quantity of protein data causes several challenges for computer scientists as well as biologists in protein function annotation [11, 17, 19]. Recently, several machine learning methods have been proposed to overcome these challenges [2, 5, 6, 14]. In [12], authors proposed a deep learning approach called Deepre, which seeks to predict Enzyme Commission (EC) numbers based on proteins sequence using feature selection and classification techniques. Furthermore, Sarker et al. [15, 16] proposed a graph-based model called GrAPFI for automatically annotating proteins with EC numbers. GrAPFI uses the label propagation technique for protein function annotation based on domain similarity graphs using the Jaccard index. In [22], a hybrid model called COFACTOR has been developed for protein sequence annotation. It fuses sequence homologs and protein structure with Protein-Protein Interaction (PPI) networks to predict EC numbers, ligand-binding, and Gene Ontology (GO) terms. In [10] Ko et al. proposed a deep learning framework called FUTUSA (Function Teller Using Sequence Alone) to predict protein functions *in silico*. It performs sequence segmentation using a convolutional neural network (CNN) to extract the regional sequence patterns.

Based on previous works, we can conclude that deep learning models have shown their high performance in improving protein function annotation. However, there are several issues, especially the large number of protein features, e.g. InterPro domain composition, that can lead to the problem of the *curse of dimensionality* and overfitting, which can significantly destroy the performance in terms of function annotation precision and accuracy. Moreover, existing models do not consider the topological local information between vertices (proteins), which can be useful to improve the annotation accuracy.

Modeling graph data is a challenging task, especially when dealing with large amounts of data. The success of deep learning has generated interest in extending this technique to non-Euclidean structures such as graphs and manifolds, namely Geometric Deep Learning (GDL). This emerging field combines the representational power of graphs with deep learning. This new framework allows to learn how to propagate the information along the entities, i.e. nodes, conforming the graph. More and more reports suggest that DL can extract useful features from non-Euclidean data [18, 23]. Furthermore, a core assumption of existing machine learning algorithms is that instances are independent of each other. However, these data do not always have a regular structure, like protein data, which can be represented as a complex big graph [1]. Graphs are composed of a variable number of unordered nodes, which in turn, have different numbers of neighboring nodes, resulting that some important operations (e.g. convolution) which are easily computed in image processing, are difficult to extend to the graph domain. In the literature, there are a limited number of existing reviews on the topic of graph neural networks [21]. Most of them need a large number of training samples and do not consider the problem of overfitting. Moreover, there is a big challenge related to a large amount of collected data. Recently, graph

deep representation learning [7, 8, 13] models have been developed to capture hidden patterns of non-Euclidean data. For example, a graph convolution can be generalized from a 2D convolution. Similar to 2D convolution, one may perform graph convolutions by taking the weighted average of a node’s neighborhood information.

The objective of this work is to design new graph deep learning techniques that will allow incorporating the non-euclidean space of data. Our models will be applied to protein function annotation which is a very challenging task that aims to understand life at the molecular level. This task is important in several scenarios including human disease and drug discovery. In the context of drug discovery, it is important to have various annotations and information about protein sequences since the majority of available drugs have protein molecules as their targets. Moreover, the wealth of protein sequences being produced has generated the need for rapid annotation of protein sequences. We will validate our model on several datasets and make available our software to the scientific community. In the literature, graph representation of protein data and machine learning methods have been used simultaneously. Depending on the number of protein entries used to construct the graph and on the number of features of each protein (protein sequence, functional domains, keywords, ...), the graph representing protein entries could be very huge and complex. Thus, extracting useful features for each node/protein while providing good annotations using only a few labeled proteins (samples) is a challenging task. Our research hypothesis is that improving graph network methods will make the protein function annotation task easier. Therefore, we propose a novel graph deep learning model for automatic protein function annotation that allows learning a better latent representation from InterPro domain composition and considers the local features in protein graphs, in order to improve the protein annotation precision.

The remainder of this paper is organized as follows. In Sect. 2, we present the proposed method for automatic protein function annotation along with the developed algorithms for the protein graph construction and the semi-supervised graph neural networks (SGNN). Afterward, in Sect. 3, experimental results are described exhibiting performance improvements of the proposed approach. In Sect. 4, we conclude and propose some perspectives and future works.

## 2 Proposed Approach

The proposed approach (see Fig. 1) aims to exploit proteins datasets by extracting all different domains enabling to automatically annotate unlabeled proteins. More precisely, the latent representation is obtained from the bottleneck hidden layer of a deep autoencoder trained from extracted domains. The aim is to extract relevant features and get a better representation for each protein by discarding the nuisance features. Finally, the latent representation is used as input of a graph convolutional neural network, which is trained for automatic protein function annotation. The annotation phase aims to annotate the unlabeled proteins using the learned latent representations.



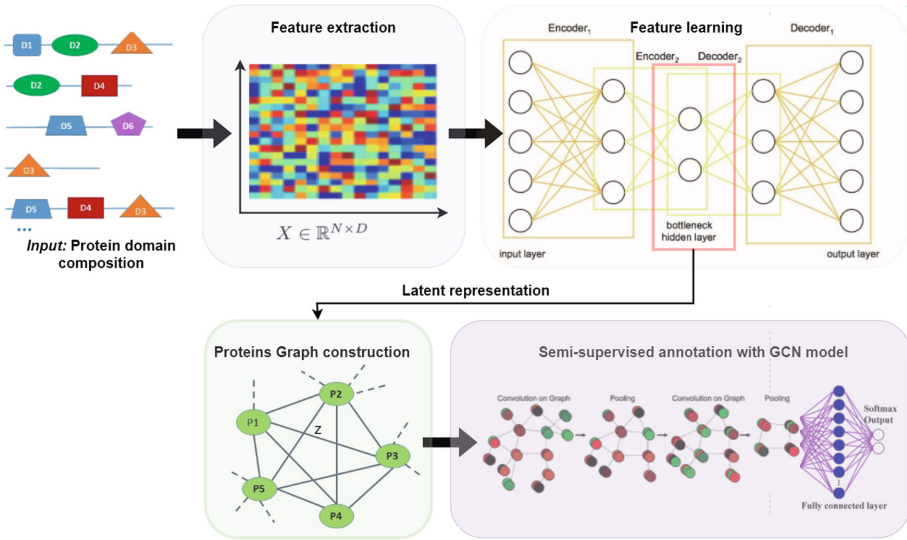


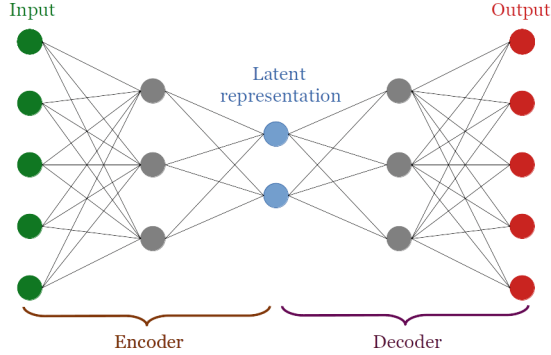
Fig. 1. Architecture of the proposed approach

## 2.1 Protein Feature Extraction

The first phase of the proposed approach is data analysis. The main goal is to process the proteins data to obtain the features vectors from domains  $\{X\}$ . We will extract all different domains for each protein and encode them in order to get a feature vector. At the end of this process, each feature vector is represented as a 1-D vector, where  $D$  is the number of different domains of all proteins. The main assumption is that proteins sharing the same function also share the same domains and identifying proteins represented by similar domain composition can be useful to enhance the annotation task. Therefore, we propose to use these feature vectors in a deep learning model in order to find a shared representation between all proteins.

## 2.2 Representation Learning with Deep Autoencoders (DAE)

We designed a deep autoencoder (DAE) that aims at extracting relevant features from the domain composition of each protein, from which this domain composition may be reconstructed. It relies on our case on the assumption that domains of proteins are relevant for the annotation task. Our goal is to extract a relevant latent representation from which one may construct a graph in order to train a graph deep representation learning model to obtain an accurate annotation. The proposed DAE model includes an encoder and a decoder noted  $Enc$ , and  $Dec$ , respectively.  $Enc$  is a multi-layer deep learning network that contains three hidden layers. It aims to encode the input proteins domain vectors  $X \in \mathbb{R}^{N \times D}$  into a new latent representation space  $z \in \mathbb{R}^{N \times d}$ , where the number of extracted



**Fig. 2.** Architecture of the deep autoencoder model (DAE).

features  $d$  is much lower than  $D$  ( $d \ll D$ ). The encoder network uses the encoding function  $g_\theta(x)$  as follows:

$$g_\theta(x) = \Phi_g(W_x + b) \quad (1)$$

where  $\Phi_g$  is an activation function such as *ReLU()*, *Sigmoid()*, *Linear()*,  $W$  is the weights of neurons, and  $b$  is the bias of the model. Therefore, we note  $z$  the corresponding representation output by the encoder,  $z = Enc(x) = g_\theta(x)$ . The second component of the DAE model is the decoder  $Dec$  that uses the obtained latent representation  $z$  to reconstruct all domains,  $\hat{x} = Dec(z)$ . The reconstructed input may be obtained using the decoding function as:

$$\hat{x} = f_{\theta'}(z) = \Phi_f(W'_z + b') \quad (2)$$

The optimization function of this model is based on mean squared error (MSE) criterion. It aims to reduce the reconstruction error between the input  $X$  and the output  $\hat{X}$ . It can be expressed as follows:

$$\mathcal{L}_{DAE}(\theta, \theta') = \frac{1}{2n} \sum_i^n \|x^{(i)} - f_{\theta'}(g_\theta(x^{(i)}))\|^2 \quad (3)$$

where  $n$  is the number of proteins,  $\theta$  and  $\theta'$  are the learned parameters of the DAE model. Figure 2 reports the architecture of the DAE model.

### 2.3 Latent Representation-Based Protein Graph Construction

We use the extracted latent representations  $z = [x_1, \dots, x_d] \in \mathbb{R}^{N \times d}$  to construct our protein graph noted  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ .  $\mathcal{V} = \{v_1, \dots, v_N\}$  is a set of vertices corresponding to the proteins,  $\mathcal{E}$  is the set of edges, and  $\mathcal{W} \in \mathbb{R}^{N \times N}$  is the weighted matrix of  $\mathcal{G}$ , where  $w_{i,j}$  is a weight attributed to the edge  $e_{i,j} = (v_i, v_j) \in \mathcal{E}$ . Each node  $v_i$  in the protein graph  $\mathcal{G}$  is associated with a  $1-d$  feature vector ( $d$

is the number of extracted features with DAE) that represents the latent representation  $z_i \in \mathbb{R}^d$ . Furthermore, an edge  $e_{i,j}$  between two nodes (proteins)  $v_i$  and  $v_j$  can be defined based on a similarity criterion, which is computed by the euclidean distance as follows:

$$\mathcal{W}(v_i, v_j) = \text{dist}(z_i, z_j) = \|z_i - z_j\| = \sqrt{\sum_{t=1}^d (z_{it} - z_{jt})^2} \quad (4)$$

where  $\mathcal{W}$  is the weighted matrix, and  $z_i, z_j \in \mathbb{R}^d$  are the feature vectors of  $v_i$  and  $v_j$ , respectively. Formally, each vertex  $v_i$  is connected to  $v_j$  if  $z_i$  belongs to the neighborhood of  $z_j$  according to a well-defined neighborhood threshold  $\alpha$ , controlling the size of the neighborhood for  $\mathcal{N}(v_i) \in \{1, \dots, N\}$ . The threshold  $\alpha$  is between 0 and 1 (0 and 1 are the minimum and maximum values of  $\mathcal{W}$ ). Therefore, the adjacency matrix  $\mathbf{A}$  based on  $\mathcal{W}$  and  $\alpha$  is computed using the following formula:

$$\mathbf{A}(v_i, v_j) = \begin{cases} 1, & \text{if } w_{i,j} \leq \alpha \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

Algorithm 1 describes the different steps of the protein graph construction.

---

**Algorithm 1.** Protein graph construction based on latent representation

---

**Require:**  $\mathbf{Z} \in \mathbb{R}^{N \times d}$  //  $Z$ : Latent representation

$\alpha$  (threshold): integer

**Ensure:**  $G = (V, E, W)$  Graph of  $Z$  and its adjacency matrix  $A$ .  $A$  depends on the threshold  $\alpha$ ,  $V \leftarrow []$ ,  $E \leftarrow []$ ,  $W \leftarrow []$ ,  $A \leftarrow []$

// Find all vertices  $V$  (proteins)

**for**  $i = 1 : N$  **do** //  $N$ : Number of protein

**for**  $j = 1 : d$  **do** //  $d$ : Dimensionality of latent representation

$V[i][j] \leftarrow Z[i][j]$

**end for**

**end for**

// Find Edges  $E$  for each protein  $v_i$  and compute the weights for each edge  $E = (v_i, v_j)$

$W(v_i, v_j) = \text{dist}(z_i, z_j)$

**if**  $\text{dist}(z_i, z_j) \leq \alpha$  **then**

$E \leftarrow E \cup \{(v_i, v_j)\}$

$A(v_i, v_j) \leftarrow 1$

**else**

$A(v_i, v_j) \leftarrow 0$

**end if**

**return**  $V, E, W, A$

---

## 2.4 Protein Function Annotation with Semi-Supervised Graph Neural Networks (SGNN)

In this section, we propose a novel graph deep representation learning method which aims to learn better latent representations for each node by taking into account the neighborhood information. In usual protein similarity graphs, each node will be associated with a high-dimensional feature vector, which contains a large number of features extracted from the raw protein data, e.g., domains. The challenge here is the exploitation of the latent representation of protein data in order to efficiently annotate unreviewed/unlabeled proteins based on the information on reviewed/labeled ones. We design a dedicated graph deep network architecture in order to obtain efficient representations of such protein data by incorporating neighborhood information and considering the irregular structure of data. Formally, let consider the protein graph  $G$  obtained by the proposed algorithm of Sect. 2.3. For semi-supervised learning, let  $TD_l = \{x_i, y_i\}_{i=1}^L$  be a set of labeled training dataset of size  $L$ , where  $x_i$  indicates a feature vector of the  $i^{th}$  labeled protein, and  $y_i$  is its corresponding label. Moreover, let  $TD_{nl} = \{x_i\}_{i=L+1}^{L+NL}$  be a set of unlabeled training samples of size  $NL$  ( $L + NL = N$ ).

The main goal of semi-supervised learning is to predict the labels of unlabeled training samples  $TD_{nl}$ , using a non-linear function  $f(X, W)$  such as ReLu [9]. Furthermore, convolution on graphs can be computed by multiplying each graph signal  $\bar{z}$  by a filter  $g_\theta$  parametrized by the Fourier coefficient  $\theta \in \mathbb{R}^N$ . Usually, the graph *Fourier transform* for a signal  $z$  is defined as:

$$\mathcal{F}(z) = U^T z = \hat{z} \in R^n \quad (6)$$

where  $\mathcal{F}^{-1}(z) = U\hat{z}$ ,  $\mathbf{U}$  is the matrix of eigenvectors of the normalized graph Laplacian  $\mathbf{L}_n = \mathbf{I}_N - \text{Diag}^{-\frac{1}{2}} \mathbf{W} \text{Diag}^{-\frac{1}{2}} = \mathbf{U} \Sigma \mathbf{U}^T$ ,  $\text{Diag}$  is the diagonal degree matrix of  $\mathcal{G}$ ,  $\mathbf{I}_N$  is the identity matrix, and  $\Sigma$  is the diagonal matrix of eigenvalues. The graph convolution of the latent representation  $z$  with a filter  $g \in R^n$  is calculated using:

$$z *_{\mathcal{G}} g = \mathcal{F}^{-1}(\mathcal{F}(z) \odot \mathcal{F}(g)) = U(U^T(z) \odot U^T(g)) \quad (7)$$

where  $\odot$  denotes the element wise product. According to [4], we can efficiently compute an approximated convolution of  $G$  as follows:

$$g_\theta * \bar{z} = \theta B \bar{z} \quad (8)$$

where  $B = \mathbf{I}_N + \text{Diag}^{-\frac{1}{2}} \mathbf{W} \text{Diag}^{-\frac{1}{2}} + (\text{Diag}^{-\frac{1}{2}} \mathbf{W} \text{Diag}^{-\frac{1}{2}})^2$ .

For the semi-supervised learning, the optimal neural network weights  $\mathbf{W}^{(0)}, \mathbf{W}^{(1)}, \dots, \mathbf{W}^{(K)}$  can be trained using the labeled set of training samples  $TD_l = (x_i, y_i)_{i=1}^L$ , by minimizing the standard cross-entropy loss function:

$$\mathcal{L}_{oss} = - \sum_{i=1}^L y_i \ln \mathbf{M}_i \quad (9)$$

where  $\mathbf{M}_i$  is the label output of node  $i$  in the final layer.

The proposed SGNN aims to predict the labels of unlabeled proteins  $\mathbf{x}_i \in TD_{nl}$  which will go through various propagation layers. Formally, given an input latent feature matrix  $\mathbf{Z}$  and a weighted adjacency matrix  $\mathbf{A}$ , GNN applies a layer-wise propagation rule using the Rectified Linear Unit (*ReLU*) as a non-linear activation function and *softmax*() as a classifier:

$$\begin{aligned} \mathbf{Z}^{(1)} &= \text{ReLU}(B\mathbf{Z}^{(0)}\mathbf{W}^{(0)}) \\ &\vdots \\ \mathbf{Z}^{(K-1)} &= \text{ReLU}(B\mathbf{Z}^{(K-2)}\mathbf{W}^{(K-2)}) \\ \mathbf{Z}^{(K)} &= \text{softmax}(B\mathbf{Z}^{(K-1)}\mathbf{W}^{(K-1)}) \end{aligned} \tag{10}$$

where  $\mathbf{Z}^{(0)} = \mathbf{Z}$ ,  $\{\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}, \dots, \mathbf{Z}^{(K-1)}\}$  are the feature map outputs of the different layers and  $\mathbf{Z}^{(K)} = M$  is a vector of output labels, i.e.,  $M_i$  is the label of vertex  $v_i$ .

### 3 Experimental Results

#### 3.1 Data Description

We conduct our experiments on 61,832 proteins collected from the UniProt-KB/SwissProt database [3] using four reference proteomes, including Human, Rat, Arabidopsis thaliana, and Mouse. For each proteome, we downloaded from UniProt-KB website the EC annotation and InterPro domain composition of each protein entry. We then filtered the dataset to keep only protein entries with at least one InterPro signature that were divided into EC-annotated and EC-non annotated ones. Table 1 gives some statistical information of the different reference proteomes, including the number of nodes and the percentage of labeled proteins.

**Table 1.** Statistical information about the four proteomes Human, Rat, Mouse, Arabidopsis thaliana. Data were downloaded from UniProtKB/SwissProt 2022 version\*. For each proteome, figures are provided for Total: total protein entries in this proteome, With InterPro: those entries having at least one InterPro domain, With InterPro and EC: those entries having both InterPro and EC number. The percentage of labeled proteins is calculated over the number of entries with InterPro domain.

Proteome	Total	With InterPro	With InterPro and EC	Labeled proteins (%)
Human	20,376	19,656	4,414	22,46 %
Rat	8,174	8,067	2,307	28,60 %
Mouse	17,107	16,959	4,239	25,00 %
<i>A. thaliana</i>	16,202	15,766	5,813	36,87 %

### 3.2 Compared Methods

Our model is compared with other methods, including GrAPFI, support vector machines (SVM), and K-nearest neighbor (KNN). In order to evaluate the protein function annotation results and to compare the effectiveness of the proposed model, three standard evaluation metrics have been used, including accuracy ( $A$ ), precision ( $P$ ), and recall ( $R$ ). The accuracy ( $A$ ) can be computed as follows:

$$A(y, y') = \frac{1}{N} \sum_{i=0}^{n-1} 1(y_i = y'_i) \quad (11)$$

where  $y$  and  $y'$  are the true label and predicted label, respectively.

The precision ( $P$ ) aims to quantify the number of positive label predictions that actually belong to the positive label. This metric can be defined as

$$P = \frac{TP}{TP + FP} \quad (12)$$

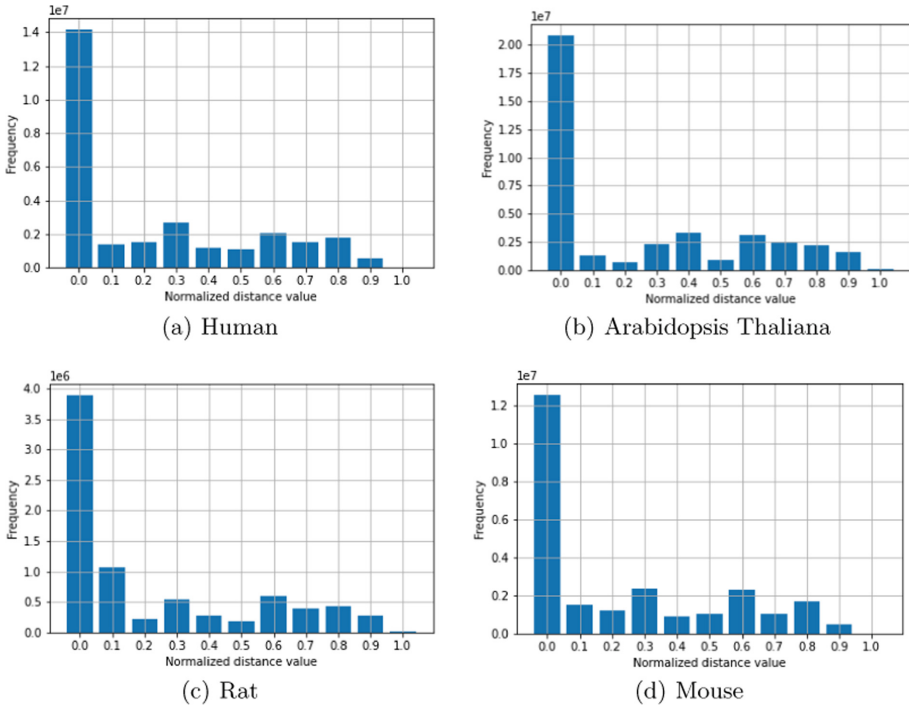
where  $TP$  is the True Positives and  $FP$  is the False Negatives. Furthermore, the Recall ( $R$ ) metric seeks to quantify the number of positive label predictions made out of all positive samples in the dataset, which can be expressed as follows

$$R = \frac{TP}{TP + FN} \quad (13)$$

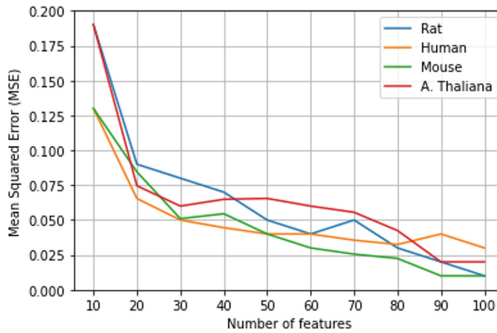
where  $FN$  is the False Negatives.

### 3.3 Parameters Settings

The DAE and GNN models were implemented using the *Keras* framework. The DAE has been trained over 200 epochs with a batch size of 32 training samples. We randomly used 80% of samples from labeled training samples for training and the remaining samples for test. Whereas, after several tests, the GNN model has been trained using a 10-fold cross-validation method on 500 epochs with a batch size of 250 training samples (from labeled proteins). Moreover, we use ADAM as an optimizer for training for both models, where the learning rate is set to  $10^{-3}$ . Furthermore, our DAE network was built as dense neural networks which contains three hidden layers:  $[D, 500, 200, d, 200, 500, D]$  ( $D$  is the number of different domains of all proteins). We opted also to variate the number of extracted features, i.e. the size for the latent representation  $d$  from 10 to 100 features, i.e.,  $d \in [10, \dots, 100]$ . Besides, in the protein graph construction step, we set the threshold  $\alpha$  to 0.1 as an Euclidean distance value to find neighboring proteins and construct the adjacency matrix. Figure 3 reports the computed normalized Euclidean distance with the latent space representation  $Z_i$  versus the frequency. Based on the obtained histograms, we can notice that they are very similar with a majority of distance values very close 0, suggesting that many proteins share with at least one other protein the same domain composition.



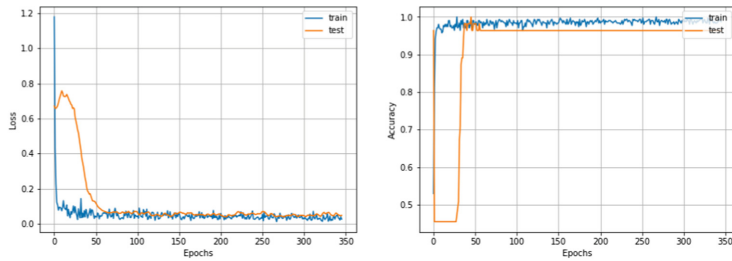
**Fig. 3.** Histograms of the normalized distance computed on the domains of different proteins using four reference proteomes (Human, Arabidopsis Thaliana, Rat, and Mouse).



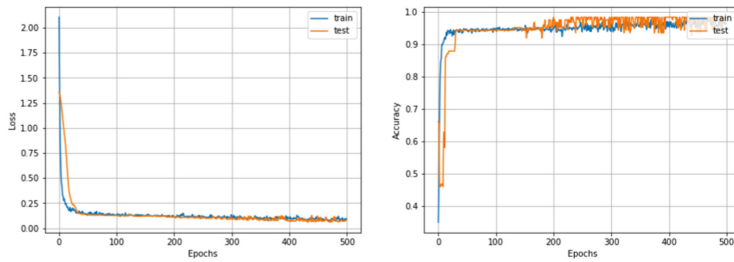
**Fig. 4.** The reconstruction error (MSE) versus the number of extracted features for four reference proteomes using the proposed DAE model.

### 3.4 Deep Representation Learning with the DAE Model

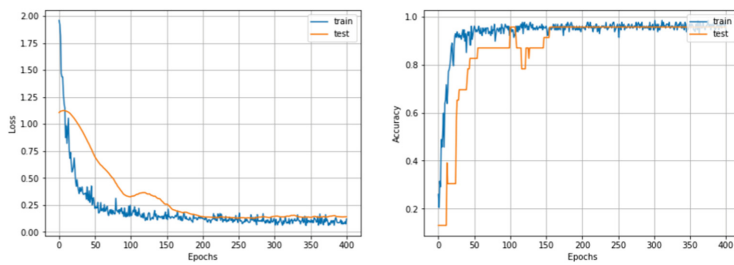
In this section, we evaluate the DAE model in representation learning based on the reconstruction error on the four reference proteomes, i.e., Human, Rat,



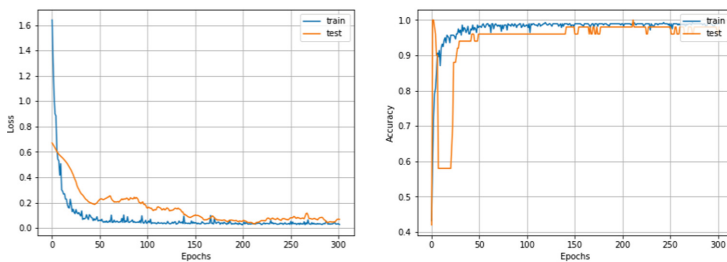
(a) Human



(b) Arabidopsis Thaliana



(c) Rat



(d) Mouse

**Fig. 5.** Best accuracy using the proposed model SGNN on the four datasets.



Arabidopsis Thaliana, and Mouse. We use the MSE metric as a loss function to quantify the reconstruction error between the initial input (InterPro domain composition)  $\mathbf{X}$  and the output (reconstructed domain composition)  $\hat{\mathbf{X}}$ . We report then the average MSE versus the number of extracted features (see Fig. 4). Thus, we can notice that the proposed DAE is able to reconstruct the input using only a few features.

**Table 2.** Best Accuracy, Precision, and Recall using GrAPFI, SVM, KNN, and SGNN on UniProtKB/SwissProt dataset

Model	Accuracy(%)	Precision(%)	Recall(%)	Accuracy(%)	Precision(%)	Recall(%)
	Human			Mouse		
GrAPFI	95.42	94.10	95.27	97.14	95.98	96.66
SVM	90.75	89.45	90.76	92.26	91.87	91.45
KNN	86.78	86.12	84.59	90.78	91.25	91.14
<b>SGNN</b>	<b>97.45</b>	<b>97.21</b>	<b>97.03</b>	<b>98.23</b>	<b>97.76</b>	<b>98.16</b>
	Rat			A. thaliana		
GrAPFI	96.23	96.44	96.71	93.78	94.25	94.77
SVM	89.12	92.12	91.24	93.45	92.34	93.25
KNN	81.45	78.90	83.22	87.45	84.65	88.12
<b>SGNN</b>	<b>98.12</b>	<b>97.95</b>	<b>98.05</b>	<b>97.96</b>	<b>97.21</b>	<b>97.85</b>

### 3.5 Performance Analysis

In this section, we compare our model with other machine learning-based models in the function protein task. Firstly, we report in Fig. 5 the history of loss and accuracy during the training and test. We can observe then that the proposed model gives a good accuracy for the four reference proteomes, e.g., for the Human dataset, we get an accuracy of 97.45%, and a loss which is very close to zero. This proves that the model has been well fitted on used datasets and can provide an accurate annotation. We give also a quantitative comparison in Table 2 on all used datasets using SVM, K-NN, and our proposed model, i.e., SGNN. We can conclude that the SGNN model gives better performances for the four reference proteomes than the other machine learning methods. For the Human dataset, the predicted annotation has as accuracy, precision, and 97.45%, 97.21%, and 97.03%, respectively. In addition, GrAPFI shows also sufficient results, where the accuracy is equal to 95.42%, the precision is equal to 94.10%, and the recall to 95.27%. Whereas, for SVM, and KNN the performance is low compared to the SGNN, and GrAPFI. For the rest of the datasets, the SGNN, and GrAPFI give also better annotation than the rest of the methods, which can prove the added value of the protein graph. That means that the incorporation of the neighboring features between all proteins (graph nodes) in the layers of the neural network can enhance the annotation performance. This can be also extended by using different similarity metrics in the step of protein graph construction [20].

## 4 Conclusion

In this paper, we applied a graph representation learning model for performing protein function annotation from the domain composition of proteins. We used a deep autoencoder model to encode the InterPro domain composition to learn a better representation, i.e., find relevant features for the protein function annotation. Furthermore, we constructed a protein similarity graph based on latent representation to train a graph neural network to perform the protein function annotation. Experimental results demonstrate the effectiveness of our approach in terms of performance in the protein function annotation. In future work, we will extend our approach by adding more features such as keywords and biological pathways, and taxonomy of proteins to improve the performance rates of the proposed model.

## References

1. Aridhi, S., Nguifo, E.M.: Big graph mining: frameworks and techniques. *Big Data Res.* **6**, 1–10 (2016)
2. Cao, Y., Shen, Y.: Tale: transformer-based protein function annotation with joint sequence-label embedding. *Bioinformatics* **37**(18), 2825–2833 (2021)
3. Consortium, U.: UniProt: a hub for protein information. *Nucleic Acids Res.* **43**(D1), D204–D212 (2015)
4. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. *Adv. Neural. Inf. Process. Syst.* **29**, 3844–3852 (2016)
5. Dohan, D., Gane, A., Bileschi, M.L., Belanger, D., Colwell, L.: Improving protein function annotation via unsupervised pre-training: robustness, efficiency, and insights. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 2782–2791 (2021)
6. Gligorijević, V., et al.: Structure-based protein function prediction using graph convolutional networks. *Nat. Commun.* **12**(1), 1–14 (2021)
7. Hamilton, W.L., Ying, R., Leskovec, J.: Representation learning on graphs: Methods and applications. *arXiv preprint [arXiv:1709.05584](https://arxiv.org/abs/1709.05584)* (2017)
8. Hanachi, R., Sellami, A., Farah, I.R.: Interpretation of human behavior from multimodal brain MRI images based on graph deep neural networks and attention mechanism. In: *VISIGRAPP (4: VISAPP)*, pp. 56–66 (2021)
9. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. *arXiv preprint [arXiv:1609.02907](https://arxiv.org/abs/1609.02907)* (2016)
10. Ko, C.W., Huh, J., Park, J.W.: Deep learning program to predict protein functions based on sequence information. *MethodsX*, p. 101622 (2022)
11. Leon, A., Pastor, O.: Towards a shared, conceptual model-based understanding of proteins and their interactions. *IEEE Access* **9**, 73608–73623 (2021)
12. Li, Y., et al.: Deepre: sequence-based enzyme EC number prediction by deep learning. *Bioinformatics* **34**(5), 760–769 (2018)
13. Ma, Y., Li, Q., Hu, N., Li, L.: Sebiograph: semi-supervised deep learning for the graph via sustainable knowledge transfer. *Front. Neurobot.* **15**, 32 (2021)

14. Saidi, R., Aridhi, S., Nguifo, E.M., Maddouri, M.: Feature extraction in protein sequences classification: a new stability measure. In: Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine, pp. 683–689 (2012)
15. Sarker, B., Khare, N., Devignes, M.-D., Aridhi, S.: Graph based automatic protein function annotation improved by semantic similarity. In: Rojas, I., Valenzuela, O., Rojas, F., Herrera, L.J., Ortuño, F. (eds.) IWBBIO 2020. LNCS, vol. 12108, pp. 261–272. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-45385-5\\_24](https://doi.org/10.1007/978-3-030-45385-5_24)
16. Sarker, B., Ritchie, D.W., Aridhi, S.: GrAPFI: predicting enzymatic function of proteins from domain similarity graphs. *BMC Bioinform.* **21**(1), 1–15 (2020)
17. Sarker, B., Ritchie, D.W., Aridhi, S.: Exploiting complex protein domain networks for protein function annotation. In: Aiello, L.M., Cherifi, C., Cherifi, H., Lambiotte, R., Lió, P., Rocha, L.M. (eds.) COMPLEX NETWORKS 2018. SCI, vol. 813, pp. 598–610. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-05414-4\\_48](https://doi.org/10.1007/978-3-030-05414-4_48)
18. Sellami, A., Tabbone, S.: Deep neural networks-based relevant latent representation learning for hyperspectral image classification. *Pattern Recogn.* **121**, 108224 (2022)
19. Singh, P., Singh, N.: Role of data mining techniques in bioinformatics. *Int. J. Appl. Res. Bioinform. (IJARB)* **11**(1), 51–60 (2021)
20. Veras, M.B., et al.: On the design of a similarity function for sparse binary data with application on protein function annotation. *Knowl.-Based Syst.* **238**, 107863 (2022)
21. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y.: A comprehensive survey on graph neural networks. *IEEE Trans. Neural Networks Learn. Syst.* **32**(1), 4–24 (2020)
22. Zhang, C., Freddolino, P.L., Zhang, Y.: Cofactor: improved protein function prediction by combining structure, sequence and protein-protein interaction information. *Nucleic Acids Res.* **45**(W1), W291–W299 (2017)
23. Zhang, J., Chen, Q., Liu, B.: Deepdrbp-2l: a new genome annotation predictor for identifying DNA binding proteins and RNA binding proteins using convolutional neural network and long short-term memory. *IEEE/ACM Trans. Comput. Biol. Bioinform.* (2019)

# **Computational Systems for Modelling Biological Processes**



# Strong Prevalence of the Function over Taxonomy in Human *t*RNA Genes

Yana Nedorez<sup>1</sup>  and Michael Sadovsky<sup>2,3,4</sup>  

<sup>1</sup> School of Fundamental Biology and Biotechnology of Siberian Federal University, Krasnoyarsk, Russia

<sup>2</sup> Institute of Computational Modelling of the Siberian Branch of the Russian Academy of Sciences, Krasnoyarsk, Russia  
`msad@icm.krasn.ru`

<sup>3</sup> V.F. Voyno-Yasenetsky Krasnoyarsk State Medical University, Krasnoyarsk, Russia

<sup>4</sup> Federal Siberian Research and Clinical Center of FMBA of Russia, Krasnoyarsk, Russia

**Abstract.** Human DNA contains many genes encoding transfer RNAs. These genes differ significantly in their primary structure, nucleotide sequence, within the same organism. This study aims to reveal the relationship between the structure of the *t*RNA gene sequence and the amino acids carried by the *t*RNA corresponding to their genes. Raw *t*RNA gene sequences were used as data. After preliminary preparation, the data were analyzed by nonlinear method of dimensionality reduction and data clusterization called elastic maps. The method application revealed the data structure (triplet frequency composition) and desired interplay between the structure and function (the type of amino acid residue to be transferred) over the set of human transfer RNA genes. Some deviations in the distribution of the isodecoders are discussed.

**Keywords:** Triplet frequency · Clustering · Elastic map · Structure

## 1 Introduction

An interplay between the structure and function of genetic entities and the taxonomy of their bearers still challenges researchers. A lot has been done here (see e.g., [4, 7, 8, 17, 24] and much more others). Obviously, the answer depends on the genetic matter taken into consideration: some entities show the substantial prevalence of the taxonomy over function [23], while another matter shows the prevalence of the function over taxonomy [3]. This paper aims to further the studies of the interplay mentioned above.

Here we study the relationship between the function encoded in the transfer RNA human genes and the structure provided by the triplet composition of those genes sequences. Let us now pose the problem more precisely and rigorously. There are three entities: structure of a genetic sequence, the function encoded in it, and taxonomy of the bearer of that former. All of them are interconnected so

that one may expect to reveal an order on a set of genetic sequences determined by the entities or their interplay. To do it, one must rigorously define a structure and function; maybe, taxonomy brings no problem in understanding. The current study does not aim to reveal the impact of taxonomy on the order we seek.

A function seems to be the entity easily to define; despite a great diversity of the functions observed both within a family of genes and for a single gene (see, e.g. [18, 19, 22]), one faces no problem in the determination of the specific function to be considered. Herein, we shall consider the very clearly defined and apparent function of the transfer of various amino acids to a ribosome; see details below.

A structure is much more complicated and diverse in nature; see, e.g. [13–15, 21, 26] and many more. We shall focus on the simplest pattern revealed from DNA sequences that is the triplet frequency dictionary (see below). Let us now describe exactly the goal of the paper and the methodology.

### 1.1 Function–Taxonomy Interplay

To reveal the interplay between structure, function, and taxonomy, we shall go the following way:

- choose the genetic entities with clearly determined and controlled function;
- convert them into a triplet frequency dictionary each;
- use up-to-date and powerful methods to cluster the points (frequency dictionaries) in the relevant metric space and identify the clusters;
- check what is the crucial factor of the clustering: a taxonomy of DNA donor organisms or a function encoded in the sequences.

Suppose the clusters are observed (otherwise, no interplay occurs at all). There are three possible outputs here:

1. the clusters are apparent, and each cluster comprises the sequences encoding the same (or highly proximal) function;
2. the clusters are apparent, and each cluster comprises the sequences belonging to organisms of high taxonomic proximity;
3. a hierarchy in the composition of the clusters takes place: e.g. there are super-clusters gathering the functionally close entities with a fine pattern of each super-cluster determined by taxonomy.

Speaking in advance, let us say that for the genes of human tRNAs, the function is the top leading factor of clustering. Here the transfer of a specific amino acid residue is considered the function.

### 1.2 Peculiarities of Codons Deciphering

Genetic code is known for redundancy: 64 triplets are available to code 20 amino acids (22 in some organisms). As a rule, 61 codons are sense, while the remaining three (usually UAA, UAG, and UAG) reserved for termination. Code redundancy

results in so-called degenerate codons; some of them are grouped into semantic blocks, where each of the synonymous codons codes for the same amino acid. Synonymous codons usually have identical first pairs of bases. While protein synthesis, only the first two bases of a codon interact with the second and third bases of the anticodon strictly complementary, while the third base of the codon and the first base of the anticodon (also known as “wobble position”) can pair either complementarily or through the U:G, G:U or inosine:C interaction [12]. This feature allows one tRNA to bind to different synonymous codons; thus, a cell requires less than 61 tRNA types for interaction between the tRNAs and all sense codons. However, the number of tRNA types with different anticodons varies among species according to their anticodons paring strategies.

Eukaryotic organisms have a strategy of depletion of tRNAs containing arginine or guanine in the first position of the anticodon. For example, each of the amino acids Phe, Tyr, His, Asn, Asp, Cys is encoded by a pair of synonymous codons that differ only in the last base, U or C (UUC and UUU for Phe, ACC and ACU for Tyr, and so on). During protein synthesis, tRNAs containing G in the first base of the anticodon read both NNC and NNU codons of these amino acids. The same is true for those of the four Ser codons in which the last base is a pyrimidine (also U or C). On the contrary, tRNAs containing A in the “wobble position” are used to decode pyrimidine when reading synonymous codons of Val, Pro, Thr, Ala, and some others. A complete list of tRNA types existing in the human body (and their number analyzed in the research) is provided in Table 1.

Selenocysteine-specific tRNAs need special attention. Selenocysteine is the 21<sup>st</sup> proteinogenic amino acid, an analogue of cysteine with the replacement of a sulfur atom with a selenium one. In mRNA world, selenocysteine is encoded by the UGA termination codon followed by a specific stimulatory nucleotide sequence (translational recoding). The structure of tRNA<sup>Sec</sup> differ from those of standard tRNAs. Thus, the acceptor region contains 10 bases (for eukaryotes) and a longer T-loop; additionally, tRNA<sup>Sec</sup> is characterized by the substitution of several rather conservative base pairs [2].

### 1.3 Transfer RNA Genes

A crucial role of tRNA genes and the encoding function bring them to severe evolutionary pressure, forcing them to remain highly conservative. A gene encoding the transfer RNA for a specific amino acid is referred to *isodecoder*. Moreover, isodecoders are the genes encoding tRNAs with the same anticodon differing in the gene sequence. In practice, it results in a significant diversity of the genes coding tRNAs with the same anticodon. The divergence of isodecoders may be as high, as 50% [5,6]. The diversity of isodecoders is also provided by introns embedded into the genes; these former are located pretty close to the anticodon loop.

## 2 Materials and Methods

### 2.1 Genetic Material

We studied the human genes of transfer RNAs to reveal the interplay between structure, function, and taxonomy. These genes encode relatively short ( $\approx 10^2$  b. p. sequences), producing the RNA molecules with the same function: to transfer specific amino acid to the ribosome. In such a capacity, they all have the same function in general and differ in the specificity to the peculiar amino acid residue. Thus, this specificity in amino acid affinity may stand behind the clustering pattern mentioned above, if any.

We used GtRNADB database as the source of genetic material<sup>1</sup> [1], the release of 2019 comprising 421 tRNA genes of high confidence; there are 28 genes with introns (−7%) in this dataset. It is commonplace that a genetic code is redundant: some amino acids are encoded with several different triplets (codons). Additionally, tRNA isodecoders are defined as tRNA molecules sharing the same anticodon but diverging elsewhere in their sequence (up to 274 different tRNA species are produced from 446 genes in humans) [20]. Table 1 enlists the set of isodecoders.

**Table 1.** Abundances of isodecoders of tRNA genes—stands for the codon with no isodecoders in the database.

	U		l		A		G		
U	Phe	–	Ser	9	Tyr	1	Cys	–	U
	Phe	10	Ser	–	Tyr	13	Cys	29	C
	Leu	4	Ser	4	Stop/SeC	1	Stop		A
	Leu	7	Ser	4	Stop		Trp	7	G
C	Leu	9	Pro	9	His	–	Arg	7	U
	Leu	–	Pro	–	His	10	Arg	–	C
	Leu	3	Pro	7	Gln	6	Arg	6	A
	Leu	9	Pro	4	Gln	13	Arg	4	G
A	Ile	14	Thr	9	Asn	–	Ser	–	U
	Ile	3	Thr	–	Asn	22	Ser	8	C
	Ile	5	Thr	6	Lys	12	Arg	6	A
	Met	10	Thr	5	Lys	15	Arg	5	G
G	Val	10	Ala	22	Asp	–	Gly	–	U
	Val	–	Ala	–	Asp	22	Gly	14	C
	Val	5	Ala	8	Glu	7	Gly	9	A
	Val	11	Ala	4	Glu	8	Gly	5	G

Initiator methionine AUG 9 genes

<sup>1</sup> <http://gtrnadb.ucsc.edu/>.



## 2.2 Triplet Frequency Dictionary

Triplet frequency dictionary  $W_j$  is the list of all 64 triplets  $\omega_k$ ,  $k = \text{AAA}, \dots, \text{TTT}$  accompanied with their frequency  $f_{\omega_k}$ ; index  $j$  here enlists the sequences under consideration. To make it, place the reading frame of the length 3 at the very beginning of a sequence and count all the triplets identified by the frame as it moves along a sequence from left to right (for determinacy). Obviously,

$$\sum_{k=\text{AAA}}^{\text{TTT}} f_{\omega} = 1. \quad (1)$$

Such transformation converts a sequence into a point in 63-dimensional metric space; the constraint (1) leaves only 63 linearly independent triplets.

The transformation maps symbol sequences into more convenient mathematical objects that are the points in metric space, thus allowing effective analysis methods. To do it, one must introduce metrics; further, we shall use Euclidean metrics

$$\rho(W_j, W_l) = \sqrt{\sum_{k=\text{AAA}}^{\text{TTT}} (f_k^{(j)} - f_k^{(l)})^2}. \quad (2)$$

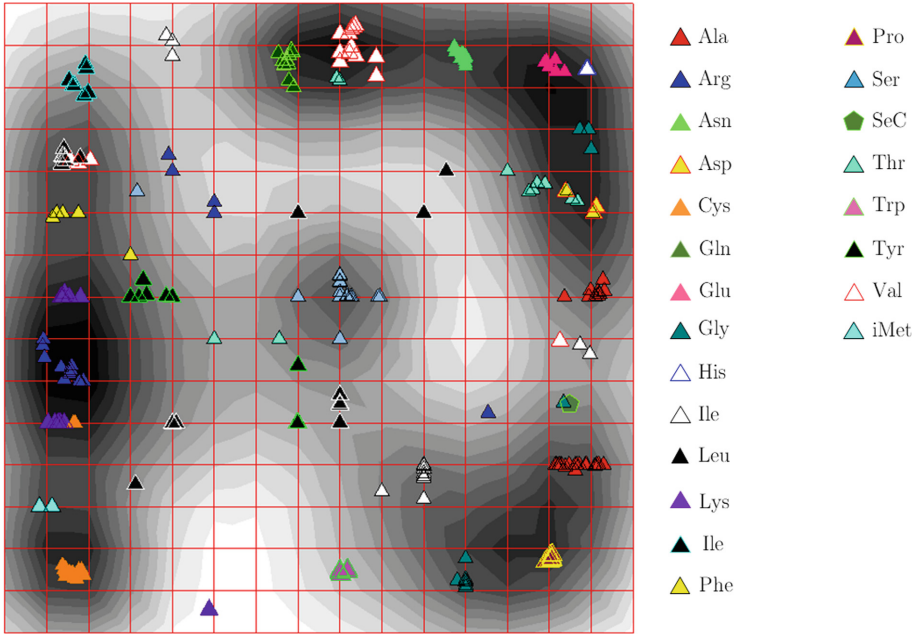
Thus, we investigate the distribution of the points corresponding to genetic sequences in this metric space revealing patterns and clusters, if any.

## 2.3 Clustering and Visualization

A variety of methods to cluster the multidimensional data falls beyond imagination. We use  $k$ -means and the elastic map technique to cluster the data.  $k$ -means is well known linear classification method [9, 11], so we focus on the elastic map technique. It is the non-linear statistics method based on the approximation of the multidimensional data by a manifold of the lower dimension; further, we shall use two-dimensional manifolds [10].

The idea of the method consists of jamming the originally plain manifold (a square in our case) to minimize the total deformation energy of the elastic manifold and mathematical springs connecting the manifold and the projection points. It is a compelling and efficient method to cluster multi-dimensional data and visualize them. A development of elastic map implies specifically organized deformation of elastic membrane so that the best possible fitting of the membrane to the points takes place. To reveal a cluster pattern, one should redefine the images of original points of the jammed surface: an orthogonal projection must be found for each data point.

The next step one should cut-off the mathematical springs so the membrane relaxes back to a straight position. Evidently, this reverse transformation relocates the original points images. Counting the number of the images of the points in an area unit of the plain elastic map, one gets the local density pattern; more rigorously this procedure could be found in [9, 11].



**Fig. 1.** A general pattern of the distribution of isodecoders in 63-dimensional triplet frequency space. Right is the legend explaining the amino acids notation. The cluster structure is revealed through local density of the isodecoders distribution in inner coordinate space.

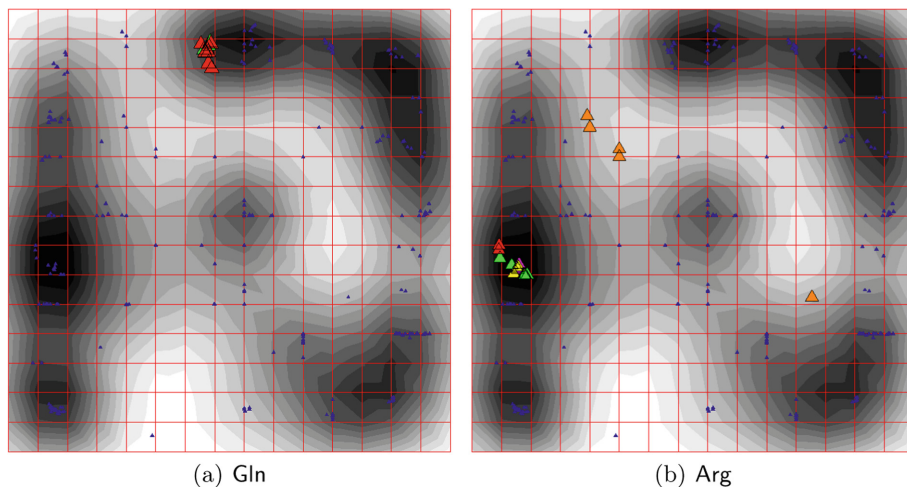
### 3 Results and Discussion

We analyzed the distribution of human tRNA genes converted into triplet frequency dictionaries in 63-dimensional Euclidean space. To do it, we excluded the triplet TAC from the analysis since it has the least standard deviation figure ( $\sigma_{\text{TAC}} = 0.00353$ ); to compare with, the triplet CAG has the greatest standard deviation figure  $\sigma_{\text{CAG}} = 0.11523$ .

Figure 1 shows the distribution of the genes of with respect to encoded amino acids in the inner coordinates of the elastic map provided in 63-dimensional Euclidean space of the triplet frequencies; right is the legend explaining the amino acid code. Doubtlessly, the clusters comprise the genes of tRNAs isodecoders of the same amino acid.

A common fact is that some amino acids are encoded with a few different codons called synonyms. So, the question arises what happens with the isodecoders bearing these different anticodons in terms of clustering pattern. Careful examination of Table 1 shows that some synonyms are absent in the study set of isodecoders. We start by analyzing the distribution of isodecoders of phenylalanine, histidine, aspartic acid, asparagine, and cysteine; typically, they are encoded with two synonyms. However, there is one in our database for each

mentioned amino acid. So, one might check whether the isodecoders bearing the same unique anticodon follow the general pattern of the distribution or deviate from that former.



**Fig. 2.** Examples of the different distribution of the genes of multi-codon *t*RNAs. Figure 2(a) shows the distribution of *t*RNA-Glu genes, two synonym codons isodecoders are present; Fig. 2(b) shows the distribution of *t*RNA-Arg genes, five synonym codons isodecoders are present. (Color figure online)

All these genes except cysteine yield a single specific cluster for themselves. The genes of histidine, cysteine, and aspartic acid are gathered into the dense cluster each; these genes form the dense clusters; however, a single isodecoder (for each gene group) escapes from the cluster to some extent. Cysteine and phenylalanine genes exhibit the farthest escape from the relevant cluster. Hence, one can see that a single cluster pattern is peculiar for the genes with a single synonymous codon. The same distribution pattern of a single dense cluster is inherent for amino acids that do not have synonymous codons. These include tryptophan, represented by a single UGG codon, and selenocysteine, encoded by the UGA stop codon.

Some amino acids occupy the opposite pole in the set of distribution patterns. Figure 2 illustrates two different patterns of both isoacceptors and isodecoders clustering for *t*RNA genes encoding Gln (Fig. 2(a)) and Arg (Fig. 2(b)). Two synonym codons normally encode gln; the database comprises 19 isodecoders for both isoacceptors (6 for CAA, green triangles and 13 for CAG, red triangles). This set of isodecoders exhibits a single-cluster pattern: both isoacceptors groups of genes are gathered into a single cluster. On the contrary, the distribution pattern for Arg exhibits an opposite behaviour.

Typically, Arg is encoded with six synonyms; the database contains 5 of them (the synonym CGC is absent). The isodecoders for four isoacceptors tend

to gather into a dense cluster; that latter comprises the isodecoders for CGU (7 isodecoders, red triangles), CGG (4 isodecoders, yellow triangles), AGG (5 isodecoders, violet triangles), AGA (6 isodecoders, green triangles) and CGA (6 isodecoders, orange triangles) synonyms. The isodecoders comprising the set for AGA anticodon differ from other ones in the presence of introns.

Isoecoders of all isoacceptors for glutamic acid and proline also are gathered in one solid cluster exhibiting similar to the glutamine model of distribution. Similar to arginine, leucine, isoleucine, and tyrosine show an unusual distribution. All these amino acids are characterized by a particular distribution of isodecoders of one of the isoacceptors: the genes AGA (Arg), TTG (Leu), AUA (Ile) and UAC (Tyr) are distributed throughout the map without forming a dense and clearly isolated cluster, unlike the genes of the other isoacceptors the above amino acids. The behaviour of “diffusing” isodecoders may be explained by incorporating introns into the relevant genes. It should be stressed that the introns in tRNA genes are highly conservative; maybe this fact manifests in the dispersion of their distribution.

The isodecoders show another example of an unusual distribution for alanine, lysine, and valine. Similar to cysteine and phenylalanine, isodecoders for the same isoacceptor of these amino acids are divided into two separated groups.

## 4 Conclusion

Here we studied the relationships between the amino acids affinity to tRNA and the ensemble of genes encoding these molecules. The paper brings an evidence for the strong prevalence of the function (that is the type of amino acid transferred by tRNA) over all other issues that might affect the cluster pattern formation. This result is of high scientific merit, since it unambiguously and rigorously addresses the key problem of up-to-date molecular biology. That is the fine pattern in *structure*, *function* and *taxonomy* interplay. The presented results do not involve an effect of taxonomy. Nonetheless, the general approach implemented in this paper provides an exhaustive analysis of the problem.

Recently [16,25], some other types of a structuredness have been reported. Further, we shall check the distribution of those types of tRNA against the observed pattern so that some fine details of the interplay between structure and function might be revealed. Similarly, this approach should be extended for the combined study of tRNA genes ensembles belonging to various organisms ranging from proximal relatives (say, gorilla) to pathogenic and/or obligatory microflora. More specifically, some deeper sight on the problem could be achieved through an analysis of various combinations of genetic entities differing either in taxonomy (say, a mutual analysis of the set of tRNA-Arg genes belonging to different organisms of the same class), or in function of the studied genetic entities.

## References

1. Chan, P.P., Lowe, T.M.: GtRNADB: a database of transfer RNA genes detected in genomic sequence. *Nucleic Acids Res.* **37**(suppl.1), D93–D97 (2008). <https://doi.org/10.1093/nar/gkn787>
2. Donovan, J., Copeland, P.R.: The efficiency of selenocysteine incorporation is regulated by translation initiation factors. *J. Mol. Biol.* **400**(4), 659–664 (2010)
3. Fedotovskaya, V., Sadovsky, M., Kolesnikova, A., Shpagina, T., Putintseva, Y.: Function vs. taxonomy: further reading from fungal mitochondrial ATP synthases. In: *IWBBIO*, pp. 438–444 (2020)
4. Frappat, L., Sciarrino, A.: Conspiracy in bacterial genomes. *Phys. Statistical Mech. Appl.* **369**(2), 699 – 713 (2006). <https://doi.org/10.1016/j.physa.2006.02.008>, <http://www.sciencedirect.com/science/article/pii/S0378437106001993>
5. Geslain, R., Pan, T.: Functional analysis of human tRNA isodecoders. *J. Mol. Bio.* **396**(3), 821–831 (2010)
6. Goodenbour, J.M., Pan, T.: Diversity of tRNA genes in eukaryotes. *Nucleic Acids Res.* **34**(21), 6137–6146 (2006)
7. Gorban, A.N., Zinovyev, A.Y.: The mystery of two straight lines in bacterial genome statistics. *Bull. Math. Biol.* **69**(7), 2429 – 2442 (2007). <https://doi.org/10.1007/s11538-007-9229-6>
8. Gorban, A., Popova, T., Zinovyev, A.: Codon usage trajectories and 7-cluster structure of 143 complete bacterial genomic sequences. *Phys. Stat. Mech. Appl.* **353**, 365 – 387 (2005). <https://doi.org/10.1016/j.physa.2005.01.043>, <http://www.sciencedirect.com/science/article/pii/S0378437105000828>
9. Gorban, A.N., Zinovyev, A.: Principal manifolds and graphs in practice: From molecular biology to dynamical systems. *Int. J. Neural Syst.* **20**(03), 219–232 (2010). <https://doi.org/10.1142/S0129065710002383>, pMID: 20556849
10. Gorban, A.N., Zinovyev, A.Y.: Fast and user-friendly non-linear principal manifold learning by method of elastic maps. In: 2015 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2015, Campus des Cordeliers, Paris, France, 19–21 October 2015, pp. 1–9 (2015). <https://doi.org/10.1109/DSAA.2015.7344818>
11. Gorban, A., Sumner, N., Zinovyev, A.: Topological grammars for data approximation. *Appl. Math. Lett.* **20**(4), 382 – 386 (2007). <https://doi.org/10.1016/j.aml.2006.04.022>, <http://www.sciencedirect.com/science/article/pii/S0893965906001856>
12. Grosjean, H., de Crécy-Lagard, V., Marck, C.: Deciphering synonymous codons in the three domains of life: co-evolution with specific tRNA modification enzymes. *FEBS Lett.* **584**(2), 252–264 (2010)
13. Hong, F., Zhang, F., Liu, Y., Yan, H.: DNA origami: scaffolds for creating higher order structures. *Chem. Rev.* **117**(20), 12584–12640 (2017). <https://doi.org/10.1021/acs.chemrev.6b00825>, pMID: 28605177
14. Jia, H., Gong, P.: A structure–function diversity survey of the RNA-dependent RNA polymerases from the positive-strand RNA viruses. *Front. Microbiol.* **10**, 1945 (2019)
15. Jin, X., et al.: Similarity/dissimilarity calculation methods of DNA sequences: a survey. *J. Mol. Graph. Model.* **76**, 342–355 (2017)
16. Lu, Z., Filonov, G.S., Noto, J.J., Schmidt, C.A., Hatkevich, T.L., Wen, Y., Jaffrey, S.R., Matera, A.G.: Metazoan tRNA introns generate stable circular RNAs in vivo. *RNA* **21**(9), 1554–1565 (2015)

17. Mascher, M., Schubert, I., Scholz, U., Friedel, S.: Patterns of nucleotide asymmetries in plant and animal genomes. *Biosystems* **111**(3), 181–189 (2013)
18. Mekhedov, S., de Ilárduya, O.M., Ohlrogge, J.: Toward a functional catalog of the plant genome. a survey of genes for lipid biosynthesis. *Plant Physiol.* **122**(2), 389–402 (2000)
19. Nikaido, M., Law, E.W., Kelsh, R.N.: A systematic survey of expression and function of zebrafish frizzled genes. *PloS one* **8**(1), e54833 (2013)
20. Pan, T.: Modifications and functional genomics of human transfer RNA. *Cell Res.* **28**(4), 395–404 (2018)
21. Pechal, J.L., Schmidt, C.J., Jordan, H.R., Benbow, M.E.: A large-scale survey of the postmortem human microbiome, and its potential to provide insight into the living health condition. *Sci. Rep.* **8**(1), 1–15 (2018)
22. Philip, M., Chen, T., Tyagi, S.: A survey of current resources to study lncRNA-protein interactions. *Non-Coding RNA* **7**(2), 33 (2021)
23. Sadovsky, M., Putintseva, Y., Chernyshova, A., Fedotova, V.: Genome structure of organelles strongly relates to taxonomy of bearers. In: Ortuño, F., Rojas, I. (eds.) *IWBIO 2015. LNCS*, vol. 9043, pp. 481–490. Springer, Cham (2015). [https://doi.org/10.1007/978-3-319-16483-0\\_47](https://doi.org/10.1007/978-3-319-16483-0_47)
24. Sadovsky, M.G., Putintseva, J.A., Shchepanovsky, A.S.: Genes, information and sense: complexity and knowledge retrieval. *Theory Biosci.* **127**(2), 69–78 (2008). <https://doi.org/10.1007/s12064-008-0032-1>
25. Schmidt, C.A., Matera, A.G.: tRNA introns: presence, processing, and purpose. *Wiley Interdiscipl. Rev. RNA* **11**(3), e1583 (2020)
26. Vinodhini, R., Suganya, R., Karthiga, S., Priyanka, G.: Literature survey on DNA sequence by using machine learning algorithms and image registration technique. In: Kolhe, M.L., Trivedi, M.C., Tiwari, S., Singh, V.K. (eds.) *Advances in Data and Information Sciences. LNNS*, vol. 39, pp. 55–63. Springer, Singapore (2019). [https://doi.org/10.1007/978-981-13-0277-0\\_5](https://doi.org/10.1007/978-981-13-0277-0_5)



# A Methodology for Co-simulation-Based Optimization of Biofabrication Protocols

Leonardo Giannantoni<sup>(✉)</sup> , Roberta Bardini , and Stefano Di Carlo 

Control and Computer Engineering Department, Politecnico di Torino,  
10129 Turin, Italy

{leonardo.giannantoni,roberta.bardini,stefano.carlo}@polito.it

**Abstract.** Biofabrication processes are complex and often unsatisfactory. Trial-and-error methods are costly and yield only incremental innovation, starting from sub-optimal and poorly represented existing processes. Although computational techniques might support efficient process design to find optimal process configurations, intelligent computational approaches must comprise biological complexity to provide meaningful insights. This paper proposes a novel co-simulation-based optimization methodology for the systematic design of protocols for cell culture and biofabrication. The proposed strategy integrates evolutionary computation and simulation for efficient design space exploration and assessment of candidate protocols. A generic library supports the modular and flexible composition of multiscale and multidomain co-simulation scenarios. The feasibility of the presented approach was demonstrated in the automatic generation of protocols for the biofabrication of an epithelial cell monolayer. The results are twofold. First, the prototype co-simulation library helps build flexible, loosely coupled simulation scenarios. Second, the in-silico experimentation on the use case shows that the proposed approach is a viable first step towards standard and automated design in biofabrication.

**Keywords:** Computational systems biology · Optimization via simulation · Biofabrication

## 1 Introduction

Biofabrication is “*the automated generation of biologically functional products with the structural organization from living cells, bioactive molecules, biomaterials, cell aggregates such as micro-tissues, or hybrid cell-material constructs, through Bioprinting or Bioassembly and subsequent tissue maturation processes*” [8]. Biofabrication in Tissue Engineering and Regenerative Medicine (TERM) has the potential to disrupt clinical and pharmacological research [19]. Yet, biofabrication of complex and large tissues and organs is still out of reach.

Biofabrication processes are highly complex biologically and technologically. Biofabrication requires the application of specific protocols representing the

dynamic configuration of relevant process control parameters, emphasizing the values they assume in space and time. However, the large number of critical parameters implies a vast design space, whose exploration is prohibitive and impairs the results obtainable by common *in vitro* trial-and-error experiments [4]. These include brute-force experimental campaigns and One Factor at A Time (OFAT) strategies exploring ranges of relevant system parameters one at a time while holding the others constant [7]. This approach is expensive in terms of time and resources. Also, it overlooks inter-dependencies among system variables, which impedes linking experimental results with process designs controlling multiple variables at a time. This can result in sub-optimal processes [26].

Automation [10] and digitalization [9,26] make trial-and-error approaches more efficient, reducing operator-dependency and human errors, thus supporting process tracking and control. This dramatically increases the yield of *in vitro* experimental campaigns, allowing a more significant number of experiments, thus a broader exploration of the design space. Yet, making the execution of experiments more efficient does not affect the underlying trial-and-error paradigm. *In silico* Design Space Exploration (DSE) approaches can instead support research design and optimization to maximize information extraction, and process improvement efficiency [12].

This work presents the first step towards optimization via simulation (OvS) for generating optimal biofabrication protocols for defined target products. In particular, the proposed framework follows the model-based simulation-optimization paradigm in which DSE and simulation modules are tightly integrated. The DSE selects the solutions which need to be evaluated by simulation [1]. The proposed method exploits heuristic DSE based on Genetic Algorithms (GA) to increase computational feasibility and combines it with a co-simulation environment relying on white-box simulation models to maximize expressivity and explainability. The original contribution of this paper also includes a library of generic components supporting the modular and flexible composition of co-simulation scenarios. Therefore, the co-simulation can easily combine different models, including the target biological entities (e.g., cells), the biofabrication environment, and the possible stimuli delivered during the biofabrication process. The entire framework is presented, resorting to a selected use case to generate optimal protocols for cultivating two-dimensional epithelial sheets with specific shapes relying on a model including both intracellular and extracellular processes. Experimental results show the capability of the proposed approach and identify a set of significant challenges to stimulate further research in this field.

## 2 Background

Model-based simulation-optimization techniques are part of the broader field of computational process design for biofabrication. Several computational methodologies for biofabrication process design exist in the state-of-the-art. Design-of-Experiments (DoE) [23,26] supports strategic and effective research design



by enabling efficient, systematic exploration and exploitation of complex design spaces [7, 14]. A variety of DoE approaches exist [11], and they prove adequate to tackle multi-factorial problems in the optimization of directed cell differentiation [3, 18], and tissue engineering scaffolds [25]. DoE can be combined with Machine Learning (ML) and Artificial Neural Networks (ANN) to improve the accuracy of the bioprocess model [23].

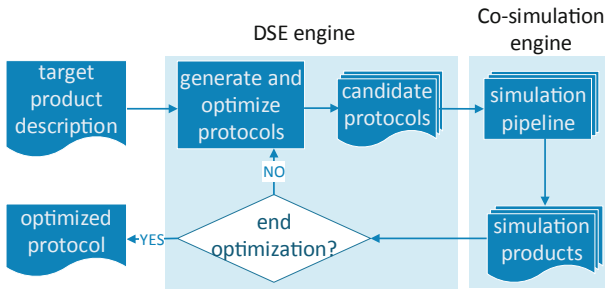
Yet, ML and ANN provide black-box models of the system. Comprehensive modeling of biological complexity is critical for developing computational approaches for biofabrication [5]. To support informed decisions in process design, the ideal model of biofabrication must be accurate, predictive, interpretable and able to analyze process dynamics. Computer simulations provide white-box models of the process, a powerful tool for analyzing complex systems, and particularly their trajectories under different conditions [1].

Simulation and optimization can work together. In optimization via simulation (OvS) methods, optimization can leverage simulations to explore the process design space, and DoE can support the design of simulations campaigns [11]. OvS leverages the simulation model of a physical process to explore its dynamic behavior after specific stimuli, where the parameter values are systematically varied to find the most performing combination towards a target objective [2]. OvS includes model-based and metamodel-based approaches [1]. In model-based OvS, the optimization engine selects the solutions evaluated by simulation. Model-based approaches combine the accuracy and interpretability of simulations with the systematic exploration of the process design space provided by optimization. This fulfills the requirements of biofabrication process optimization, yet it poses strong limitations in terms of computational feasibility when the modeled system is complex. A strategy to reduce the computational complexity of OvS is to include a metamodel that estimates input-output relations of the simulation model to significantly reduce the computational time at the cost of accuracy [17] and a partial fallback to black-box modeling. Also, in this case, white-box is preferable to black-box modeling since interpretability and explainability of OvS results build their relevance for the design of an actual biofabrication process. Heuristic methods allow us to find an approximate solution faster than full-space search methods by trading accuracy and completeness for speed while maintaining the simulated model intact. Among them, the Genetic Algorithm (GA) mimics biological evolutionary dynamics where solutions in the design space undergo a process similar to natural selection [15].

### 3 Methods

Figure 1 summarizes the main architecture of the presented framework, including a Design Space Exploration (DSE) engine for the generation of biofabrication protocols and a simulation engine for testing them. The framework receives the high-level specification of the target product and iteratively computes a biofabrication protocol optimized to grow it. The DSE assembles potential biofabrication protocols and feeds them to simulation instances. Simulation results

are compared against the specifications of the target product used to rank the corresponding protocols and generate new ones at the next iteration. This procedure continues until an optimal protocol is produced, a predetermined number of iterations is reached, or the protocol performance stalls for a given number of iterations. To help the reader, the paper introduces the proposed framework using a running use case focusing on the fabrication of a human epithelial cells monolayer with selected shapes.



**Fig. 1. A high-level representation of the simulation-optimization pipeline.** Given a target product, the DSE engine generates a pool of candidate protocols to be simulated. The products obtained by simulation are compared to the desired target. The previous steps are iterated until an optimal protocol is found.

### 3.1 Use Case Description

As a proof of concept, this work presents the generation of protocols for fabricating human epithelial sheets. To this end, the proposed use case includes a computational model of a population of epithelial cells, modeling intracellular and extracellular processes.

The intracellular model is a Boolean Network (BN) based on a published and well-documented work synthesizing epithelial cells behavior (i.e., survival, proliferation, and apoptosis) in response to a combination of cues [22]. These include environmental factors such as cell density, extracellular matrix stiffness, and growth factor signaling. The high abstraction level of this model allows for low computational complexity and easy integration of new knowledge. A graphical representation of the used Boolean network is available in Fig. 3 of the above paper.

The extracellular model describes interactions among cells and between cells and the environment. It models a discrete 3D grid supporting cells evolving on an extracellular matrix (ECM) surface and interacting with neighboring cells and environmental stimuli.

Biofabrication of a target product can be guided in this model by administering growth factors (GF) at a given 3D coordinate, i.e., molecules that stimulate cell proliferation, and by exposing it to TNF-related apoptosis-inducing ligand

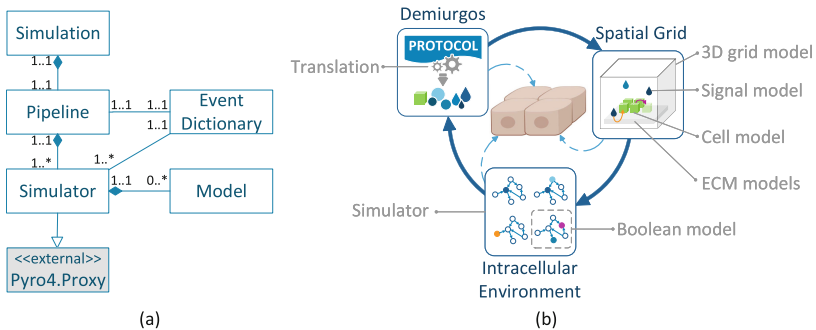
(TRAIL), a protein inducing cell death by apoptosis [24]. The biofabrication process can also control the deposition of cells in the culturing environment.

The two models are coupled and interact through specific inputs. For instance, the intracellular model includes a `CellDensity_High` node, which is used by the Boolean equation to determine the value for the `Replication` node. If, according to the extracellular model, a cell ends up in a very dense area, the cell is informed by setting its `CellDensity_High` to `True`. This, in turn, affects the cell's ability to replicate, thus simulating the inhibition of proliferation by contact inhibition.

### 3.2 Co-simulation Engine

The co-simulation engine interacts with the DSE engine to simulate and evaluate the candidate biofabrication protocols. It sets up and evolves the biological system for a predetermined number of simulation steps, administering stimuli according to the protocol under test. As described in Subsect. 3.1, biofabrication requires the co-simulation of intertwined aspects, each based on a different formalism. Therefore, different simulators must be connected through an interface to exchange data and handle different scales and domains. Several freely-available libraries were tested to ease such implementation (e.g., Mosaik [21]). However, they cannot dynamically change the topology of the simulators required to handle the intrinsic dynamical nature of biological systems.

To overcome this limitation, a prototypal co-simulation framework was developed. It is a Python library of generic components that can set up loosely-coupled co-simulation scenarios, either standalone or associated with a DSE engine. It supports multiscale and multidomain systems and provides mechanisms for transparent distributed execution and third-party software encapsulation.



**Fig. 2. Co-simulation framework.** (a) UML diagram of the simulation framework. (b) Co-simulation scenario for the modeled use case.

Figure 2(a) depicts the overall architecture of the simulation framework. A `Simulation` encapsulates a `Pipeline` of `Simulators`, each executing an arbitrary number of `Model` entities. This design provides common interfaces to

ensure interoperability between multiscale and multidomain simulators, either custom or pre-existing, that can be transparently instantiated on local or remote machines relying on the Pyro4 library [13]. Flexible composition and clear separation are coupled with an ad-hoc loose-coupling mechanism (i.e., it does not involve an orchestrator) for information exchange using a shared **Event Dictionary** collecting and relaying all the events in the simulation pipeline. The high degree of modularity and integration does not enforce consistent conceptual interrelations. Therefore, the data exchanged between the simulators might need suitable translation layers provided by intermediate simulators. With this architecture, a simulation scenario can be easily set up through a single file listing the simulators, both local and remote, and their configurations (Listing 1.1).

```

1 # Composition of the simulation scenario
2 SIMS = {
3     'Local_Sim_Name' : { 'python': 'library.simulator1:Sim1' },
4     'Remote_Sim_Name': { 'remote': 'myuser@remotemachine.domain.it:9999' },
5     ... }
6 # Configuration parameters for local simulators
7 SIM1_CONFIG = {
8     "TIMESCALE": ...,
9     "PARAM2": ...,
10    ... }
11 # Configuration parameters for the simulation
12 SIMULATION_CONFIG = {
13     "SIMULATION_STEPS": 1000,
14     "SIM1_CONFIG": SIM1_CONFIG,
15     ... }

```

**Listing 1.1.** Configuration example for the simulation engine.

The considered use case is implemented by the co-simulation setup illustrated in Fig. 2(b), supported by the above-described library. Two separate discrete-event simulators, Spatial Grid and Intracellular Environment are dedicated to simulating a discrete 3D grid model with object instances (cells, ECM, signals) and BN models of cells. A third simulator (Demiurgos) translates and administers the protocol commands to the appropriate simulator. Demiurgos, like its Platonic entity namesake, is the means through which protocols manifest in and influence the simulation universe. It acts as a purely functional layer (i.e., it does not simulate any entity) by translating and delivering the culturing protocol under simulation.

The Spatial Grid instantiates cells, ECM, and signal objects, as instructed by Demiurgos, and manages the assignment of unique universal identifiers (UUIDs) to cells. It also mimics the diffusion of GFs and TRAIL in the culturing space with a simplified algorithm. At each simulation step, each signal is displaced according to its drift attribute, which decays over time. Signals are removed from the simulation when their keepalive counter reaches zero. With a random probability, each signal reaching the same coordinate of a cell is tagged to be *consumed* by it and removed from the simulation. If a cell is in an area with a high density of neighboring cells, Spatial Grid prevents replication and communicates to the cell that it is in a high-density area. Such information, in turn, might affect that cell's behavior in the Intracellular Environment (see Subject. 3.1).

The Intracellular Environment is informed about a newly issued cell's UUID if the cell can replicate.

The Intracellular Environment manages Boolean network entities modeled after [22] and relies on the PyBoolNet library [16]. It spawns new entities with the UUID provided by Spatial Grid. It feeds them a Boolean input modified by both the protocol and the events coming from the extracellular environment, thus informing them about the cell density of their surroundings, the quality of their supporting ECM, and the availability of nutrients and other signals. Suppose a cell entity enters an apoptotic state. In that case, Intracellular Environment removes it and broadcasts its UUID so that the other simulators in the pipeline (in this case, Spatial Grid) remove the corresponding model too.

### 3.3 Design Space Exploration Engine

A biofabrication protocol is a list of signals organized in time and space to guide a specific biological product synthesis. Optimizing such arbitrary long lists of instructions is a complicated combinatorial problem. Therefore, exhaustive exploration is not an option. Our DSE component employs a Genetic Algorithm (GA), an evolutionary computation metaheuristic, to generate populations of candidate solutions or individuals (i.e., the protocols). In this context, a protocol is an individual characterized by a genome whose genes are the signals composing the protocol. Candidate individuals are ranked based on a fitness function and mutated to evolve the population at each new generation. The proposed DSE engine is built on top of the  $\mu$ GP(microGP) library, a tool tailored to problems whose solutions can be expressed similarly to assembly programs [20].

The individuals defined for the proposed use case scenario are organized in two sections (Listing 1.2). The placing section lists the 3D coordinates of the cells to be deposited at the beginning of the biofabrication process. The signal section lists nutrients and environmental stimuli organized in space and time.

```

1 % placing section
2 CELL (99, 72, 3), (150, 162, 3), (67, 56, 3), ...
3
4 % signals section
5 0: GF LOW (147,84,3), GF LOW (133,101,3), GF HIGH (26,22,3), GF LOW
      (137,158,3), TRAIL (81,148,3), TRAIL (43,8,3), TRAIL (75,177,3), ...
6
7 5: TRAIL (5,24,3), GF HIGH (104,24,3), ...
8
9      :
10     :
11 300: ...

```

**Listing 1.2.** Sample protocol built by the DSE engine.

The *language* used to build the protocols for the epithelial cell model includes CELL, GF LOW/HIGH, and TRAIL macros, each followed by 3D coordinates. They derive from the inputs (cells deposition and exposure to stimuli) specific to the model described in Subsect. 3.1. This abstract and compact representation minimizes the resources required to compute and store the individuals that in  $\mu$ GP are encoded as directed multigraphs constrained by user-defined rules [20].

The DSE engine requires the specification of a target product, i.e., the biological construct obtained at the end of the simulation and a timescale. In the example provided in Listing 1.3, a STRIPES target composed of two parallel planar stripes is described using the sum of two CUBOID primitives from our `geometry` library. An optional bounding box can provide cues to the DSE engine for the assembly of new individuals. The timescale allows tuning the granularity of the protocol for the simulated system. For instance, if the protocol step is set to 5, the signal section of the protocol uses a `simulation.steps/5` length, and Demiurgos issues one protocol instruction every five simulation steps.

The DSE engine uses three genetic operators to evolve the population of candidate protocols. When creating a new offspring, each operator is applied with an initial probability equal to the `strength` parameter  $\alpha$ , which in our setup is equal to 0.9 ( $\alpha \in [0...1]$ ). Strength is a self-adapting parameter. `μGP` increases it when an operator shows a high success rate (i.e., the mutated individuals' fitness improved compared to its parents) and decreases it otherwise. `singleParameterAlterationMutation` chooses a new random value for one parameter of an individual. For instance, it might alter the  $x$  coordinate for the deposition of a signal at protocol step  $n$ . `onePointCrossover` generates two offspring individuals from two parent individuals by recombining them over a single cut point. For instance, given two 100-step protocols, it might cut them at step 20 and swap the parts containing steps 21 to 100. `twoPointCrossover` operates similarly. It chooses two cut points and swaps the middle portion.

The protocols' fitness is assessed by comparing the product obtained by simulation with the target product. For the presented application, the fitness is represented using two values:

$$f_0 = \frac{\text{cellsInsideTargetArea} * 100}{1 + \text{cellsInsideTargetArea} + \text{cellsOutsideTargetArea}} \quad (1)$$

$$f_1 = \frac{\text{cellsInsideTargetArea} * 100}{1 + \text{targetAreaPoints}} \quad (2)$$

Equation 1 ( $f_0$ ) expresses the precision, i.e., the fraction of the biofabricated product that matches the target. Eq. 2 ( $f_1$ ) expresses the coverage, i.e., how much of the target product has been obtained. `targetAreaPoints` is defined as the number of integer 3D coordinates included in the target shape. For instance, given a square target covering a  $m \times n$  area (i.e.,  $m \times n$  `targetAreaPoints`), a fitness  $f = [84.1, 29.2]$  means that 29.2% of the desired product has been obtained (i.e., it covers 29.2% of the  $m \times n$  area), and 84.1% of the material is where expected (inside the target area). That is, the remaining 15.9% of cells is misplaced (outside the target area).

`μGP` provides two methods to evaluate a multi-parameter fitness functions, `Enhanced` and `MultiObjective`. The `Enhanced` method attributes decreasing importance to the  $f_i$  parameters. The `MultiObjective` method attributes the same weight to  $f_0$  and  $f_1$ , thus leading to the choice of the best individuals based on the joint evaluation of the two parameters. The best individual, in this

case, is chosen among those dominating the individuals belonging to the Pareto frontier of the previous generation.

The proposed implementation of the DSE engine employs the `MultiObjective` method, as it is tailored to multi-objective optimization problems. That is, those requiring trade-offs between multiple and potentially conflicting objectives. Cell proliferation is helpful for coverage for the presented use case but must be restrained to avoid abnormal growth. Simultaneously, precision (summarizing a proliferation process under control and limited to a well-defined area) should not prevent obtaining the target product in the desired quantity and shape.

```

1 from library.common import geometry
2 STRIPES = {"DESCR": {
3     "CUBOID 1":
4         {"width": 200, "depth": 25, "height": 1, "origin": (0, 0, 3)},
5     "CUBOID 2":
6         {"width": 200, "depth": 25, "height": 1, "origin": (0, 174, 3)} },
7     "BOUNDING_BOX": ((0, 200), (0, 200), (3, 4)) }
8 CIRCLE = {"DESCR": {
9     "CYLINDER": {"center": (100, 100, 3), "radius": 30, "height": 1}},
10    "BOUNDING_BOX": ((70, 130), (70, 130), (3, 4)) }
11 UGP_CONFIG = { "PROTOCOL_STEP": 5, "TARGET": STRIPES }
```

**Listing 1.3.** Configuration example for the DSE engine.

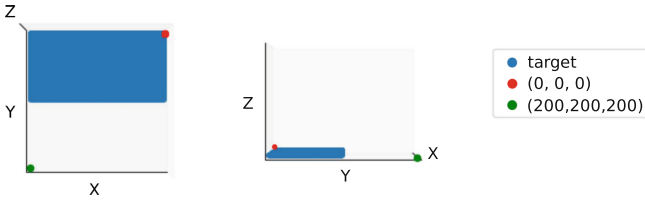
## 4 Results

This section presents the validation strategy employed to demonstrate the functioning of the proposed approach.

### 4.1 Experimental Setup

The experimental setup starts with the definition of a target product, that is, an epithelial cells monolayer covering half the ECM surface (Fig. 3). The culturing environment simulated by the Spatial Grid is a  $200 \times 200 \times 200$  cube, with its base covered by a  $200 \times 200 \times 3$  layer of ECM entities. The target product is then a  $200 \times 100 \times 1$  rectangle lying on the ECM layer. At the beginning of each simulation, the Intracellular Environment sets all new cells in the proliferative state defined in [22]. Demiurgos issues one protocol instruction every five simulation steps. The co-simulation engine evolves the system for 1,500 simulation steps per simulation, stopping in advance if all cells die.

As for this experiment, the placing section of the protocols lists 1 to 15 (*average* = 8, *sigma* = 2) coordinates for cells deposition. The signals section contains 300 (1500/5) instructions, and each instruction contains 0 to 50 (*average* = 15, *sigma* = 5) occurrences of macros (`GF HIGH`, `GF LOW`, and `TRAIL`), as detailed in Subsect. 3.3). The population evolved by  $\mu$ GPIs of *MultiObjective* type, it has an initial size  $\nu = 10$  and a maximum size  $\mu = 10$ . The genetic operators described in Subsect. 3.3 can be applied  $\lambda = 10$  times at every step of the evolution, with a  $\sigma = 0.9$  strength, and an  $\alpha = 0.9$  inertia. To rank protocols by fitness, the DSE engine uses the pair of values  $f = [f_0, f_1]$  (Eq. 1 and Eq. 2), measuring precision and coverage of the target, respectively.

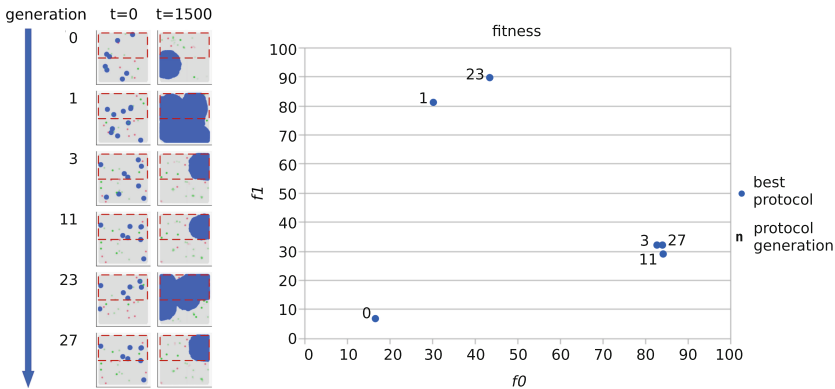


**Fig. 3. Target product.** Left: top view, right: lateral view. The green and red dots help figure out the orientation.

### 4.2 Experimental Results

As of the writing of this document, the experiments have been running for 39 days on an Intel(R) Xeon(R) CPU E5-2680 @ 2.70 GHz with 64 GB RAM, evaluating 884 protocols along 50 generations.

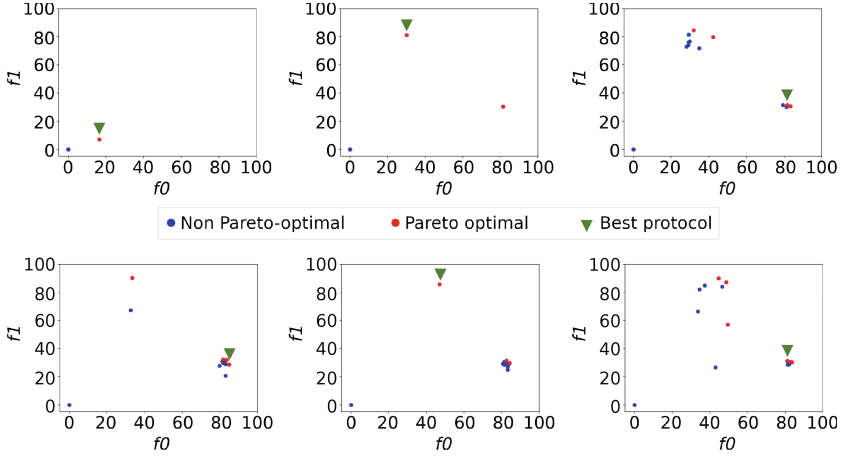
Results obtained demonstrate that (1) the proposed framework proves capable of automatically generating a protocol for the simulated biofabrication of the illustrated target product and use case and that (2) the DSE can drive the optimization toward protocols with better fitness expressed as similarity to a target product.



**Fig. 4. Fitness trend during optimization.** Left: product obtained by the best protocols at the beginning and end of the simulation. The dashed red box highlights the target area. Right: fitness of the best protocols.

Figure 4 (left) shows a 2D view of the Spatial Grid at the beginning ( $t = 0$ ) and end ( $t = 1500$ ) of the simulation of the best protocols identified during different generations of the optimization process. On the right panel of the figure, the chart shows the evolution of the fitness of best protocols along generations. This plot highlights two trends for the fitness of the best protocols consistent with observations performed on all the 884 best protocols (data not shown).





**Fig. 5. Evolution of the Pareto front.** Each chart reports the fitness of the individuals from the same generation, with the Pareto-dominant protocols highlighted in red. The green arrow pinpoints the best individual emerged from that generation. Only generations 0, 1, 3 (top) and 11, 23, 27 (bottom) are shown. (Color figure online)

Some best protocols (generations 3, 11, 27) exhibit higher precision ( $f_0$ ), while others (generations 1, 23) higher coverage ( $f_1$ ). Therefore, the best protocol might be selected from both clusters, depending on the maximum lifetime of the individuals and the advancement of the Pareto front.

Figure 5 shows the evolution of the Pareto front, taking into account only the protocols evaluated in the same generation to which the best protocols belong. The best protocol, indicated by a green arrow, is chosen by  $\mu$ GP among the red dots in the image. According to  $\mu$ GP definition of the MultiObjective optimization, “two fitness may be equal, may dominate each other, meaning that all the components of one fitness are greater or equal to the corresponding parts of the other, or they may be not comparable” [20].

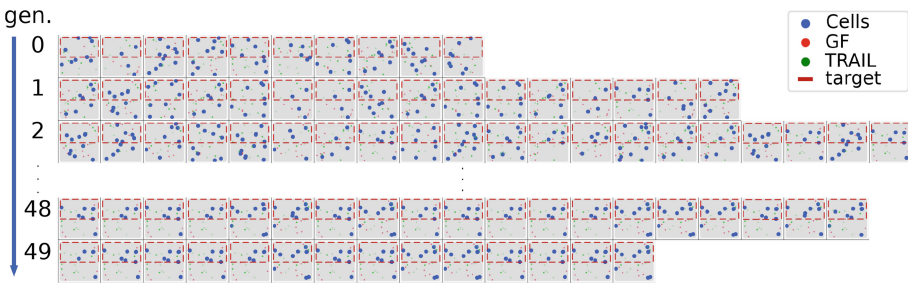
Figure 6 shows per each generation (rows) different initial configurations of the cells (blue), provided by the placing section of the protocols at the beginning of each simulation. 2D views of the Spatial Grid at  $t=0$  represent different simulations for each generation, corresponding to other simulated protocols. At the beginning of the optimization process (generations 0, 1, and 2, top three rows), the DSE engine generates protocols as random individuals. After several rounds of evolution (generations 48 and 49, bottom couple of rows), the initial configuration of cell placing shows more consistency among different simulated protocols.

Indeed, both Fig. 4 and Fig. 6 show that, through the generations, the initial placing of the cells gradually shifts and concentrates towards the target area. Figure 7 shows the same simulated protocols at the end of the simulation ( $t=1500$ , or  $t$  corresponding to death of all the cells). In the first three generations (top three rows), few protocols guarantee cells survival to the end of the

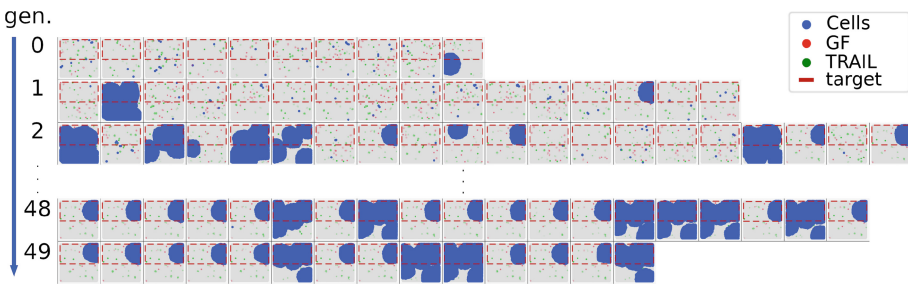
simulation. Indeed, those simulations stopped before executing the 1,500 steps (we show the last simulation step with alive cells), corresponding to very low protocol fitness. The two bottom rows (generations 48 and 49) demonstrate that those very poorly fit individuals have permanently given way to protocols that are indeed able to grow conglomerates of cells.

Figure 7 indicates the same two-fold tendencies that Fig. 4 discovered: some of the fittest protocols yield higher precision, others higher coverage.

The source code and the results obtained are available on GitHub and archived in Zenodo [6].



**Fig. 6. Initial configuration of the cells (blue) obtained from the simulated protocols at the beginning of the simulation.** At the beginning of the optimization process, the top three rows are from generations 0–2 (random individuals). The bottom two after several rounds of evolution (generations 48 and 49). The red and green clouds represent Trail and GF signals, respectively. The dashed red box highlights the target area. (Color figure online)



**Fig. 7. Final configuration of the cells (blue) obtained from the simulated protocols after 1500 simulation steps.** At the beginning of the optimization process, the top three rows are from generations 0–2 (random individuals). The bottom two rows (generations 48 and 49) show the progress after several rounds of evolution. The red and green clouds represent Trail and GF signals, respectively. The dashed red box highlights the target area. (Color figure online)

## 5 Conclusions

In this work, we presented a simulation-optimization methodology for generating biofabrication protocols and a co-simulation framework supporting our strategy. To the best of our knowledge, we are the first to propose this kind of approach. We chose the human epithelium as a use case to validate our methodology and demonstrate the developed framework's usefulness.

Our results are twofold. First, the prototype framework backing our simulations helps build flexible loosely-coupled co-simulation scenarios. Secondly, the preliminary experimental results show that the proposed approach might provide viable support to biofabrication process design.

In the future, we plan to expand this work in several directions, first, by addressing hyperparameter optimization for the DSE engine. That is the exploration and tuning of optimal GA parameters. Second, by integrating better quantitative models for the realization of more accurate digital twins for both the biofabrication process and the modeled biological system. Finally, we plan to extend our use case by adding cells differentiation so that their diverse functional and phenotypical types let us build more complex products. We are already taking steps towards a massive parallelization, which would allow faster experimentation of larger and more complex biological systems.

While still in its infancy, we can foresee this new methodology as the first step towards standard and automated design in biofabrication.

## References

1. do Amaral, J.V.S., Montevechi, J.A.B., de Carvalho Miranda, R., de Sousa Junior, W.T.: Metamodel-based simulation optimization: a systematic literature review. *Simul. Model. Pract. Theory* **114**, 102403 (2022)
2. Amaran, S., Sahinidis, N.V., Sharda, B., Bury, S.J.: Simulation optimization: a review of algorithms and applications. *Ann. Oper. Res.* **240**(1), 351–380 (2015). <https://doi.org/10.1007/s10479-015-2019-x>
3. Bukys, M.A., et al.: High-dimensional design-of-experiments extracts small-molecule-only induction conditions for dorsal pancreatic endoderm from pluripotency. *Iscience* **23**(8), 101346 (2020)
4. Eggert, S., Hutmacher, D.W.: In vitro disease models 4.0 via automation and high-throughput processing. *Biofabrication* **11**(4), 043002 (2019)
5. Geris, L., Lambrechts, T., Carlier, A., Papantonioni, I.: The future is digital: in silico tissue engineering. *Curr. Opin. Biomed. Eng.* **6**, 92–98 (2018)
6. Giannantoni, L.: coherence v1.0.0 (2022). <https://doi.org/10.5281/zenodo.6462768>, <https://github.com/smilies-polito/Coherence/releases/tag/v1.0.0>
7. Gilman, J., Walls, L., Bandiera, L., Menolascina, F.: Statistical design of experiments for synthetic biology. *ACS Synth. Biol.* **10**(1), 1–18 (2021)
8. Groll, J., et al.: Biofabrication: reappraising the definition of an evolving field. *Biofabrication* **8**(1), 013001 (2016)
9. Herwig, C., Pörtner, R., Möller, J.: Digital Twins: Applications to the Design and Optimization of Bioprocesses, vol. 177. Springer, Heidelberg (2021). <https://doi.org/10.1007/978-3-030-71656-1>

10. Hofmann, P., Samp, C., Urbach, N.: Robotic process automation. *Electron. Mark.* **30**(1), 99–106 (2019). <https://doi.org/10.1007/s12525-019-00365-8>
11. Jankovic, A., Chaudhary, G., Goia, F.: Designing the design of experiments (doe)-an investigation on the influence of different factorial designs on the characterization of complex systems. *Energy Build.* **250**, 111298 (2021)
12. Jazdi, N., Talkhestani, B.A., Maschler, B., Weyrich, M.: Realization of AI-enhanced industrial automation systems using intelligent digital twins. *Procedia CIRP* **97**, 396–400 (2021)
13. de Jong, I.: Pyro - python remote objects (2020). <https://pyro4.readthedocs.io>
14. Kasemiire, A., et al.: Design of experiments and design space approaches in the pharmaceutical bioprocess optimization. *Eur. J. Pharmaceut. Biopharmaceut.* **166**, 144–154 (2021)
15. Katoch, S., Chauhan, S.S., Kumar, V.: A review on genetic algorithm: past, present, and future. *Multimedia Tools Appl.* **80**(5), 8091–8126 (2020). <https://doi.org/10.1007/s11042-020-10139-6>
16. Klarner, H., Streck, A., Siebert, H.: Pyboolnet: a python package for the generation, analysis and visualization of boolean networks. *Bioinformatics* **33**(5), 770–772 (2017)
17. Kleijnen, J.P.: Simulation-optimization via kriging and bootstrapping: a survey. *J. Simul.* **8**(4), 241–250 (2014)
18. Kuterbekov, M., Machillot, P., Baillet, F., Jonas, A.M., Glinel, K., Picart, C.: Design of experiments to assess the effect of culture parameters on the osteogenic differentiation of human adipose stromal cells. *Stem Cell Res. Therapy* **10**(1), 1–9 (2019)
19. Moroni, L.: Biofabrication: a guide to technology and terminology. *Trends Biotechnol.* **36**(4), 384–402 (2018)
20. Sanchez, E., Schillaci, M., Squillero, G.: Evolutionary optimization: the  $\mu$ gp toolkit (2011)
21. Schütte, S., Scherfke, S., Tröschel, M.: Mosaik: a framework for modular simulation of active components in smart grids. In: 2011 IEEE First International Workshop on Smart Grid Modeling and Simulation (SGMS), pp. 55–60. IEEE (2011)
22. Sizek, H., Hamel, A., Deritei, D., Campbell, S., Regan, E.R.: Boolean model of growth signaling, cell cycle and apoptosis predicts the molecular mechanism of aberrant cell cycle progression driven by hyperactive pi3k. *PLOS Comput. Biol.* **15**, e1006402 (3 2019)
23. Walsh, I., Myint, M., Nguyen-Khuong, T., Ho, Y.S., Ng, S.K., Lakshmanan, M.: Harnessing the potential of machine learning for advancing “quality by design” in biomanufacturing. In: *Mabs*, vol. 14 (1), p. 2013593. Taylor & Francis (2022)
24. Wiley, S.R.: Identification and characterization of a new member of the TNF family that induces apoptosis. *Immunity* **3**(6), 673–682 (1995)
25. Zhang, S., Vijayavenkataraman, S., Lu, W.F., Fuh, J.Y.: A review on the use of computational methods to characterize, design, and optimize tissue engineering scaffolds, with a potential in 3d printing fabrication. *J. Biomed. Mater. Res. Part B Appl. Biomater.* **107**(5), 1329–1351 (2019)
26. Zobel-Roos, S., Schmidt, A., Uhlenbrock, L., Ditz, R., Köster, D., Strube, J.: Digital twins in biomanufacturing. In: Herwig, C., Pörtner, R., Miller, J. (eds.) *Digital Twins. ABE*, vol. 176, pp. 181–262. Springer, Cham (2020). [https://doi.org/10.1007/10\\_2020\\_146](https://doi.org/10.1007/10_2020_146)



# A 3D Multicellular Simulation Layer for the Synthetic Biology CAD Infobiotics Workbench Suite

Richard Oliver Matzko<sup>(✉)</sup>, Laurentiu Mierla, and Savas Konur

Department of Computer Science, University of Bradford, Bradford, UK  
{r.matzko, l.m.mierla, s.konur}@bradford.ac.uk

**Abstract.** Although computational solutions are commonplace in Synthetic Biology laboratories that use software, proprietary or otherwise, the use of multicellular simulations promises to enhance such workflows by *in silico* prototyping via spatiotemporal biological simulations prior to the *in vitro* genetic manipulation of cell lines. Thus, a multicellular layer for the Infobiotics Workbench Synthetic Biology platform was pursued with a view towards impactful multicellular case studies. This work interrogates and benchmarks a variety of multicellular modalities, including the application of parallel subcellular stochastic simulations for computing biochemical network models. Phenotypic and microenvironmental characteristics were meticulously reviewed to effectively construct the multicellular designs that could elicit emergent population level consequences. The result was the benchmarking of a batch processed solution compared to one utilizing Unreal Engine 4 in real time, along with the elucidation of relative characteristics and performances. Such simulations proved very computationally intensive as prototyped on conventional hardware, hence this work alluded to the need for high performance computing especially regarding biochemical parallelization.

**Keywords:** Synthetic · Biology · Multicellular · Stochastic · Simulation · CAD

## 1 Introduction

The objective of Synthetic Biology has been described as the utilization of biology technologically [1], especially from the DNA level, for essentially unlimited possible outcomes. The challenge explored here is in elevating CAD (computer assisted design) to the multicellular level. Data is available within various repositories upon which models and simulations can be constructed. In fact, bioregulatory models acquired from repositories can be harnessed and applied dynamically to spatiotemporal simulations [2]. Thus, computers are poised for the computer assisted design of blueprints, upon which *in silico* proofing can be performed, with potential parameter optimization [3] and finally laboratory application and/or verification [4] from ‘*in silico* first’ efforts.

The integration of the NGSS (Next Generation Stochastic Simulator) [5] into two unique multicellular simulation layers was pursued, with subsequent benchmarking of

performances with and without this subcellular processing layer. In this way the scalability, tractability and feature differences could be assessed, commencing at the personal computing level. With NGSS integrated into the Infobiotics Workbench (IBW) platform [6], the present work pursued the extension of this Synthetic Biology CAD system to the multicellular context through the mutual SBML [7] model exchange format. Multicellularity was found to be absent or limited in such suites [1, 6, 8], especially with regards to compelling physical, spatiotemporal 3D solutions. Coupled with a phenomenological assessment of cellular behavior and the respective microenvironment (Sect. 2.2), the work sets the foundation for promising developments towards increasingly realistic, instructive and useful multicellular simulations with diverse, emergent spatiotemporal potential. Such elucidation coordinates with the phenomenological approach that was encouraged in the literature [9].

Unreal Engine 4 (UE4) as a real-time platform (including physics) for multicellular simulation and CAD design was assessed as contrasted to a rules-based batch-processed dynamic mesh generation approach. The distinction between real-time and batch processed performances in multicellular simulation are not clear from the literature and is one of the critical design aspects to consider when developing a multicellular simulator (see Sect. 2.3 for more details). The work would demonstrate the call for batch processing over real time solutions, as well as performance profiles of parallel NGSS processing that highlighted the need for future high performance computing (HPC) implementations for subcellular models (see Sect. 4.3) in the pursuit of expanding IBW.

## 2 Multicellular Simulation Principles and Technologies

The features of multicellular simulation as well as the data exchange technologies discussed in this section were considered for the construction of the multicellular layers presented in the methods section (Sect. 3) as well as for the subcellular biochemical processing layer used in association with IBW's stochastic simulator (NGSS).

### 2.1 Bioregulatory/Metabolic Simulations and Exchange Standards

SBML is an exchange format designed to exchange modeled data within Systems Biology and between computational modeling and metabolic simulation tools [8]. SBML can be used to capture the mathematics of biochemical reactions and regulatory models, and it has been used for multiscale simulations at the subcellular level [10].

There are a number of simulators available for the chemical level that can solve various biological computations, such as biochemical reactions or state transitions (e.g. transport). These simulators tend to be designed for solving SBML models and have been used to compute subcellular models within multicellular simulators [10]. They can utilize various biochemical simulation modalities [11], including ordinary differential equations (ODEs), stochastic algorithms and flux balance analysis, and may even possess parameter estimation capabilities. Whilst stochastic approaches are computationally expensive, they are considered more principled than ODEs, and the justification for their use has been given [12] by the argument that stochastic models can capture the noise of biochemical systems, whilst also effectively fulfilling the modelling of genetic switches.

It was argued that deterministic ordinary differential equations are incapable of fulfilling these objectives effectively. Hybrid approaches have also been pursued [11], where low particle numbers suited stochastic simulations whilst faster reactions containing more reactants could be solved deterministically. Boolean models provide a convenient alternative solution by avoiding the need for kinetics data [13].

### NGSS and SSAPredict

NGSS [12] possesses one approximate and 8 exact Gillespie stochastic algorithms [6] and was notably incorporated into the Infobotics Workshop Synthetic Biology suite. Stochastic Simulation Algorithms (SSAs) behave equivalently to Chemical Master Equations; a set of probabilistic differential equations. NGSS can operate on a single logical core (i.e. serially) or on multiple CPU logical cores, and outputs average concentrations over one or more parallel runs. The web-based SSAPredict tool [5] can purportedly predict the fastest SSA to use for a given model via topological network property analysis. As will be seen, it is not always correct. According to Sanassy et al. (2015) [5], despite being one of the top 4 algorithms out of the 9, Tau Leaping still performed worse than other algorithms on many occasions. However, Tau Leaping often far outperforms other algorithms for economy of time (see Sect. 4.3) and reportedly has better performance at higher reaction and species graph densities. Thus, SSAPredict should only be treated as a guideline for the best algorithm.

## 2.2 Multicellular Simulator Characteristics and Potential Characteristics

With regards to the mechanisms underlying multicellular simulation, it is apparent that small cellular phenotypic changes can have significant biological implications [2]. The outcome of understanding the genetic and phenotypic properties of cells is the ability to mechanically predict their emergent consequences. Cellular and subcellular phenotypic phenomena can be derived from a variety of multicellular and biological literature sources [2, 13–21] and operate in conjunction with extracellular characteristics [13, 14, 16, 22–24] to produce emergent consequences such as cell sorting [9, 24], morphogenesis [9], patterning [24], fitness [25] and many more. As a reassurance to modelers, it was observed that the emergent phenomena list was far more extensive than the fundamental cellular and extracellular phenomena from which they emerged, although the permutations, including spatial organizations, are innumerable.

## 2.3 Computational Considerations

High performance computing and extensive parallelization is not uncommon with multicellular simulation [22, 24, 26]. Other computational enhancements include the clustering of similar cells phenotypically [25], referred to as ‘super-individuals’ [22], perhaps based on the assumed similarity of the local biochemical microenvironment [26, 27], or into nearest neighbor lists by proximity [22], use of state outputs with subsequent external visualization [22, 26] following “batch processing” [19] as opposed to real-time approaches [16, 20], GPU and CPU parallelization [19, 26, 28] with as few as one cell per CPU, random update ordering [25] to remove bias, Voronoi tessellations to abstract spatial distributions [29], graphical merging of objects [30], client-server architecture



[19] and the use of small scale representations of a functionally identical larger system [27]. Domain-based computing is an essential hallmark of multicellular simulation tractability and computation, allowing for parallelization, as well as structural and functional discretization. Also, with the need to consider multiscale phenomena, multiple timesteps are often used, referred to as a “pseudo steady-state approximation”, because temporal resolutions may be different enough that certain processes are “frozen” during those smaller time steps [22, 31].

The initialization of spatial configurations, or what might be considered ‘bioprinting’ *in silico*, can allow for proportionally distributed heterogenous populations, for example in the cortical layers of neurological tissue simulations [28], thereby bypassing stages of developmental emergence. Initial simulated arrangements of cells [16] as well as model generation [32–34] have also been attempted using micrographs. Multicellular states, emergent or otherwise, could be saved and experimented on *in silico* [2] and manipulated by playback controls [19].

## 2.4 Multicellular Simulation Methodologies

An on-lattice [9] approach refers to a spatially discretized space, where only the discretized spaces of fixed resolution can be occupied. Off-lattice refers [26] to less defined increments of space, for example 3D localization at floating point precision, often using an agent/individual based approach. That said, hybrid methods are common, for example diffusion is often represented through voxel discretization [16, 22, 35] in otherwise agent-based solutions, providing for fine and even spatial control. Some solutions are entirely on-lattice [9], notably the Cellular Potts method, which was described as an Ising lattice [36], utilizing ‘index-copy’ occurrences via Monte-Carlo Metropolis dynamics method with Boltzmann acceptance [9]. Lattice approaches tend to be more morphologically manipulable due to total discretization, but with inevitable computational costs. Cellular Potts (aka. Glazier-Graner-Hogeweg) multicellular simulators include, perhaps most convincingly, CompuCell3D [9]. Vertex approaches can take on a nodal form in the case of the Finite Element Method, with the discretization of a body into nodes on a mesh to solve complex problems utilizing degrees of freedom. An example using a “subcellular element model” with nodal meshes was the multicellular EmbryoMaker [21] solution, which alluded to an apparently computationally expensive yet high resolution solution with significant morphological flexibility. A hybrid Finite-Element Cellular Potts approach is in VirtualLeaf [37]. Agent-based multicellular simulators, almost always hybridized with a domain-based discretized layer or possibly other modalities, are apparently the most abundant [2, 4, 16, 19, 22, 24–26, 30, 36].

## 3 Methods

### 3.1 Overview

The benchmarking of two novel 3D multicellular simulators with and without NGSS integration on a high-end personal computing system will be described to demonstrate computational limits, reveal enhancements and demonstrate the scalabilities of different



approaches. A specific NGSS version was tailored for Windows and its integration with multicellular layers was pursued to bridge the gap between multicellular simulation and Synthetic Biology CAD design. This methodology would provide insights into CAD considerations regarding simulation architecture, ergonomics, and demonstrated principled *in silico* population level emergence from the algorithms.

SBML-Constructor was a utility developed to automatically generate simple SBML level 2 format biological reaction pathways of differing homogeneity, lengths and topologies for benchmarking with the NGSS stochastic simulator coupled to a multicellular simulation layer. It was developed using SBML level 2 documentation as formatting guidance [38] to overcome interoperability issues [6].

NGSS-Invoker is a simple utility program developed to execute and benchmark an adapted Windows version of NGSS multiple times and hence fully saturate the CPU to measure the time taken for a user defined number of NGSS activations to complete.

### 3.2 UnrealMulticell3D



**Fig. 1.** UnrealMulticell3D circular, raised cellular colony formation of bacillus cells.

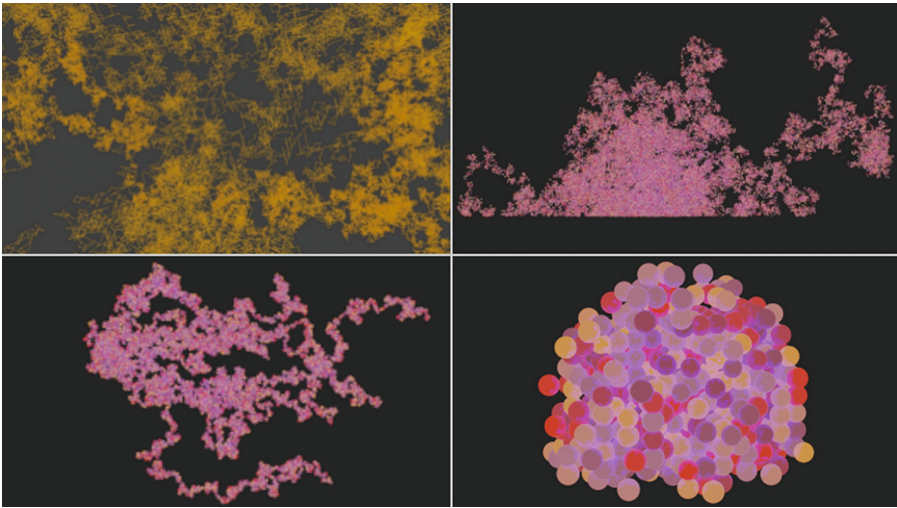
UnrealMulticell3D (UM3D) is a prototype, agent-based, off-lattice, real-time, 3D multicellular simulation software developed in Epic Games' Unreal Engine 4 (UE4) and C++. UM3D addressed inferior graphical solutions [16, 20] with the state of the art 3D UE4 used for blockbuster gaming productions. UE4 provides GPU support for physics calculations utilizing PhysX which works with GeForce GPUs and performs Newtonian physics. UM3D also addressed the lack of real time ergonomic user interfaces and Windows accessibility compared to otherwise very robust methods [22, 26].

Cellular assets were produced through 3D skeletal mesh designs in Blender 2.90.1, including the use of shape-key animations for bacterial cleavage. The basis mesh consisted of 3,551 vertices for a bacillus shaped cell, but a simplified mesh was successfully

trialed using only 8 vertices with only visual implications since physical interactions used a ‘Capsule Component’, inherent to the UE4 ‘Character’ blueprint class. The bacteria in UM3D would split upon reaching a shape-key setting of double the size from 1.5 microns to 3 microns in length, based in the simulation literature [16]. UE4 use has the capacity to overcome limitations in the modelling of morphology and heterogeneity compared to the most promising agent-based multicellular solutions [22, 26]. Also, such solutions lack interdisciplinary ease of use, with the literature recommending the harnessing of graphical user interfaces [11] as exhibited in UM3D.

Figure 1 demonstrates a circular, raised colony that, in this case, took 28.112 s to reach 16,384 simplified mesh cells from a single cell, as measured by the epoch-based timer. This type of colony morphology was used for benchmarking purposes.

### 3.3 SynthMeshBuilder



**Fig. 2.** SynthMeshBuilder’s diverse morphology generation of mesh-based cell networks (upper left), highly scalable one million cell colonies (upper right), parallelized “stochastic chain extension” reminiscent of staphylococcus clusters [39] used for benchmarking (lower left) and an alternative on-lattice algorithm with random update order (lower right), as visualized with Blender.

SynthMeshBuilder (SMB) is a procedural, multithreaded, vertex-based, batch processed, 3D multicellular mesh generation prototype software, developed in .NET C# for Windows, blurring the line between On-Lattice and Off-Lattice approaches, with an agent-based character and utilizing generative rules-based decision making. Mesh generation is practical since vertices can be represented by minimal data and meshes can be used to construct and regulate large objects such as tissues.

SMB shares similarity to a ‘family’ of multicellular tools [22, 24–26]. Commonalities include batch processing, retrospective visualization, spherical agents, proliferation focused, computationally parallelized, highly scalable, independent solutions; although

SMB has not yet utilized HPC. Batch processing can be recommended as it negates the need for live rendering costs, and can provide for interface-free computation if embedded in a suite. Animation of growth was possible in SMB due to the ordered sequence of vertices in the OBJ output file that Blender 2.90.1 could iterate over with particles (Fig. 2).

SMB was trialed with various approaches, including on and off lattice solutions. The “Pseudo Off-Lattice” algorithm used no fixed lattice boundaries but placed vertices at unit distance from the parent and was used for benchmarking. Within this Pseudo Off-Lattice approach, the “Stochastic Chain Extension” ruleset insighted the most recently generated cell on a computational thread to replicate in a random direction into an unoccupied location. “Tunnelling” of cells was used if the cell could find no free space to expand into locally. OBJ encoding of edges occurred between parent and daughter cells to generate the histological mesh. In multicellular scenarios such bonds may be formed by extracellular substances, cell-matrix adhesion and cell-cell adhesion. Such interwoven webs could be used for intercellular communication simulations. High entropy stochastic outcomes can be progressed towards low entropy organization and behavior by increasing algorithmic control. The benchmarked colony formations were morphologically different from those of UM3D due to the underlying algorithmic differences.

### 3.4 NGSS Use via NGSS-Invoker, UnrealMulticell3D and SynthMeshBuilder

No actual model feedback into the simulators was attempted or achieved when running the external NGSS metabolic simulator besides NGSS completion checks via output files. Whether it is through stochastic simulation or another solution such as the use of Boolean networks [29], flux balance analysis [27], deterministic or hybrid algorithms [11] subcellular “decision making” affecting cellular phenotypes is vital in order to elicit regulated, emergent multicellular behavior *in silico*. However, for prototyping and benchmarking purposes it was sufficient to execute the control flow without feedback.

## 4 Results

### 4.1 Hardware, Software and Models Used for Benchmarking

Benchmarking was performed using a G7 7700 Dell Laptop, Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz (6 cores, 12 logical cores) processor, 16.0 GB RAM, NVIDIA GeForce RTX 2060 6 GB VRAM graphics card on Windows 10 64-bit.

SBML-Constructor was used to produce sets of SBML models with serial pathways up to 128 reactions in length with low enzyme and high substrate concentrations to homogenize performance. Two sets of models were generated, a set with separate enzymes (multi-enz) for each reaction and one with a single enzyme (single-enz) mediating every reaction. Most SBML models from the BioModels database were reported as 50 reactions or less, with a few having as many as 1800 reactions [5]. Thus the range generated by SBML-Constructor, up to 128 reactions, could give a reasonable sense of tractability for models from curated model archives. NGSS-Invoker was used to benchmark NGSS without spatial simulation using the SBML models. UM3D and SMB engines were benchmarked with and without NGSS, with spatial and graphical consequences but without actual logical feedback from the SBML model itself.

## 4.2 Benchmarking Without the Multicellular Layer

### Single Cell Performance via NGSS-Invoker

Initially, three different Stochastic Simulation Algorithms (SSAs) were tested. A fourth was added retrospectively on recommendation by the SSAPredict tool [5]. With the multienzyme model, SSAPredict concluded for the 4, 8, 32, 64 and 128 reaction models that the Logarithmic Direct Method should be optimal. For 2 and 3 pathway reactions the Optimized Direct Method was recommended. Nevertheless, Tau Leaping performed faster, likely because it favors low propensity (slow, low probability) scenarios [12] matching the models generated. For all algorithms, time to complete increased with a polynomial trend as the number of reaction steps to complete increased. Up to 6 reactions the Tau Leaping algorithm possessed a nuance where it followed the relatively slow Direct Method.

A comparison of the behavior of different SSA algorithms was also performed for the Single Enzyme model set, where NGSS performed identically in every case despite SSAPredict recommending the Logarithmic Direct Method (LDM) and the Partial Propensity Direct (PPD) Method. The single-enz topology was far slower than multi-enz. Out of the conditions tested, NGSS turnover was deemed fastest under Tau Leaping, interval 0.1, MAX\_TIME 3, with multiple enzyme models over a single run.

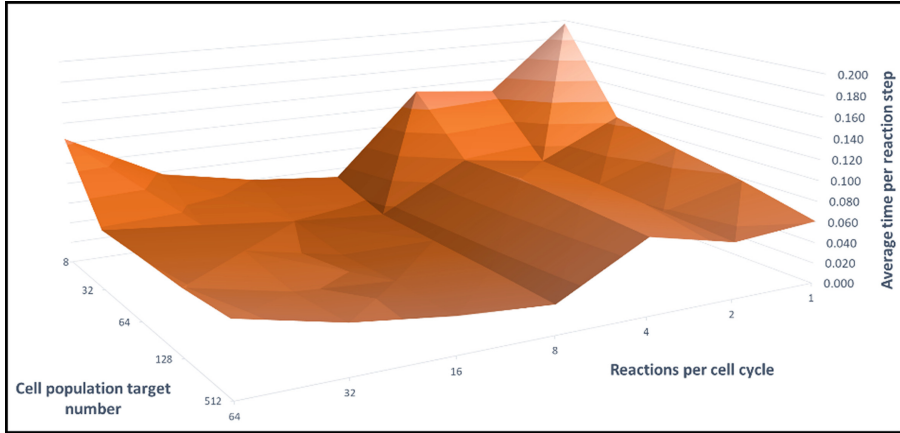
### Multicellular Performance via NGSS-Invoker

The fastest performing settings were brought forward and NGSS-Invoker was used to saturate the CPU with multiple concurrent NGSS process activations. Because only a single run was being made, the ‘parallel on’ NGSS setting served no purpose, but behaved differently from the ‘parallel off’ setting. Note that the NGSS parallel thread setting is independent from the parallelization by multiple NGSS activations and is used to average multiple stochastic runs. As the “cell target” (NGSS completions) increased, initial performance increase towards plateau was due to the concurrency of NGSS activations on the processor, reducing time per cell. The conclusion that was brought forward was that for a single run, the ‘parallel off’ NGSS setting was the less time consuming algorithm.

## 4.3 Benchmarking with the Multicellular Layer

**Multicellular Performance via UnrealMulticell3D with NGSS** Using the conditions established from the previous experiments, NGSS could be run in the spatial multicellular simulators once per mitotic cell cycle and the reaction count could be varied by changing the multi-enz SBML model. Here we discuss this implementation into UM3D. The three variables (time, population, reactions) resulted in 3D statistical data (Fig. 3). Starting from a single cell, the duration to reach a given population size was longer for larger cell target populations and as the number of reactions per cell cycle was increased. Polynomial time scaling with reactions per cycle induced by the NGSS algorithm was likely because the reactions in the model were not mutually exclusive and, hence, had computational interference, matching NGSS behavior. The optimum number of reactions in the model was 8 due to the nuances of the Tau Leaping algorithm and the range of models tested (note that 7 reactions was not tested). On the other hand, scaling

towards a target population was essentially linear given constant reactions per cycle. A linear scaling was unsurprising as increasing the cell population target simply increased the number of repetitions of the same action, especially once processor saturation was reached.

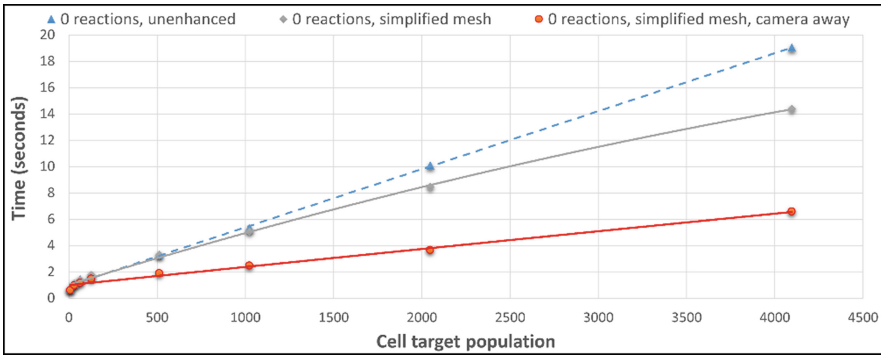


**Fig. 3.** Benchmarked time performance data subset of UnrealMulticell3D on a time per reaction step basis. Note that greater cell target populations ensured processor saturation, explaining the peaks, with performance consistency upon saturation given unchanging reactions per cell cycle (RPC). There was polynomial scaling with RPC beyond the NGSS nuance up to  $\sim 7$  RPC.

**Multicellular Performance via UnrealMulticell3D Without NGSS** Starting from a single cell, target populations were reached over a measured time without NGSS processing to evaluate the behavior of the multicellular simulation layer alone. Physics was compared in the on and off states with increasing model sizes. Both cases performed undiscernibly, implying that physics computations played little part in overall performance. This led to the recognition that Unreal Engine 4 was using PhysX to calculate physics on the GPU. Thus toggling physics had no statistical impact because the CPU was equally saturated whilst the GPU apparently remained unsaturated. A benchmarking effort was also made to assess other factors such as the impact of cell textures, with no discernible computational costs. That said, the performance enhancement from turning the camera away from the cells was dramatic, demonstrating a significant slowdown associated with rendering costs. By eliminating the need to graphically render, the GPU could compute physics with reduced graphical responsibilities. In the future the camera might be completely turned off in a batch processing mode accompanied by the export of simulation states. The removal of an aesthetic animation shape-key for the bacterial cleavage site had no impact on the simulation time, but simplifying the bacterial mesh from 3,551 vertices down to 8 vertices significantly sped up the simulation, particularly at higher cell populations.

The impact of the collective performance enhancements were compared to the conditions prior to benchmarking (Fig. 4). Regardless of these significant baseline engine

enhancements, use with NGSS had very limited to no benefit since NGSS was the limiting factor in the total simulation time, justifying the future pursuit of HPC.



**Fig. 4.** Performance enhancement of UnrealMulticell3D without NGSS, due primarily to a move towards batch-processing and reduced rendering costs.

### Multicellular Performance via SynthMeshBuilder Without NGSS

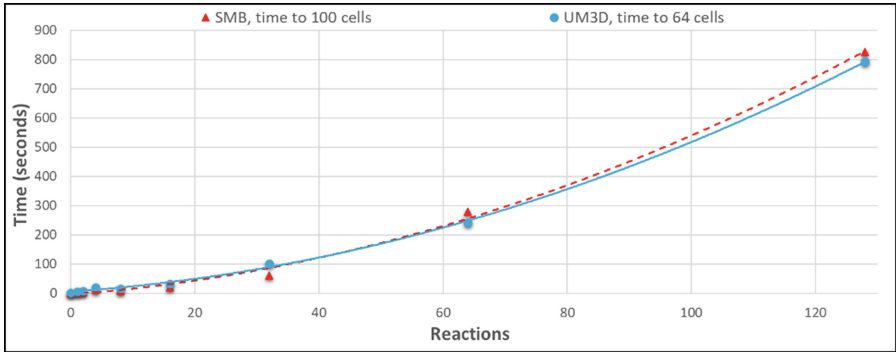
The first benchmarking experiment with SMB targeted 252,000 cells. The number of threads ranged from 1 to 50, with cells per thread ranging from 252,000 to 5,040 respectively. Performance plateaued as threads reached the number of logical cores (12) beyond which there was a very slight decrease in performance. 10 threads were brought forward for benchmarking to maintain a responsive UI and operating system. Linear scaling was not achieved as the number of cells increased, rather there was a polynomial increase in simulation time. This is almost certainly because as the cell population grew, the overlap checks on proliferation also grew in number since all cell coordinates were iterated over. Thus many unnecessary points were scanned as the point-cloud developed. This is where domain-based parallelization or nearest neighbor lists [16, 22] could be considered, with a reduction of cell by cell processing and thereby the linearization of the trend. An On-Lattice approach with local scanning across the restricted lattice geometry is one option, but Off-Lattice provides for more diverse spatial potential going forward. Alternatively, distinct cell populations could be computed on separate processors within a heterogenous pool of cells. The probabilistic “bridge” concept between physically isolated populations might also be considered [19].

### SynthMeshBuilder Versus UnrealMulticell3D Performances with NGSS

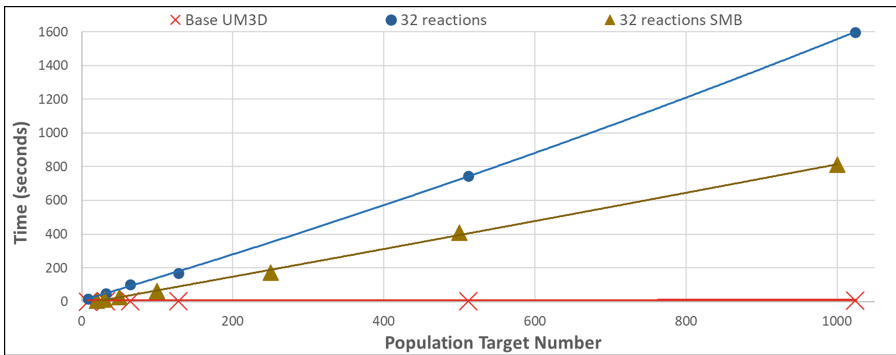
SMB scaled in a similarly polynomial fashion but performed faster than UM3D when NGSS was processing models (Fig. 5). This should be attributable to the fact that SMB is algorithmically far less complex than UM3D and was able to leave the majority of the CPU for NGSS to utilize and was entirely batch-processed, thereby circumventing live rendering costs. The scaling followed NGSS behavior for the network sizes. With NGSS, SMB was able to perform almost twice as quickly as UM3D (evidenced in both Figs. 5 and 6) due to its simpler ground-up algorithmics, specifically streamlined for



NGSS performance. The performances of the multicellular engines with and without NGSS were very different with even modest SBML model conditions imposed, with NGSS lengthening the simulation time dramatically (Fig. 6).



**Fig. 5.** Both coupled with NGSS, SynthMeshBuilder scaled faster than UnrealMulticell3D due to its streamlined algorithmics, but with a similar trend. The target populations differ above.

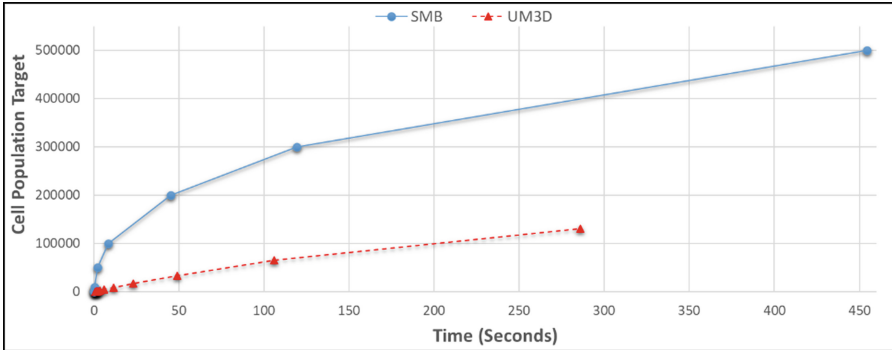


**Fig. 6.** The base multicellular simulation layers could generate 1000-cell populations within seconds (UM3D) or fractions of a second (SMB). However adding a moderately sized reaction network (32 reaction steps) with parallel NGSS processes (one for each cell cycle) resulted in a drastically more time consuming performance profile operating on the order of several minutes to complete. Because NGSS could use as much as a single core (two threads) of processing power for each activation, by the time only 6 cells was reached the processor could be saturated, giving an overall linear scaling as NGSS processes were queued for completion on the rapidly saturated single processor. The same linear scaling would be expected with HPC but with a shallower gradient, at least once the HPC cores were fully saturated.

**SynthMeshBuilder Versus UnrealMulticell3D Scalabilities Without NGSS**

The scalabilities of SMB to UM3D without NGSS were compared (Fig. 7). SMB demonstrated far greater scalability (with only 10 of 12 threads) compared to UM3D (with no imposed resource restrictions). UM3D was slower and more unstable, eventually reaching a respectable 131,072 cells. SMB was stopped at 500,000 cells, although it can scale

much further. That said, a high cell count is not required for multicellular life. The adult nematode worm *Caenorhabditis Elegans*, an oft used model organism, has been reported to have as few as 959 cells [40]. However, should regulatory computations (subcellular models) and more multicellular simulation phenomena be added (e.g. diffusion, extracellular agents), scalabilities would become much lower. On the other side of the spectrum, the human retina photoreceptor topography has been reported as being composed of as many as 5.29 million cone and 107.3 million rod cells [41]. Far beyond tractability on conventional hardware without simplification.



**Fig. 7.** SynthMeshBuilder proved much more scalable without NGSS than UnrealMulticell3D on the personal computing system. By contrast, HPC solutions from the literature could process millions [26] or even tens of millions [22, 28] of cells, with thousands [20] or hundreds of thousands [26] reported on modest hardware.

## 5 Conclusions

Multiple ground-up approaches to agent-based multicellular simulation were demonstrated; a scalable, prototype, mesh-based, batch processed approach (SMB) and a state-of-the-art 3D engine approach (UM3D). SMB demonstrated that low level abstractions can have scalable yet compelling outcomes reminiscent of classical Cellular Automata, with a reduction of entropic behavior achieved by increasing algorithmic regulatory control. Both SMB and UM3D can benefit from many additional multicellular features that are described in the literature and obtainable through open-source code. A critical progression for SMB and UM3D is subcellular model feedback with phenotypic effects, with the choice of subcellular models also of critical importance. Subcellular regulatory models would either be designed or downloaded from a repository.

The temporal use of UE4 physics would need to be carefully considered in order to make multiscale performance accurate, along with the overall careful orchestration of temporality in general. On the other hand, Unreal Engine ensured that physics and other computations could be performed on the GPU but if live rendering is occurring for large numbers of cells, there could be a negative impact on performance. Thus the results alluded strongly towards batch processing methodologies with rendering minimized,



such as provided by SMB that harbors similarity to various extant tools [22, 24–26]. UM3D could be adapted towards batch processing away from real time rendering but while retaining the visualization options.

NGSS saturation of the processor via process executions demonstrated the limits of a high end desktop computer. Temporal multiscale implications were observed as NGSS’s significant temporal usage contrasted with cell growth, physics and population growth dynamics in the multicellular layer, particularly as seen in UM3D. NGSS has a peculiar performance nuance at fewer than 7 reaction models with the Tau Leaping algorithm, however many reaction networks will likely be larger than 6 reactions. For NGSS, metabolic network topology has a significant impact on performance, as demonstrated by the Single Enzyme models versus the Multienzyme models.

Subsequent work should challenge the limits of multiscale, multicellular simulations including the implementation of novel case studies of morphogenetic and functional multicellularity/histology, including subcellular model feedback, heterogenous populations and the continued hybridization of both agent-based and lattice-based (domain-based) modelling. The importance of having identified the key features of multicellular simulation should not be underestimated for subsequent work, specifically when considering the integration of, what has been described as, ‘sub-models’ [26].

The current work highlighted just how computationally intensive cell by cell computations are, especially with the use of subcellular biochemical stochastic simulations, and implicated the future use of HPC that might be applied on a cell by cell basis with the splitting of NGSS activations between processors [28] (see Fig. 6). The ‘first-come-first-serve’ consequences of NGSS activation across processors potentially skews realism, rectifiable primarily by HPC distribution of NGSS and/or a possible Monte Carlo approach. The use of clustering into “super-individuals” [22] can be considered, along with the possibility of population-based or hybrid individual/population approaches.

Model verification would need to be considered perhaps through mechanisms such as micrographic analysis [10], machine learning [3] and/or with further insights garnered from extant projects [28]. Once convincing simulations are operational at a small scale with verification protocols, the simulations can then be scaled up using HPC. The use of HPC is planned for upcoming work.

**Acknowledgements.** The work presented in this paper was supported by EPSRC research grant EP/R043787/1.

## References

1. Chandran, D., Bergmann, F.T., Sauro, H.M.: Computer-aided design of biological circuits using tinkercell. *Bioeng. Bugs* **1**(4), 276–283 (2010)
2. Sütterlin, T., et al.: A 3D self-organizing multicellular epidermis model of barrier formation and hydration with realistic cell morphology based on EPISIM. *Sci. Rep.* **7**(1), 43472 (2017)
3. Preen, R.J., Bull, L., Adamatzky, A.: Towards an evolvable cancer treatment simulator. *Biosystems* **182**, 1–7 (2019)
4. Mirams, G.R., et al.: Chaste: an open source C++ library for computational physiology and biology. *PLoS Comput. Biol.* **9**(3), e1002970 (2013)

5. Sanassy, D., Widera, P., Krasnogor, N.: Meta-stochastic simulation of biochemical models for systems and synthetic biology. *ACS Synth. Biol.* **4**(1), 39–47 (2015)
6. Konur, S., et al.: Toward full-stack in silico synthetic biology: integrating model specification, simulation, verification, and biological compilation. *ACS Synth. Biol.* **10**(8), 1931–1945 (2021)
7. Keating, S.M., et al.: SBML level 3: an extensible format for the exchange and reuse of biological models. *Mol. Syst. Biol.* **16**(8), 1–21 (2020)
8. Watanabe, L., et al.: iBioSim 3: a tool for model-based genetic circuit design. *ACS Synth. Biol.* **8**(7), 1560–1563 (2019)
9. Swat, M.H., et al.: Multi-cell simulations of development and disease using the CompuCell3D simulation environment. *Methods Mol. Biol. (Clifton, N.J.)* **500**, 361–428 (2009)
10. Sütterlin, T., et al.: Bridging the scales: semantic integration of quantitative SBML in graphical multi-cellular models and simulations with EPISIM and COPASI. *Bioinformatics (Oxford, England)* **29**(2), 223–229 (2013)
11. Hoops, S., et al.: COPASI—a complex pathway simulator. *Bioinformatics* **22**(24), 3067–3074 (2006)
12. Sanassy, D., et al.: Modelling and stochastic simulation of synthetic biological boolean gates. In: 2014 IEEE Intl Conf on High Performance Computing and Communications, 2014 IEEE 6th Intl Symp on Cyberspace Safety and Security, 2014 IEEE 11th Intl Conf on Embedded Software and Syst (HPCC, CSS, ICSS) (2014)
13. Karagöz, Z., et al.: Towards understanding the messengers of extracellular space: computational models of outside-in integrin reaction networks. *Comput. Struct. Biotechnol. J.* **19**, 303–314 (2021)
14. Okuda, S., Inoue, Y., Adachi, T.: Three-dimensional vertex model for simulating multicellular morphogenesis. *Biophys. Physicobiol.* **12**, 13–20 (2015)
15. Matyjaszkiewicz, A., et al.: BSim 2.0: an advanced agent-based cell simulator. *ACS Synth. Biol.* **6**(10), 1969–1972 (2017). <https://doi.org/10.1021/acssynbio.7b00121>
16. Naylor, J., et al.: Simbiotics: a multiscale integrative platform for 3D modeling of bacterial populations. *ACS Synth. Biol.* **6**(7), 1194–1210 (2017)
17. Belmonte, J.M., et al.: Virtual-tissue computer simulations define the roles of cell adhesion and proliferation in the onset of kidney cystic disease. *Mol. Biol. Cell* **27**(22), 3673–3685 (2016)
18. Butler, M.T., Wallingford, J.B.: Planar cell polarity in development and disease. *Nat. Rev. Mol. Cell Biol.* **18**(6), 375–388 (2017)
19. Liberman, A., et al.: Cell studio: a platform for interactive, 3D graphical simulation of immunological processes. *APL Bioeng.* **2**(2), 026107 (2018)
20. Gutiérrez, M., Gregorio-Godoy, P., Pérez, G., del Pulgar, L.E., Muñoz, S.S., Rodríguez-Patón, A.: A new improved and extended version of the multicell bacterial simulator gro. *ACS Synth. Biol.* **6**(8), 1496–1508 (2017). <https://doi.org/10.1021/acssynbio.7b00003>
21. Marin-Riera, M., et al.: Computational modeling of development by epithelia, mesenchyme and their interactions: a unified model. *Bioinformatics (Oxford, England)* **32**(2), 219–225 (2016)
22. Li, B., et al.: NUFEB: A massively parallel simulator for individual-based modelling of microbial communities. *PLoS Comput. Biol.* **15**(12), e1007125 (2019)
23. Hellweger, F.L., et al.: Advancing microbial sciences by individual-based modelling. *Nat. Rev. Microbiol.* **14**(7), 461–471 (2016)
24. Kang, S., et al.: Biocellion: accelerating computer simulation of multicellular biological system models. *Bioinformatics (Oxford, England)* **30**(21), 3101–3108 (2014)
25. Lardon, L.A., et al.: iDynoMiCS: next-generation individual-based modelling of biofilms. *Environ. Microbiol.* **13**(9), 2416–2434 (2011)

26. Ghaffarizadeh, A., et al.: PhysiCell: an open source physics-based cell simulator for 3-D multicellular systems. *PLoS Comput. Biol.* **14**(2), e1005991 (2018)
27. Karimian, E., Motamedian, E.: ACBM: an integrated agent and constraint based modeling framework for simulation of microbial communities. *Sci. Rep.* **10**(1), 8695 (2020)
28. Markram, H.: The blue brain project. *Nat. Rev. Neurosci.* **7**(2), 153–160 (2006)
29. Rubinacci, S., et al.: CoGNAC: a chaste plugin for the multiscale simulation of gene regulatory networks driving the spatial dynamics of tissues and cancer. *Cancer Inform.* **2015**(Suppl. 4), 53–65 (2015)
30. Bloch, N., et al.: An interactive tool for animating biology, and its use in spatial and temporal modeling of a cancerous tumor and its microenvironment. *PLoS ONE* **10**(7), e0133484 (2015)
31. Eberl, H., et al.: *Mathematical Modeling of Biofilms*, vol. 18. IWA Publishing, London (2006)
32. Lee, C.T., et al.: 3D mesh processing using GAMer 2 to enable reaction-diffusion simulations in realistic cellular geometries. *PLoS Comput. Biol.* **16**(4), e1007756 (2020)
33. Murphy, R.F.: (3) The CellOrganizer project: an open source system to learn image-derived models of subcellular organization over time and space. *IEEE* (2012)
34. Murphy, R.F.: Building cell models and simulations from microscope images. *Methods (San Diego, Calif.)* **96**, 33–39 (2016)
35. Ghaffarizadeh, A., Friedman, S.H., Macklin, P.: BioFVM: an efficient, parallelized diffusive transport solver for 3-D biological simulations. *Bioinformatics (Oxford, England)* **32**(8), 1256–1258 (2016)
36. Delile, J., et al.: A cell-based computational model of early embryogenesis coupling mechanical behaviour and gene regulation. *Nat. Commun.* **8**(1), 13929 (2017)
37. Roeland, M.H.M., et al.: VirtualLeaf: an open-source framework for cell-based modeling of plant tissue growth and development1[C][W][OA]. *Plant Physiol. (Bethesda)* **155**(2), 656–666 (2011)
38. Hucka, M., et al.: Systems biology markup language (SBML) level 2: structures and facilities for model definitions. *Nat. Prec.* (2007)
39. Taj, Y., et al.: Study on biofilm-forming properties of clinical isolates of staphylococcus aureus. *J. Infect. Dev. Ctries.* **6**(5), 403–409 (2012)
40. Sulston, J.E., et al.: The embryonic cell lineage of the nematode caenorhabditis elegans. *Dev. Biol.* **100**(1), 64–119 (1983)
41. Curcio, C.A., et al.: Human photoreceptor topography. *J. Comp. Neurol.* **292**(4), 497–523 (1990)



# Integrating *in-vivo* Data in CFD Simulations and in *in-vitro* Experiments of the Hemodynamic in Healthy and Pathologic Thoracic Aorta

Alessandro Mariotti<sup>1</sup>(✉), Emanuele Gasparotti<sup>2</sup>, Emanuele Vignali<sup>2</sup>,  
Pietro Marchese<sup>2,3</sup>, Simona Celi<sup>2</sup>, and Maria Vittoria Salvetti<sup>1</sup>

<sup>1</sup> Dipartimento di Ingegneria Civile ed Industriale, University of Pisa, Pisa, Italy  
alessandro.mariotti@unipi.it, mv.salvetti@ing.unipi.it

<sup>2</sup> BioCardioLab - Heart Hospital, Fondazione Toscana G. Monasterio, Massa, Italy  
{emanuele.gasparotti, evignali, s.celi}@ftgm.it

<sup>3</sup> Institute of Life Sciences, Scuola Superiore Sant'Anna, Pisa, Italy  
pietro.marchese@santannapisa.it

**Abstract.** A comparison between the results obtained integrating *in-vivo* measurements in numerical simulations and *in-vitro* experiments is presented. Three aorta geometries are considered: a patient-specific healthy aorta, an aneurysmal aorta, and a coarctated aorta, both derived from the former geometry. Hemodynamic simulations are carried out by using the open-source code Simvascular. *In-vitro* data are obtained by a fully controlled and sensorized circulatory mock loop for 3D-printed aortic models. This experimental setup allows the elimination of a few uncertainties conversely present in *in-vivo* data: the flow rate is controlled and the same waveform is present in each cardiac cycle, the model is fixed, and the wall model properties are known. In this way, clearer indications can be obtained to assess and possibly to improve the accuracy of CFD models. The comparison between CFD and *in-vitro* data is excellent for all the considered cases. The agreement with *in-vivo* data is satisfactory and consistent with the possible controlled and uncontrolled differences with the numerical and *in-vitro* set-up. The validated CFD and *in-vitro* platforms are then used to investigate in detail the hemodynamics and to point out, in particular, the differences between the healthy and pathological cases.

**Keywords:** Hemodynamic simulations · CFD · Ascending thoracic aorta · *in-vivo* measurements · CFD vs. *in-vitro* results

## 1 Introduction

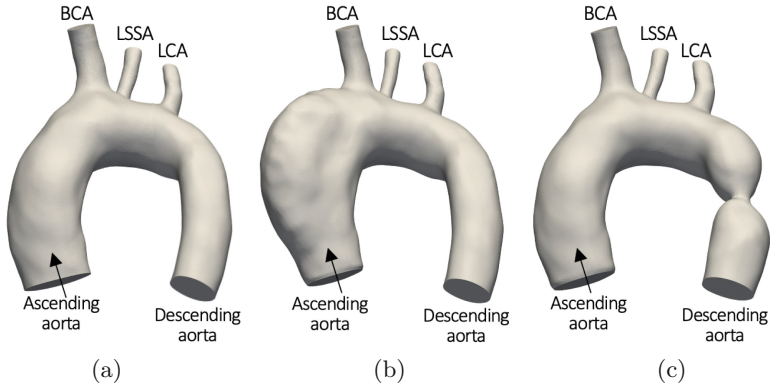
Hemodynamic forces play an important role in the initiation and progression of cardiovascular diseases. In the last years, the merging of Computational Fluid Dynamics (CFD) with Magnetic Resonance Imaging (MRI) has been used to

provide clinical information at a patient-specific level. *In-vivo* MRI data can be successfully used to obtain patient-specific boundary conditions for the simulations (see e.g. [1–5]), as well as for comparison against numerical results, providing cross-validation. However, MRI suffers from spatial and temporal resolution limitations and it is not able to provide quantitative flow descriptors, such as wall shear stresses, with sufficient accuracy. On the other hand, CFD allows the flow and pressure fields to be investigated with space and time resolutions that are not achievable by any *in-vivo* measurement. The accurate morphological features and the non-invasive quantification of blood flow [6–8] provided by *in-vivo* measurements have been combined with CFD to investigate hemodynamics on a patient-specific basis, in both healthy [1–3, 5, 12, 13] and diseased subjects [9–11, 14–17]. Nevertheless, the accuracy of CFD predictions strongly depends not only on the need for accurate MRI data but also on modeling assumptions and computational set-up. Different sources of uncertainties are indeed present in CFD models, e.g., inlet flow rate, outflow pressure waveform, and arterial stiffness for fluid-structure interaction, and these uncertainties may affect the accuracy of the output quantities of interest (see e.g. [9, 11, 17–22]).

A comparison between numerical simulations and *in-vitro* data is presented in this paper. Three aorta geometries are considered, viz. the healthy aorta from [5], an aneurysmal aorta and a coarctated aorta, both derived from the former geometry. Phase-contrast MRI is used to provide *in-vivo* temporally-resolved velocity data for the inlet conditions in simulations and experiments. Simulations are carried out with the open-source code Simvascular [23], whereas *in-vitro* data are obtained by a fully controlled and sensorized circulatory mock loop for 3D-printed aortic models [24]. This experimental setup allows the elimination of given uncertainties that are conversely present *in-vivo* data: the flow rate is perfectly controlled in each cardiac cycle, the model is fixed, and the wall model properties are known. In this way, clearer indications can be obtained to assess and possibly improve the accuracy of CFD models. Results are compared in terms of velocity and pressure waveforms at the outlet sections and of velocity fields in different portions of the aorta, which are obtained through echography in experiments.

## 2 Problem Definition, Numerical Methodology and Experimental Set-Up

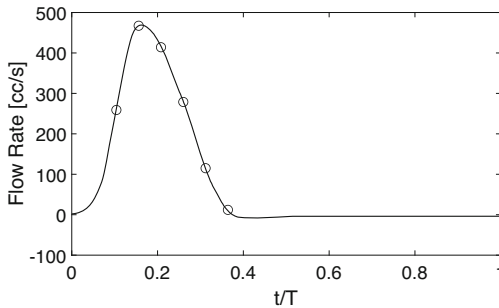
The geometry of the healthy thoracic ascending aorta, shown in Fig. 1a, is obtained from MRI acquisitions performed by means of a 3T MR-scanner on one healthy subject (28 years, male) with a tricuspid aortic valve ([5]). Anatomical and functional data are extracted from MRI. From this geometry, two additional ones reproducing an aneurysmal and a coarctated aorta are obtained by means of computer aided design tools (see Fig. 1b and Fig. 1c, respectively). The flow rate at the inlet section of the considered aortas as it was directly evaluated from *in-vivo* functional MRI data is reported in Fig. 2. The MRI dataset volume is retrospectively reconstructed with the Phase-Contrast Magnetic Resonance Angiography (PC-MRA) technique.



**Fig. 1.** Sketch of the considered geometries: (a) healthy aorta, (b) aortic aneurysm, and (c) coarctation of the aorta.

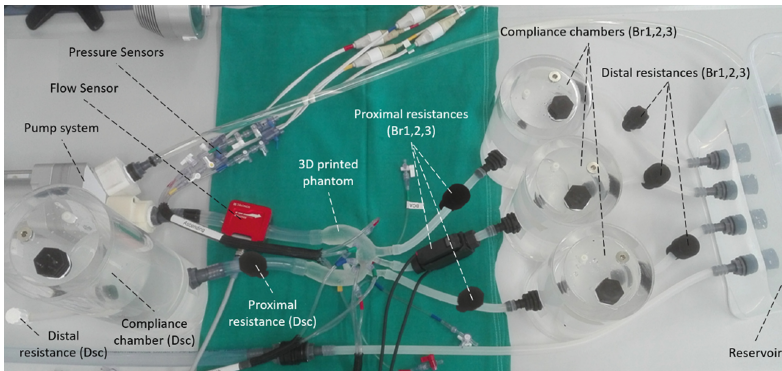
In the numerical simulations, blood is considered as a Newtonian and incompressible fluid with density and kinematic viscosity equal to  $\rho = 1.06 \text{ g cm}^{-3}$  and  $\nu = 3.77 \times 10^{-2} \text{ cm}^2 \text{ s}^{-1}$ . The three-dimensional Navier-Stokes equations for incompressible flows are thus considered as governing equations. At the inlet section of the computational domain we imposed plug flow with the measured flow-rate waveform, while at the outflow boundaries, we used the 3-element Windkessel model. Since the experimental model is rigid, rigid-wall simulations are considered for comparison. The open-source code *SimVascular* is used to carry out the hemodynamic simulations. A finite-element method, including SUPG/PSPG stabilizing terms, is used to discretize the governing equations. The stabilized formulation allows to choose P1-P1 elements, i.e. linear shape functions for both velocity and pressure.

For the *in-vitro* experiments, the mock-circulatory loop setup described in [24] is used. The active component of the setup is given by a custom speed-



**Fig. 2.** Inlet flow-rate waveform for the deterministic comparison with *in-vivo* and *in-silico* data

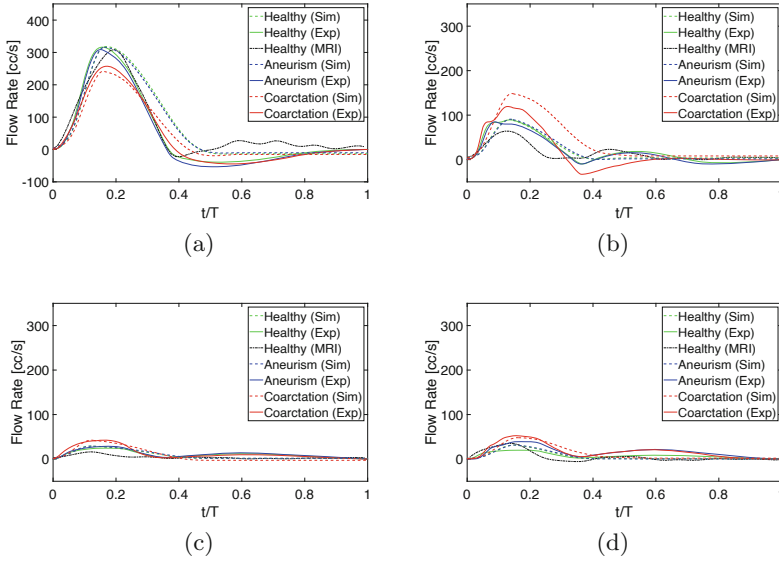
controlled piston pump. The pump speed profile is configured through an automatic interpolation process of the given patient-specific flow waveform shown in Fig. 2. 3D-printed models of the three aortas are realized and velocity and pressure measurements are carried out. The physiologic pressure conditions were controlled by three-element Windkessel models. The resistive and capacitive components were modeled with pinch valves and air-filled rigid chambers respectively. Flow/pressure measurements were monitored through dedicated clamp-on ultrasound sensors (Sonotec) and strain-gauge transducers (TruWave) at each outlet branch, whereas flow fields are obtained through echography. For all the tests, the chosen fluid was a mixture of water and glycerol (60/40%) to reproduce the density ( $1060 \text{ kg/m}^3$ ) and the viscosity ( $3.6 \cdot 10^{-3} \text{ Pa}\cdot\text{s}$ ) of the blood. In addition, echocardiographic acquisitions were carried out for the experimental set-up. In particular, Vortex Flow Analysis and color Doppler acquisition procedures were performed on the three aortic phantoms to measure the velocity distributions inside the vessels (Fig. 3).



**Fig. 3.** Picture of the circulatory mock loop.

### 3 Results and Discussion

The flow-rate profiles obtained in the numerical simulations are compared in Fig. 4 with the *in-vitro* results from the circulatory mock-loop. The results for the three geometries of the aorta are presented, together with *in-vivo* MRI data for the healthy aorta. In particular, the flow rate at the outlet section of the descending aorta is shown in Fig. 4a, whereas the ones for the three branches are reported in Fig. 4b,c,d. The agreement between the experiments and the simulations is successful in terms of flow rate, with acceptable errors at systolic peaks. A perfect agreement in the time of the *in-vitro* and numerical peaks of the flow rate in the descending aorta section is found. The agreement with *in-vivo* data from the healthy case is satisfactory and consistent with the possible controlled and uncontrolled differences with the numerical and *in-vitro* set-ups.



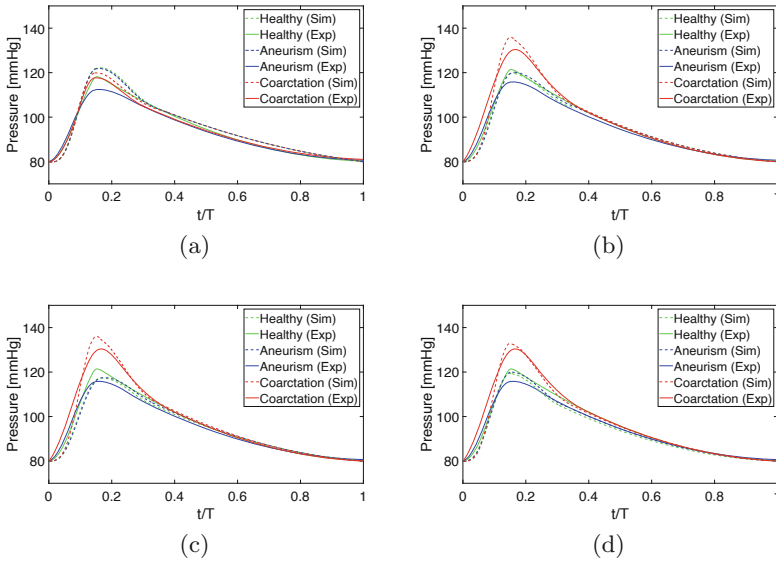
**Fig. 4.** Flow-rate waveforms for *in-vivo*, *in-vitro* and numerical data at the outlet sections: (a) descending aorta, (b) BCA, (c) LCCA, and (d) LSA.

The time lag at the systolic peak between simulated and *in-vitro* results and MRI data is reasonably related to the wall compliance of the real aorta and it is consistent with the findings in [5] in which the time delay between simulations and MRI data reduces by increasing wall compliance.

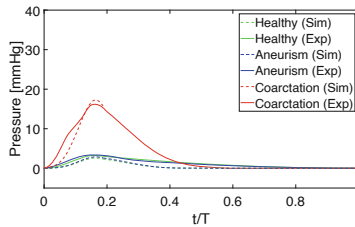
The pressure waveforms at the four outlet sections are reported in Fig. 5. It is worth noting that the physiological range was maintained in both numerical and *in-vitro* environments. Again, the comparison between CFD and *in-vitro* data is excellent for all the considered cases. The differential pressure waveforms between the ascending and the descending aorta are reported in Fig. 6 for the three aortas. As can be expected, the highest differential pressure is found for the coarctated aorta, due to the restricted section. Again, a satisfactory agreement between *in-vitro* and numerical results is found.

Once validated the flow and pressure waveforms, we consider the velocity fields in the three geometries. Velocity distributions in the cross-sections of the healthy aorta, of the aneurismatic aorta and of the coarctated aorta are shown in Fig. 7, Fig. 8 and Fig. 9, respectively. Six different times during a cardiac cycle of time-length  $T$  are considered (the time instants are highlighted with symbols in Fig. 2). Compared with the healthy aorta (Fig. 7), low-velocity regions and local recirculations are present in the aneurysms (Fig. 8) at the systolic peak ( $t/T = 0.156$ ). This low-velocity behavior is actually expected for the aneurysmatic condition. Negligible differences are present in the descending aorta. As for the coarctated aorta (Fig. 9), the main differences with the healthy aorta are found at section S4, with significantly higher velocity values.



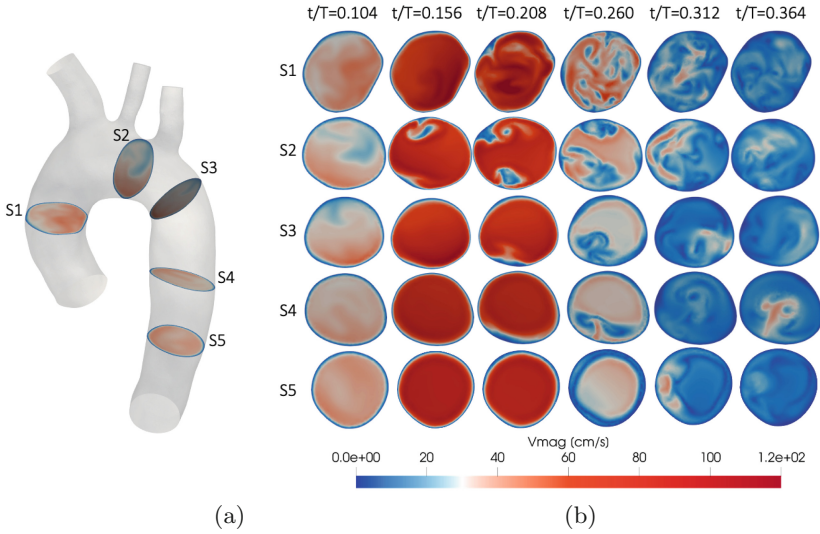


**Fig. 5.** Pressure waveforms for *in-vitro* and numerical data at the outlet sections: (a) descending aorta, (b) BCA, (c) LCCA, and (d) LSA.

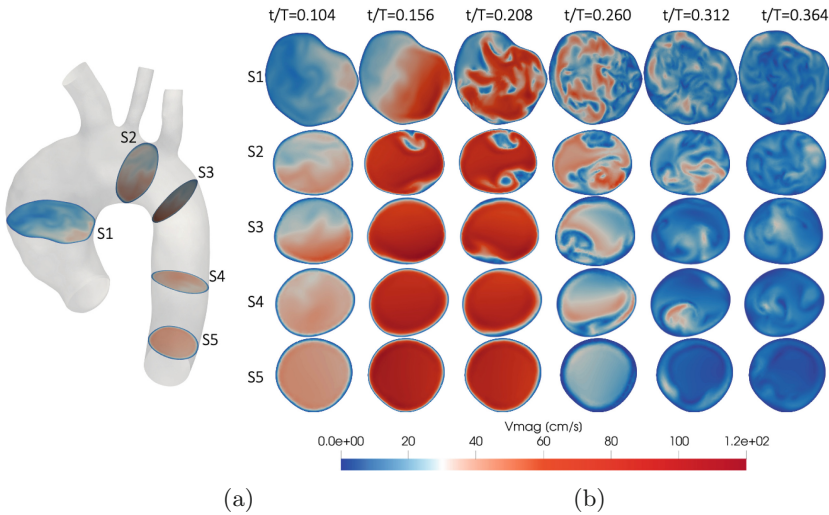


**Fig. 6.** Differential pressure between the ascending and the descending aorta. Comparison between *in-vitro* and numerical data.

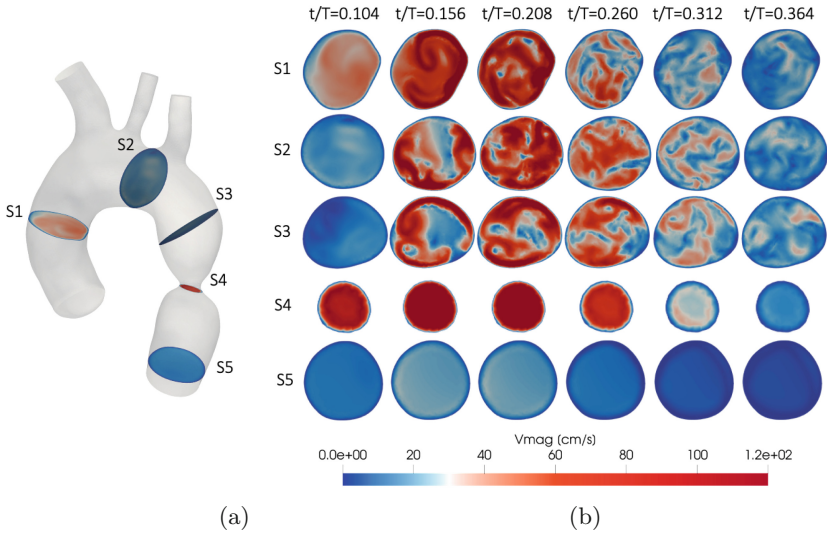
Velocity distribution and streamlines in the sagittal plane through the aorta are then compared with the echography results in the ascending and descending aorta tested in the circulatory mock loop (see Figs. 10, 11, 12 and 13). The velocity fields obtained through CFD are in good agreement with the echography results in *in-vitro* experiments and these fields allowed us to point out the differences between the healthy and pathological cases. In particular, recirculations are evident only in the aneurismatic region at the systolic peak (Fig. 10b at  $t/T = 0.156$ ) and at the early diastole (Fig. 11b at  $t/T = 0.208$ ). The trend is also shown in the numerical results in Fig. 8. On the contrary, blood recirculations are present in all the geometries during the diastole (Fig. 12 at  $t/T = 0.260$ ). Finally, the flow pattern through the coarctation is shown in Fig. 13. A qualitative agreement between simulations and echography patterns is found with an



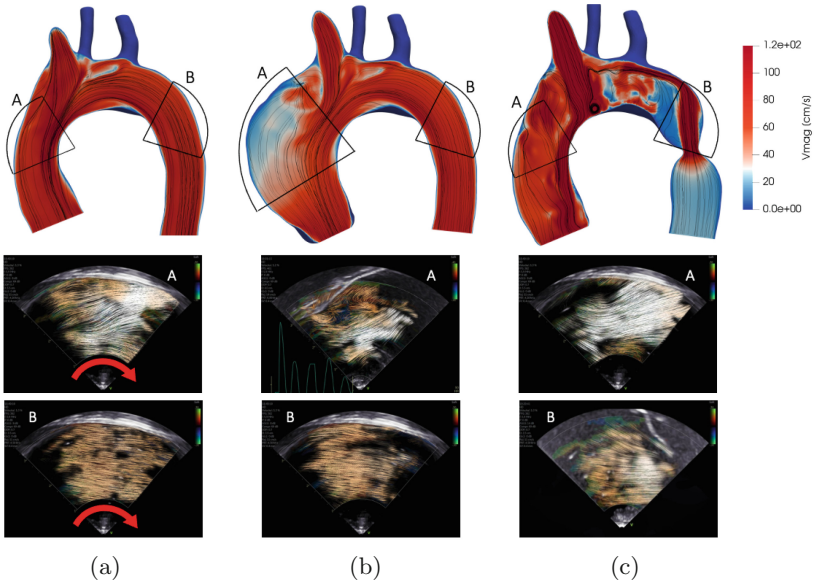
**Fig. 7.** (a) Position of the considered cross-sections in the healthy aorta. (b) Velocity distribution in the cross-sections of the healthy aorta in numerical simulations at different times (from left to right):  $t/T = 0.104$ ,  $t/T = 0.156$ ,  $t/T = 0.208$ ,  $t/T = 0.260$ ,  $t/T = 0.312$ ,  $t/T = 0.364$ .



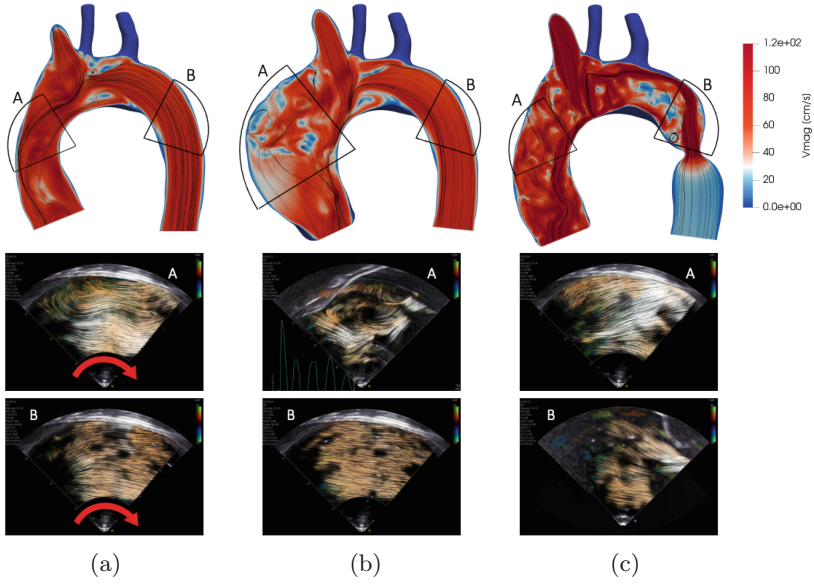
**Fig. 8.** (a) Position of the considered cross-sections in the aortic aneurysm. (b) Velocity distribution in the cross-sections of the aortic aneurysm in numerical simulations at different times (from left to right):  $t/T = 0.104$ ,  $t/T = 0.156$ ,  $t/T = 0.208$ ,  $t/T = 0.260$ ,  $t/T = 0.312$ ,  $t/T = 0.364$ .



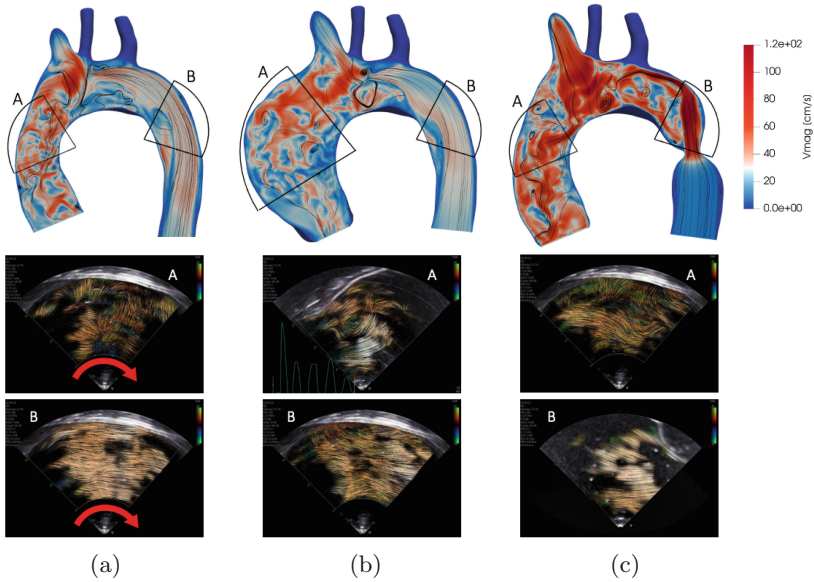
**Fig. 9.** (a) Position of the considered cross-sections in the coarctation of the aorta. (b) Velocity distribution in the cross-sections of the coarctation of the aorta in numerical simulations at different times (from left to right):  $t/T = 0.104$ ,  $t/T = 0.156$ ,  $t/T = 0.208$ ,  $t/T = 0.260$ ,  $t/T = 0.312$ ,  $t/T = 0.364$ .



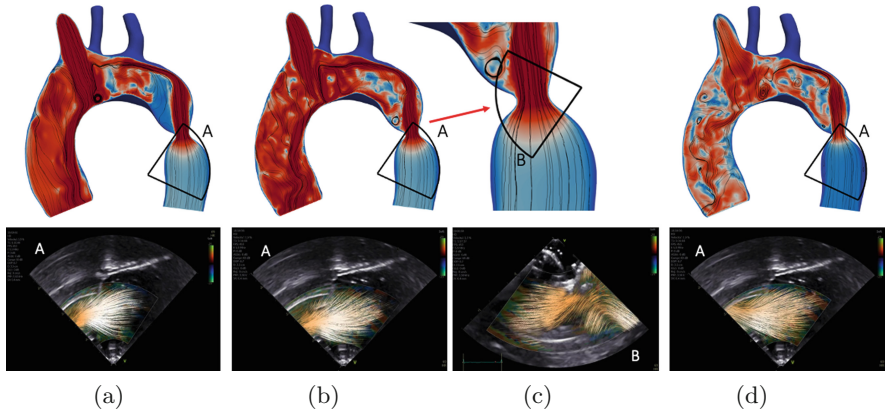
**Fig. 10.** Velocity distribution and streamlines (top), echography results in the ascending (middle) and descending (bottom) aorta at  $t/T = 0.156$ . Considered cases: (a) healthy aorta, (b) aneurismatic aorta and (c) coarctated aorta.



**Fig. 11.** Velocity distribution and streamlines (top), echography results in the ascending (middle) and descending (bottom) aorta at  $t/T = 0.208$ . Considered cases: (a) healthy aorta, (b) aneurismatic aorta and (c) coarctated aorta.



**Fig. 12.** Velocity distribution and streamlines (top), echography results in the ascending (middle) and descending (bottom) aorta at  $t/T = 0.260$ . Considered cases: (a) healthy aorta, (b) aneurismatic aorta and (c) coarctated aorta.



**Fig. 13.** Velocity distribution and streamlines (top) and echography results (bottom) in the coartation region at: (a)  $t/T = 0.156$ , (b)  $t/T = 0.208$ , (c)  $t/T = 0.208$  (zoom), and (d)  $t/T = 0.260$ .

absence of recirculations in both the ascending and descending aorta sections at systolic peak.

## 4 Conclusions

In this work, we integrate *in-vivo*-measured patient-specific data in numerical simulations and in *in-vitro* experiments. Simulations, carried out by using the open-source code *Simvascular*, are compared with experiments performed in a fully-controlled and sensorized circulatory mock loop for 3D-printed aortic models. Both healthy and diseased aortas are considered. The experimental setup allows eliminating the uncertainties on the flow rate and on the wall properties, which are conversely always present in *in-vivo* data. The comparison between CFD and *in-vitro* data is excellent for all the considered cases and the agreement with *in-vivo* data is satisfactory and consistent with the possible controlled and uncontrolled differences with the numerical and *in-vitro* set-up. The velocity fields obtained through CFD are in good agreement with the echography results in *in-vitro* experiments and allowed us to point out the differences between the healthy and pathological cases.

## References

1. Gallo, D., et al.: On the use of in vivo measured flow rates as boundary conditions for image-based hemodynamic models of the human aorta: implications for indicators of abnormal flow. *Ann. Biomed. Eng.* **40**(3), 729–741 (2012)
2. Morbiducci, U., Ponzini, R., Gallo, D., Bignardi, C., Rizzo, G.: Inflow boundary conditions for image-based computational hemodynamics: impact of idealized versus measured velocity profiles in the human aorta. *J. Biomech.* **46**(1), 102–109 (2013)





3. Morbiducci, U., et al.: A rational approach to defining principal axes of multidirectional wall shear stress in realistic vascular geometries, with application to the study of the influence of helical flow on wall shear stress directionality in aorta. *J. Biomech.* **48**(6), 899 (2015)
4. Condemni, F., et al.: Fluid- and biomechanical analysis of ascending thoracic aorta aneurysm with concomitant aortic insufficiency. *Ann. Biomed. Eng.* **45**(12), 2921 (2017)
5. Boccadifuoco, A., Mariotti, A., Capellini, K., Celi, S., Salvetti, M.V.: Validation of numerical simulations of thoracic aorta hemodynamics: comparison with in-vivo measurements and stochastic sensitivity analysis. *Cardiovasc. Eng. Technol.* **4**, 688–706 (2018)
6. Efstathopoulos, E.P., Patatoukas, G., Pantos, I., Benekos, O., Katritsis, D., Kelekis, N.L.: Wall shear stress calculation in ascending aorta using phase contrast magnetic resonance imaging. Investigating effective ways to calculate it in clinical practice. *Physica Medica* **24**(4), 175–181 (2008)
7. Markl, M., Wallis, W., Harlo, A.: Reproducibility of flow and wall shear stress analysis using flow-sensitive four-dimensional MRI. *J. Magn. Reson. Imaging* **33**(4), 988–994 (2011)
8. Morbiducci, U., et al.: Mechanistic insight into the physiological relevance of helical blood flow in the human aorta: an in vivo study. *Biomech. Model. Mechanobiol.* **10**(3), 339–355 (2011)
9. Boccadifuoco, A., Mariotti, A., Celi, S., Martini, N., and Salvetti, M.V., Uncertainty quantification in numerical simulations of the flow in thoracic aortic aneurysms. In: *Proceedings of the 7th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS Congress 2016)*, vol. 3, 6226–6249 (2016)
10. Capellini, K., Vignali, E., Costa, E., et al.: Computational fluid dynamic study for ATAA hemodynamics: an integrated image-based and radial basis functions mesh morphing approach. *J. Biomech. Eng.* **140**(11), 111007 (2018)
11. Boccadifuoco, A., Mariotti, A., Celi, S., Martini, N., Salvetti, M.V.: Impact of uncertainties in outflow boundary conditions on the predictions of hemodynamic simulations of ascending thoracic aortic aneurysms. *Comput. Fluids* **165**, 96–115 (2018)
12. Youssefi, P., Gomez, A., Arthurs, C., Sharma, R., Jahangiri, M., Figueroa, C.A.: Impact of patient-specific inflow velocity profile on hemodynamics of the thoracic aorta. *J. Biomech. Eng.* **140**(1), 1011002 (2018)
13. Antonuccio, M.N., Mariotti, A., Celi, S., Salvetti, M.V.: Effects of the distribution in space of the velocity-inlet condition in hemodynamic simulations of the thoracic aorta. In: Rojas, I., Valenzuela, O., Rojas, F., Herrera, L.J., Ortuño, F. (eds.) *IWBIO 2020. LNCS*, vol. 12108, pp. 63–74. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-45385-5\\_6](https://doi.org/10.1007/978-3-030-45385-5_6)
14. Campbell, I.C., Ries, J., Dhawan, S.S., Quyyumi, A.A., Taylor, W.R., Oshinski, J.N.: Effect of inlet velocity profiles on patient-specific computational fluid dynamics simulations of the carotid bifurcation. *J. Biomech. Eng.* **134**(5), 051001 (2012)
15. Chandra, S., et al.: Fluid-structure interaction modeling of abdominal aortic aneurysms: the impact of patient-specific inflow conditions and fluid/solid coupling. *ASME J. Biomech. Eng.* **135**(8), 081001 (2013)
16. Pasta, S., et al.: Difference in hemodynamic and wall stress of ascending thoracic aortic aneurysms with bicuspid and tricuspid aortic valve. *J. Biomech.* **46**(10), 1729 (2013)



17. Antonuccio, M.N., et al.: Effects of uncertainty of outlet boundary conditions in a patient-specific case of aortic coarctation ANN. Biomed. Eng. **49**(12), 3494–3507 (2021)
18. Schiavazzi, D.E., et al.: Uncertainty quantification in virtual surgery hemodynamics predictions for single ventricle palliation. Int. J. Num. Methods Biomed. Eng. **32**(3), 1–25 (2016)
19. Sarrami-Foroushani, A., Lassila, T., Gooya, A., Geers, A.J., Frangi, A.F.: Uncertainty quantification of wall shear stress in intracranial aneurysms using a data-driven statistical model of systemic blood flow variability. J. Biomech. **49**, 3815–3823 (2016)
20. Brault, A., Dumas, L., Lucor, D.: Uncertainty quantification of inflow boundary condition and proximal arterial stiffness-coupled effect on pulse wave propagation in a vascular network. Int. J. Num. Meth. Biomed. Eng. **33**(10), e2859 (2017)
21. Bozzi, S., et al.: Uncertainty propagation of phase contrast-MRI derived inlet boundary conditions in computational hemodynamics models of thoracic aorta. Comput. Methods Biomech. Biomed. Engin. **20**(10), 1104–1112 (2017)
22. Mariotti, A., Boccadifuoco, A., Celi, S., Salvetti, M.V.: Hemodynamics and stresses in numerical simulations of the thoracic aorta: Stochastic sensitivity analysis to inlet flow-rate waveform. Comput. Fluids **230**, 105123 (2021)
23. Updegrove, A., Wilson, N.M., Merkow, J., Lan, H., Marsden, A.L., Shadden, S.C.: Simvascular: an open source pipeline for cardiovascular simulation. Ann. Biomed. Eng. **45**(3), 525–541 (2016)
24. Vignali, E., Gasparotti, E., Mariotti, A., Haxhiademi, D., Ait-Ali, L., Celi, S.: High-versatility left ventricle pump and aortic mock circulatory loop development for patient-specific hemodynamic in vitro analysis. ASAIO J., Article in Press (2022). <https://doi.org/10.1097/MAT.0000000000001651>



# Sensitivity Analysis of Adhesion in Computational Model of Elastic Doublet

Alžbeta Bohiniková<sup>1</sup>(✉) , Iveta Jančígová<sup>2</sup> , Ivan Cimrák<sup>1,2</sup> ,  
and James J. Feng<sup>3</sup> 

<sup>1</sup> Research Centre, University of Žilina, 010 26 Žilina, Slovakia  
[Alzbeta.Bohinikova@rc.uniza.sk](mailto:Alzbeta.Bohinikova@rc.uniza.sk)

<sup>2</sup> Cell-in-fluid Biomedical Modelling and Computations Group, Faculty of  
Management Science and Informatics, University of Žilina, 010 26 Žilina, Slovakia

<sup>3</sup> Department of Mathematics, University of British Columbia,  
Vancouver, BC V6T 1Z2, Canada

<https://cellinfluid.fri.uniza.sk/>

**Abstract.** This work introduces a computational model of elastic double cluster. We describe a method to create a partially flattened spherical cell and a mirroring process that creates a symmetrical double cluster with desired adhesion surface. The main focus is on the adhesion between the two cells modeled by repulsive-attractive Lennard-Jones potential. We study the stability of the adhesion with respect to the parameters of the Lennard-Jones potential and to the elasticity of the cells. Based on these, a baseline cluster is created and calibrated to a specific separation force using computational experiment that mimics a dual micropipette assay. This cluster is then immersed into elongation flow to create a parallel between the two types of cell stretching experiments: one that mechanically pulls the cell membrane and another where fluid flow creates stress on the membrane. Thus validated, our model of adhesion can be used in more complex clusters and serve as a building block in future computational studies.

**Keywords:** Computational model · Cell clusters · Adhesion · PyOIF

## 1 Introduction

The motivation to separate circulating tumor cell (CTC) clusters into individual cells arises from their higher metastatic potential [9] compared to the individual CTCs as well as their higher resistance to drugs [3]. To better understand how to break up the clusters, it is important to understand their bonds. There are experiments measuring such bonds in flow [17] or using micropipette aspiration [15]. There is also evidence that high shear stress [16] and specific drugs [4, 5] can also help to break them apart.



The work [17] investigates the separation force for clusters consisting of two cells. Using microfluidic chip with sudden narrow constriction they tested different flow conditions (by varying differential pressure in the microfluidic channel) and measured how many clusters separated. With the use of a computational model they determined that a separation force of  $173 nN$  is necessary to separate 50% of the clusters. Even at separation force  $542 nN$ , there were 30% of clusters that did not separate. However, the range of the separation force varies across the literature, and the separation force measured by [17] is very much dependent on the channel design.

Other works look at the behaviour of cluster in various flow situations. A 2D liquid-drop model is used in [16] to represent single cells and doublets. In [1], [13], the clusters are modelled as one stiff mesh consisting of 2, 3 and 4 cells. A 3D elastic model is used to model clusters squeezing into a capillary sized channels in [2]. More detailed study about the adhesion of a single cluster cell to a microvasculature wall was performed in [7]. However, we could not find a study focused on the adhesion between individual cells.

In order to investigate this, we focus on a doublet of two identical cells. First, we briefly describe the model with details on modeling the adhesion bonds and contact surfaces. Then we focus on the pulling experiment (similar to optical tweezers experiment done with biological cells) and finally we consider a comparable elongation flow.

## 2 Computational Model of Double Cluster

### 2.1 Elastic Cells

**Cell Model.** The cells forming the cluster are modeled using a dissipative immersed boundary method [6] in 3D. The membrane is represented by a triangular mesh of nodes connected by elastic bonds. The five employed elastic moduli are stretching, bending, conservation of local area, conservation of global area and conservation of volume. The individual nodes are then coupled to the underlying lattice-Boltzmann model of the surrounding fluid. The model allows for viscosity contrast of the inner and outer fluid by using DPD particles inside the cell.

**Cluster Model.** While the individual (spherical) cells have a relaxed shape defined by their initial geometry and bonds of the mesh points, the cluster shape is determined by non-bonded interactions of points on neighboring membranes. As a consequence, the clusters may change shape and also the cells forming a cluster may separate. More information about the model and its implementation can be found at [18].

**Cell Size.** The size of the CTC varies depending on the type and stage of cancer and on the variation within the cell population. In [21] the cell line MDA-MB-231 has diameters  $12.4 \mp 2.1 \mu m$  (average of 128 cells) and the line MCF-10A has  $11.2 \mp 2.4 \mu m$  (average of 158 cells). For this work we chose cell diameter at the lower end of these ranges:  $2r_{cell} = 10 \mu m$ .

**Elastic Parameters.** The original PyOif model [12] was calibrated for red blood cells (RBCs) using the optical tweezers data [11]. Qualitative observations [13] suggest that the elasticity of tumor cells can vary considerably, and as shown in e.g. [14], computational experiments are sensitive to the elastic parameters in the model. Considering that the CTCs are generally stiffer than RBCs [20], we have chosen the following moduli for our model cells:  $k_s = 0.05 \mu\text{N}/\text{m}$ ,  $k_b = 0.005 \text{Nm}$ ,  $k_{al} = 0.02 \mu\text{N}/\text{m}$ ,  $k_{ag} = 0.7 \mu\text{N}/\text{m}$ ,  $k_v = 0.9 \mu\text{N}/\text{m}^2$ .

## 2.2 Adhesion

For the cells to create a cluster, they need to have an attractive force between them. However, the force cannot be only attractive, because computationally this would lead to cells collapsing onto each other. Real biological cells do not collapse but connect with bonds of small but finite length. To achieve this behaviour, we also need a repulsive force at very close range, that would prevent the cells from overlapping.

**Lennard-Jones (LJ) potential** is frequently used to model particle-particle interactions in coarse-grained simulations to represent interactions that are attractive at large distance and strongly repulsive at short distances [19]. Typically, in simulations it also has a cutoff distance and is only evaluated when the two particles (in our case a pair of points, one on each cell membrane) are closer than this cutoff distance. To calculate the LJ interaction energy one needs to consider the number of pairwise LJ interactions per square unit of membrane surface.

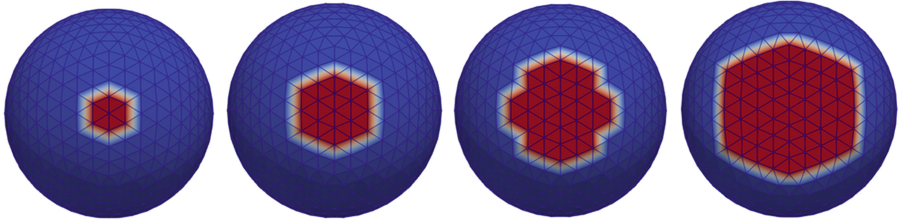
The potential is defined as:

$$V_{LJ}(r) = \begin{cases} 4\epsilon_{LJ} \left[ \left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6 \right] & \text{if } r < r_{cut} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

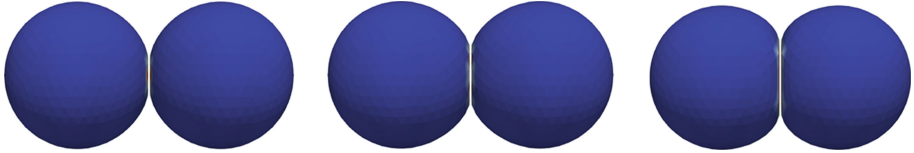
where  $r$  is the distance between the interacting particles,  $r_{cut}$  is the cutoff distance and  $\epsilon$  scales the strength of the interaction. The parameter  $\sigma$  determines the distance  $r_{min}$  where the repulsion changes to attraction. This switch occurs at the minimum of the potential, when  $r = r_{min} = \sqrt[6]{2}\sigma$ .

**Adhesion Surface.** Two cells in a double cluster are adhered by a circular area with a given adhesion diameter. In double clusters images, e.g. in [9], the ratio between adhesion diameter and cell diameter is around 60%. So for our clusters we selected  $r_{surf} = 3 \mu\text{m}$  as a suitable radius of the spherical contact surface.

The typical shape of double cluster has two cells that are flattened at the contact surface. The  $yz$  plane is the plane of symmetry along which we flattened the cells. The points on the cells were selected to achieve the desired adhesion radius. And then the distance between the adhesion surfaces of the cells was set to  $r_{min}$ . The second cell was created as a mirror image of the first. This guarantees that we have pairs of points facing each other on the adhesion surface and with an appropriate choice of LJ parameters we can have each mesh point interacting



**Fig. 1.** Possible adhesion areas for baseline cluster with  $k_b = 0.005 Nm$ , see Table 3.



**Fig. 2.** Profiles of clusters with various adhesion areas shown in Fig. 1 for baseline cluster with  $k_b = 0.005 Nm$ , see Table 3. Number of points on adhesion surface, from left to right: 19, 33, 61.

with exactly one mesh point of the other cell, which offers more control over the interaction and more stability of adhesion.

**Adhesion Strength.** Apart from the size of the surface, the strength of the adhesion is important, too. In [8], the authors used a micropipette aspiration method to measure the cell-cell adhesion strength of various human embryonic kidney cell clones, and determined them to be  $2\text{--}12 nN$ . The cell-cell adhesion measurements in [15] give the separation force of mesoderm and endoderm cells in the range  $2\text{--}5 nN$ . Based on these measurements we aimed to model a cluster with adhesion which separates under applied force between  $1\text{--}2 nN$ . More specifically our baseline is a cluster that holds up to  $1.5 nN$  and separates at  $1.6 nN$ . We also discuss how we can change the cluster properties through the LJ interaction parameters to model other separation forces.

**Stable Clusters.** The clusters were created by putting two flattened and mirrored cells next to each other, flat sides facing, and applying the LJ interaction. We placed the cells at the equilibrium distance  $r_{min}$ , where there should be no LJ influence, provided that the only points in the interactions are the ones facing each other. This can be achieved by setting the parameters  $r_{min}$  and  $r_{cut}$  in such a way that the closest neighbour of the opposite point (considering the smallest edge length in the triangulation of the mesh) is further than  $r_{cut}$  and thus out of the range of the interaction. The clusters were then left to relax until the change of axial length of the cell was less than  $0.01\%$  per  $10 \mu s$ .

It is important to note that even though the adhesion surfaces were set to be at the equilibrium of the LJ interaction, the relaxed distance between the cells was always slightly under  $r_{min}$ . This was expected, since the cells are elastic and

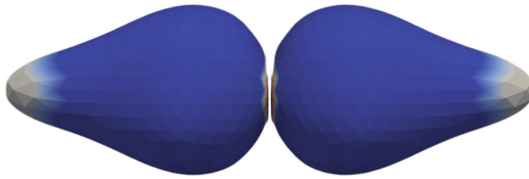
attempt to resume their original spherical shape. This pushes the points from the flattened area closer together. These points are then closer than  $r_{min}$  and the LJ interaction starts to repulse them. The final distance between the cells is then the distance where the forces are in balance.

### 2.3 Pulling Experiment

To determine the strength of the adhesion modeled with LJ potential we designed a simulation experiment mimicking the dual micropipette assay, such as the biological experiment in [8]. To achieve similar stretching, we pulled a cap of each cell with radius  $2\ \mu m$ . The pulling cap can be seen in Fig. 3. This size was selected as the most typical pipette radius [10].

Mirrored and flattened clusters were loaded into channels with static fluid that provides damping. The viscosity of the fluid was set to  $1.5\ mPa\ s$ . The simulation was run until the gap between the cells was larger than  $1\ \mu m$ , or until the cell stretched and the adhesion area stabilized, assuring that the cluster will not separate.

Using this experiment, we studied how individual parameters of cluster model influence the final behaviour of the cluster.



**Fig. 3.** Snapshot of cell deformation halfway through the pulling experiment. The white part of the mesh marks points to which the outward force is equally applied.

## 3 Stability of Adhesion Surface

Changes in parameters  $r_{min}$  and  $r_{cut}$  can improve the stability of the adhesion area. As shown in Table 1, the change to  $r_{min}$  does not influence the size of the stable area. For the same cell cluster we carried out a set of experiments where  $r_{min}$  was fixed and  $r_{cut}$  increased. As long as the  $r_{cut}$  was smaller than the distance to the second closest point on the opposite cell, the changes had no influence. This shows that the cell cluster with given elastic parameters, cell radius, mesh, adhesion area (represented by contact radius  $r_{surf}$  and number of mesh points shown in the final column of Table 1) and  $\epsilon_{LJ}$  is stable. However, this stability changes if the elasticity of the cell changes.

Of the five employed elastic moduli, bending, which conserves angles between pairs of mesh triangles, is the most important modulus to the adhesion surface. In order to test how the changes in elasticity influence the stability of the adhesion, we tested changes in bending parameter  $k_b$ . With increased  $k_b$ , the cell becomes

**Table 1.** Contact radius  $r_{surf}$  does not depend on the size of the LJ repulsive region, with a set width of the attraction region of  $0.15 \mu m$  and other elastic and interaction parameters held constant (baseline cluster). The gap between the cells' flat adhesion surfaces is set to  $r_{min}$  (the actual gap is shown in column 3) and there is no other force applied to the cells. They are left to relax until the change in their axial length is less than 0.01% per  $10 \mu s$ .

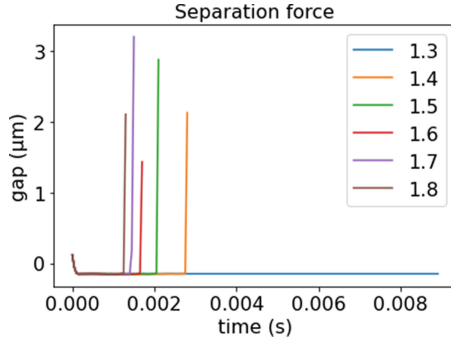
$r_{min}[\mu m]$	$r_{cut}[\mu m]$	gap $[\mu m]$	$r_{surf}[\mu m]$	points [-]
0.10	0.25	0.0999	3.08521	61
0.15	0.30	0.1497	3.08521	61
0.20	0.35	0.1994	3.08520	61
0.25	0.40	0.2491	3.08519	61
0.30	0.45	0.2988	3.08518	61
0.35	0.50	0.3484	3.08517	61

more rigid and resists the flattening of part of its surface more. Though the stable area changes only slightly, in Table 2 we can see that for  $k_b \geq 0.02 Nm$  the gap between the surfaces collapses and is no longer at  $r_{min}$ .

The gap is calculated as distance (in  $x$ -direction) between the leftmost point of the right cell and the rightmost point of the left cell, see also Fig. 4 for separation outcomes with a given elasticity and varying separation force. A negative gap represents the fact that the cells are overlapping. The contact area is still a flat surface, e.g. Fig. 2, but we see that higher pulling force leads to more prolonged cluster and smaller adhesion area.

**Table 2.** Influence of bending elasticity  $k_b$  on the stability of the adhesion radius  $r_{surf}$  and separation force  $F_s$ . **o** denotes a cluster that holds when the given force is applied. **x** denotes a cluster that separates when the given force is applied. Other elastic coefficients and interaction parameters are set to the baseline cluster. The gap between the cells' flat adhesion surfaces is set to  $r_{min}$  (the actual gap is shown in column 3). There is no other force applied to the cells. They are left to relax until the change in their axial length is less than 0.01% per  $10 \mu s$ . Number of mesh points on the contact surface is in column 4.

$k_b[Nm]$	gap $[\mu m]$	$r_{surf}[\mu m]$	points[-]	$F_s [nN]$					
				1.3	1.4	1.5	1.6	1.7	1.8
0.000625	0.1499	3.1863	65	o	o	o	o	o	x
0.00125	0.1499	3.1864	65	o	o	o	o	o	x
0.0025	0.1498	3.1865	65	o	o	o	o	x	x
0.005	0.1497	3.0852	61	o	o	o	x	x	x
0.01	0.1485	3.0854	61	o	x	x	x	x	x
0.02	-0.1580	3.0812	61	x	x	x	x	x	x
0.04	-0.1585	3.0810	61	x	x	x	x	x	x



**Fig. 4.** Gap between cells for double cluster from Table 2 with  $k_b = 0.01 Nm$ . Forces are given in  $[nN]$ . The abrupt ends of almost vertical lines represent the fact that the cluster has separated. The initial downward shift in all cases means that the starting gap was  $0.15 \mu m$  and at the beginning of the simulation the membranes crossed over and stabilized at distance  $-0.15 \mu m$ .

The adhesion surface of more rigid cells is smaller than the one we have selected as the baseline and consequently more stable as shown in Table 3.

**Table 3.** Size of the adhesion area, represented by the number of points, depending on cells' elasticity and initial radius of the flattened surface. The stars note that the cells are overlapping.

$k_b[Nm]/r_{surf}[\mu m]$	0.5	1	1.5	2	3
0.000625	7	7	19	33	65
0.00125	7	7	19	33	65
0.0025	7	7	19	33	65
0.005	7	7	19	33	61
0.01	7	7	19	33	61
0.02	7	7	19*	29*	61*
0.04	7	7	19*	25*	61*

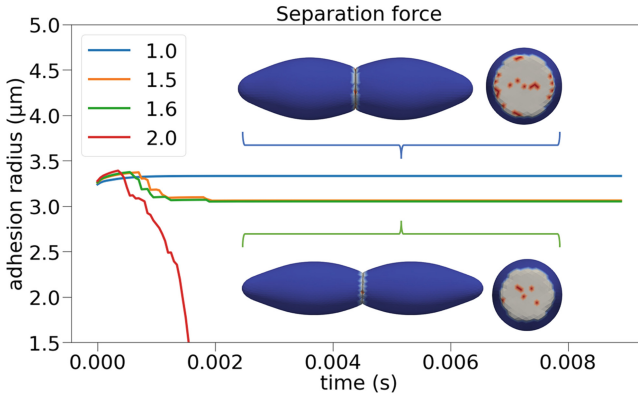
Adhesion strength parameter  $\epsilon_{LJ}$  can be used to prevent cells from overlapping, but it also influences the magnitude of separation force necessary, see Table 4. With increasing  $\epsilon_{LJ}$  the separation force  $F_s$  also increases.

To achieve more stable adhesion surface with gap at  $r_{min}$  levels, even if the cells are more rigid and the separation force kept at the desired level, a change in  $r_{min}$  and  $r_{cut}$  can help.

Table 5 shows that the first estimate for the value of  $\epsilon_{LJ}$  (for mesh with 1182 points) was approximately  $0.0036 fNm$ . This led to cluster separating even at the smallest separation force we tested,  $F_s = 0.5 nN$ . As we can see in

**Table 4.** Separation force necessary for baseline cluster (642 mesh nodes per cell) depending on the adhesion parameter  $\epsilon_{LJ}$ . The columns  $r_{surf}$ ,  $points$  and  $gap$  show the stability of the adhesion surface when no external forces are applied. The LJ potential is set to  $r_{min} = 0.15 \mu m$  and  $r_{cut} = 0.3 \mu m$ .

$\epsilon_{LJ}[fNm]$	gap[ $\mu m$ ]	$r_{surf}[\mu m]$	points[-]	$F_s[nN]$				
				0.5	1.0	1.5	1.6	2.0
0.0025	0.14817	3.08517	61	o	x	x	x	x
0.005	0.14920	3.08520	61	o	o	x	x	x
0.0066	0.14968	3.08521	61	o	o	o	x	x
0.0075	0.14972	3.08521	61	o	o	o	o	x
0.01	0.14979	3.08521	61	o	o	o	o	o



**Fig. 5.** Small variability in stabilized adhesion of clusters that do not separate, demonstrated using 1992 node cluster with LJ parameters  $\epsilon_{LJ} = 0.005 fNm$ ,  $r_{cut} = 0.25 \mu m$  and  $r_{min} = 0.2 \mu m$ . The inset figures show profile and the adhesion area for pulling forces 1 nN and 1.6 nN.

Table 7, to achieve  $F_s$  between 1.5 nN and 1.6 nN,  $\epsilon_{LJ}$  needs to be approximately 0.0065 fNm. This is in contrast to what can be seen for coarser discretization of our baseline cluster with 642 mesh points. The stability of the adhesion surface is not influenced by changing  $\epsilon_{LJ}$ , as seen in Table 4, nor is it changed by moving  $r_{min}$  and  $r_{cut}$  as seen in Table 1.

For denser meshes, there is a higher risk of instability. This arises mainly from the interplay between the cell trying to achieve its original shape and the LJ interaction. During the initialization of the cluster, the cells are flattened, and positioned at distance  $r_{min}$ . In the next iteration step, some of the points on the flattened adhesion surface are pushed out, mainly by the bending interaction. This instantly puts them into the repulsive region of the LJ potential. Depending on how close they get to the second cell they are repulsed by a corresponding force, which pushes them into the attractive region of the LJ potential.

This fluctuation, stabilizes into either a gap less than  $r_{min}$  or above  $-r_{min}$  (when the cells overlap), or the whole system diverges. By increasing  $r_{min}$  we allow for more space. So if we take similarly deformed meshes, the point (close to the border of the flattened surface) that is pushed into the repulsive region of LJ, is pushed with about the same force (since the deformation of the cell's surface is the same) for any value of  $r_{min}$  (since at the beginning the cells are  $r_{min}$  apart). However, with higher  $r_{min}$  the repulsive force given to this point is smaller and allows for more stable adhesion between cells.

**Table 5.** Stability of adhesion surface for 1182 node discretization.

simID	$\epsilon_{LJ}[fNm]$	$r_{min}[\mu m]$	$r_{cut}[\mu m]$	gap $[\mu m]$	$r_{surf}[\mu m]$	points [-]
1	0.0036	0.15	0.3	0.1474	3.36467	133
2	0.0060	0.15	0.3	-0.1337	3.36085	133
3	0.0060	0.15	0.25	-0.1404	3.36052	133
4	0.0060	0.2	0.3	0.1993	3.36467	133
5	0.0060	0.2	0.25	0.1993	3.36466	133

Thus,  $r_{min}$  should be set as small as possible to mimic the qualitative shape of biological cell clusters, whose membranes touch at the adhesion area, but large enough so that the adhesion is stable. The interaction cutoff  $r_{cut}$  should be set smaller than the distance of the point to its second closest neighbour on the opposite cell. If we set  $r_{cut}$  higher than this value, we could end up with one point being repulsed by one point but at the same time attracted by all six neighbours of this point, which would lead to instability. We calculated this threshold value for each mesh we used, see Table 6, as follows.

Since the cells are mirrored at the beginning, taking a pair of points facing each other from each cell and one of their closest neighbours, creates a right angle triangle. The distance between the opposing points is  $r_{min}$  and we estimate the distance between a point and its closest neighbour as the smallest edge length of our mesh  $e_{min}$  and then the maximum value for  $r_{cut}$  can be calculated as:  $r_{cut_{max}} = \sqrt{r_{min}^2 + r_{cut}^2}$ .

**Baseline Cluster.** As mentioned in previous sections, our baseline cluster has the following parameters:  $r_{cell} = 5 \mu m$ ,  $r_{surf} = 3 \mu m$ ,  $k_s = 0.05 \mu N/m$ ,  $k_b = 0.005 Nm$ ,  $k_{al} = 0.02 \mu N/m$ ,  $k_{ag} = 0.7 \mu N/m$ ,  $k_v = 0.9 N/m^2$ . For discretization we selected a mesh with 642 points. LJ parameters were set to  $r_{min} = 0.15 \mu m$ ,  $r_{cut} = 0.3 \mu m$  and  $\epsilon_{LJ} = 0.0066 fNm$ .  $r_{min}$  was selected the smallest possible to keep the cells from overlapping.  $r_{cut}$  was selected smaller than  $r_{cut_{max}} = 0.62 \mu m$ , as calculated in Table 6 to have only one-to-one point LJ interaction on the adhesion surface, and then adjusted to achieve separation at 1.6 nN.  $\epsilon_{LJ}$  was also tuned to achieve the desired separation force. This was done by running multiple parameter combinations in the pulling experiment. Similarly, we ran experiments



**Table 6.** Maximum value of  $r_{cut}$  for various discretisations.  $e_{min}$   $e_{max}$  and  $e_{mean}$  denote minimal, maximal and mean edge.

$n_{nodes}[-]$	$e_{min}[\mu m]$	$e_{max}[\mu m]$	$e_{mean}[\mu m]$	$r_{min}[\mu m]$	$r_{cut_{max}}[\mu m]$
482	0.686	0.956	0.868	0.15	0.70
642	0.602	0.823	0.750	0.15	0.62
1182	0.332	0.779	0.556	0.2	0.39
1524	0.293	0.689	0.489	0.2	0.36
1922	0.338	0.480	0.434	0.2	0.39

for other mesh discretizations, see Table 7, to demonstrate consistent behavior across different levels of coarse-graining.

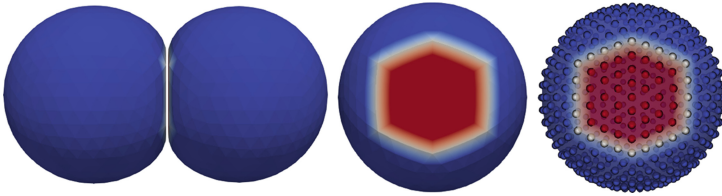
It is important to note that also values close to that stated in the table would lead to similar separation force. This would be valuable for more precise fine-tuning of the LJ interaction. This table should be used as a tool to initialise a cluster with similar behaviour as the baseline, only with different discretizations, that might be needed for simulations with more narrow channels.

## 4 Elongation Flow

To better mimic the microfluidic conditions, we also considered separation of double clusters in elongation flow. The flow is achieved by having the inflow at two opposite sides of the microfluidic chamber, as shown in Table 8, and outflow on two perpendicular sides. The cells are placed at the center in such a way that the contact area is perpendicular to the outflow. This way the flow drives the separation. The boundary inflow velocity is then adjusted to determine which velocities lead to separation and which are not strong enough.

**Table 7.** Various discretizations for baseline cluster. Values of  $r_{min}$  and  $r_{cut}$  were selected to achieve stable adhesion area with radius of  $3\mu m$ .  $\epsilon_{LJ}$  was selected through series of pulling experiments with various forces and various values of  $\epsilon_{LJ}$ .  $\epsilon_{LJ}$  stated in the table results in clusters separating between 1.5 and 1.6 nN.

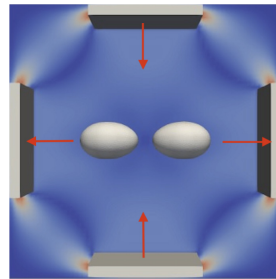
$n_{nodes}[-]$	$\epsilon_{LJ}[fNm]$	$r_{min}[\mu m]$	$r_{cut}[\mu m]$
482	0.0070	0.15	0.3
642	0.0066	0.15	0.3
1182	0.0065	0.2	0.25
1524	0.0060	0.2	0.25
1992	0.0048	0.2	0.25



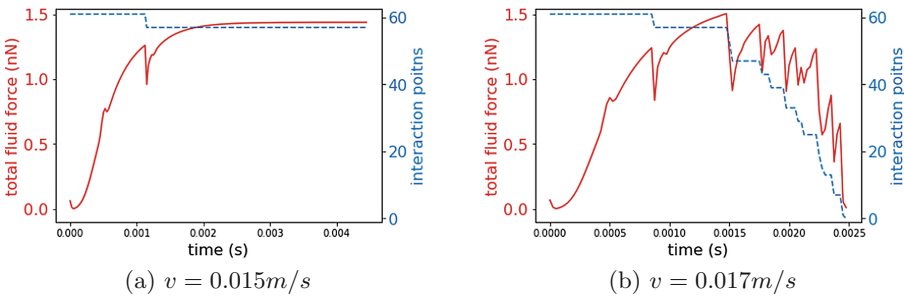
**Fig. 6.** Baseline cluster. The red points are at distance less than  $r_{min}$ , the white points are at distance between  $r_{min}$  and  $r_{cut}$ . (Color figure online)

**Table 8.** Validation of various discretization setting for the baseline cluster. The discretizations from Table 7 were used in elongation flow and exhibit consistent behavior: separating at inflow velocities 0.017 mm/s and above (x) and holding attached at inflow velocities 0.015 mm/s and below (o).

$n_{nodes} [-]$	$v_{inflow} [mm/s]$	
	0.015	0.017
482	o	x
642	o	x
1182	o	x
1524	o	x
1992	o	x



Using the cluster discretizations from Table 7 we determined the separation inflow velocity of elongation flow to be  $\sim 0.016 \text{ mm/s}$  (as measured at the center of the boundary, see Table 8).



**Fig. 7.** Fluid force on cells in elongation flow. The red line indicates the fluid force acting on a cell at a given time, the blue dashed line shows number of points on the contact area. (a) no separation at lower fluid velocity (b) higher flow results in cluster separation. (Color figure online)

## 4.1 Fluid Force on Cell

To link the flow and force conditions needed to separate a given cluster, we measured the total fluid force acting on each cell in the elongation flow. This force has the same magnitude and opposite direction for the two cells and is calculated as a sum of fluid forces from all the individual mesh points, Fig. 7.

At lower applied fluid velocity in the elongation flow, Fig. 7 (a), we see an increase in the total fluid force as the cell membrane stretches and thus moves relatively to the surrounding fluid. The sharp jump corresponds to the moment when the contact area decreases (some of the bonded pairs no longer hold). The fluid force on the object then equalizes with the adhesion force and the system is at equilibrium.

At larger applied fluid velocity in elongation flow, Fig. 7 (b), we see a similar initial increase in the total fluid force, followed by multiple sharp jumps. Each of these corresponds to the contact area decreasing (blue line), when some of the bonded points no longer hold. Before the fluid force has a chance to equalize with the adhesion force another jump occurs, ultimately leading to cell separation. At that point the total fluid force is 0, indicating both cells are moving with the fluid.

While the correspondence is not perfect (most likely due to numerical reasons), we see that a cluster that separates at 1.5–1.6  $nN$  pipette pulling force, holds at  $\sim 1.4 nN$  fluid pulling force and separates around 1.5  $nN$ . This means we can use the total acting fluid force as a proxy when evaluating the strength of adhesion in flow.

## 5 Conclusion

The adhesion area and its stability depend on many factors. With increased cell rigidity, represented by higher values of the bending parameter, the adhesion surface becomes less stable, especially for larger contact surfaces. We have shown that to a certain extent this instability can be managed by appropriate settings of the LJ potential parameters. To increase the stability, the repulsion/attraction threshold  $r_{min}$  can be increased, which leads to fewer fluctuations. The increase of  $\epsilon_{LJ}$  can also improve the stability of the relaxed adhesion surface, however it is directly proportional to the adhesion strength.

To satisfy the need for various discretizations of cell membrane, we have calibrated our baseline cluster for five meshes of various densities. Based on these, appropriate parameters for other meshes can be reasonably interpolated. The values in Table 7 suggest that to achieve the same behaviour of the cluster with an increased number of nodes,  $\epsilon_{LJ}$  must be lowered and if the stability requires it,  $r_{min}$  increased and  $r_{cut}$  adjusted accordingly. We have explained and calculated the upper boundary for the value of  $r_{cut}$ , see Table 6.

This setup allows us to simulate any type of double cluster with varying elasticity, adhesion strength and adhesion surfaces. Building on this, more complex clusters can be explored, with higher number of cells and varying cell sizes.

Another direction of future work is to look at the behavior of this type of cluster under different flow conditions, such as in shear flow, parabolic flow or more complex flows with other types of cells. We have shown the first step in this direction with the elongation flow and determining the flow velocity that corresponds to the separation force in the pulling experiment.

**Acknowledgements.** This research was supported by Operational Program “Integrated Infrastructure” of the project “Integrated strategy in the development of personalized medicine of selected malignant tumor diseases and its impact on life quality”, ITMS code: 313011V446, co-financed by resources of European Regional Development Fund.

This work was also supported by the Slovak Research and Development Agency (contract number APVV-15-0751).

James J. Feng acknowledges support by the Natural Sciences and Engineering Research Council of Canada (Discovery Grant No. 2019-04162).

## References

1. Anderson, K.J., de Guillebon, A., Hughes, A.D., Wang, W., King, M.R.: Effect of circulating tumor cell aggregate configuration on hemodynamic transport and wall contact. *Math. Biosci.* **294**, 181–194 (2017)
2. Au, S.H., et al.: Clusters of circulating tumor cells traverse capillary-sized vessels. *Proc. Natl. Acad. Sci.* **113**(18), 4947–4952 (2016)
3. Bithi, S.S., Vanapalli, S.A.: Microfluidic cell isolation technology for drug testing of single tumor cells and their clusters. *Sci. Rep.* **7**(1), 1–12 (2017)
4. Bonnet, J., et al.: Mitotic arrest affects clustering of tumor cells. *Cell Div.* **16**(1), 1–13 (2021)
5. Choi, J.W., Kim, J.K., Yang, Y.J., Kim, P., Yoon, K.H., Yun, S.H.: Urokinase exerts antimetastatic effects by dissociating clusters of circulating tumor cells. *Can. Res.* **75**(21), 4474–4482 (2015)
6. Cimrák, I., Jančigová, I.: *Computational Blood Cell Mechanics*, 1st edn. CRC Press, Boca Raton (2018)
7. Dabagh, M., Gounley, J., Randles, A.: Localization of rolling and firm-adhesive interactions between circulating tumor cells and the microvasculature wall. *Cell. Mol. Bioeng.* **13**(2), 141–154 (2020)
8. Daoudi, M., et al.: Enhanced adhesive capacities of the naturally occurring ile249-met280 variant of the chemokine receptor cx3cr1. *J. Biol. Chem.* **279**(19), 19649–57 (2004)
9. Edd, J.F., et al.: Microfluidic concentration and separation of circulating tumor cell clusters from large blood volumes. *Lab Chip* **20**(3), 558–567 (2020)
10. Hochmuth, R.M.: Micropipette aspiration of living cells. *J. Biomech.* **33**(1), 15–22 (2000)
11. Jančigová, I., Kovalčíková, K., Bohiniková, A., Cimrák, I.: Spring-network model of red blood cell: From membrane mechanics to validation. *Int. J. Numer. Meth. Fluids* **92**(10), 1368–1393 (2020)
12. Jančigová, I., Kovalčíková, K., Weeber, R., Cimrák, I.: Pyoif: computational tool for modelling of multi-cell flows in complex geometries. *PLOS Comput. Biol.* **16**(10), 1–21 (2020)

13. King, M.R., et al.: A physical sciences network characterization of circulating tumor cell aggregate transport. *Am. J. Physiol. Cell Physiol.* **308**(10), C792–C802 (2015)
14. Kovalcikova, K., Bachraty, H., Bachrata, K., Buzakova, K.: Numerical experiment characteristics dependence on red blood cell parameters, pp. 286–292 (2021)
15. Maître, J.L., et al.: Adhesion functions in cell sorting by mechanically coupling the cortices of adhering cells. *Science* **338**(6104), 253–256 (2012)
16. Marrella, A., et al.: High blood flow shear stress values are associated with circulating tumor cells cluster disaggregation in a multi-channel microfluidic device. *PLoS ONE* **16**(1), e0245536 (2021)
17. Mutlu, B.R., et al.: In-flow measurement of cell-cell adhesion using oscillatory inertial microfluidics. *Lab Chip* **20**(9), 1612–1620 (2020)
18. PyOIF: Computational Tool for Modelling of Multi-Cell Flows in Complex Geometries: Resources webpage for pyoif. online January 2022. <https://cellinfluidfriunizask/resources-espresso/>
19. Rapaport, D.C.: *The Art of Molecular Dynamics Simulation*, 2nd edn. Cambridge University Press, Cambridge (2004)
20. Shaw Bagnall, J., Byun, S., Begum, S.E.A.: Deformability of tumor cells versus blood cells. *Sci Rep* **5**(1), 1–11 (2015)
21. TruongVo, T., et al.: Microfluidic channel for characterizing normal and breast cancer cells. *J. Micromech. Microeng.* **27**(3), 035017 (2017)



# Increasing the Accuracy of OptiPharm's Virtual Screening Predictions by Implementing Molecular Flexibility

Savíns Puertas-Martín<sup>1</sup>(✉) , Juana L. Redondo<sup>1</sup> , Ester M. Garzón<sup>1</sup> ,  
Horacio Pérez-Sánchez<sup>2</sup> , and Pilar M. Ortigosa<sup>1</sup>

<sup>1</sup> Supercomputing - Algorithms Research Group (SAL), University of Almería, Agrifood Campus of International Excellence, ceiA3, Almería 04120, Spain  
{savinspm, jlredondo, gmartin, ortigosa}@ual.es

<sup>2</sup> Structural Bioinformatics and High Performance Computing Research Group (BIO-HPC), Universidad Católica de Murcia (UCAM), 30107 Murcia, Spain  
hperez@ucam.edu

**Abstract.** Recently, a new piece of software called OptiPharm has been proposed to optimize the similarity between two given molecules. A comprehensive study proved it was very competitive compared with state-of-the-art algorithms such as WEGA and ROCS. However, all of these methods, including OptiPharm, assume the proteins as rigid, resulting in poor or inefficient predictions. The consideration of conformational changes and thus the molecule's flexibility is necessary. In this work, we have extended the OptiPharm's functionality by applying a methodology that considers the flexibility of the molecules. Apart from that, the new OptiPharm presents some strengths regarding its previous version. More precisely, it reduces the search space dimension and introduces circular limits for the angular variables to enhance searchability. As results will show, these improvements help OptiPharm to achieve better predictions.

**Keywords:** Ligand based virtual screening · Molecule's flexibility · Optimization

## 1 Introduction

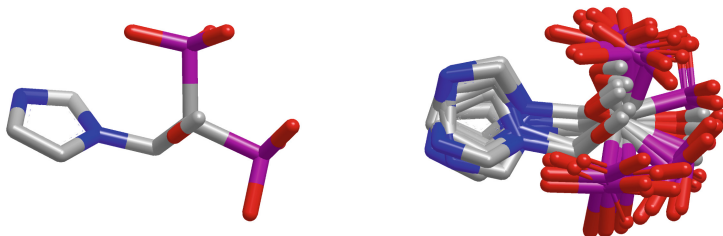
Virtual Screening (VS) methods can be divided into structure-based (SBVS) and ligand-based (LBVS) methods. When the structure of the protein target is known, SBVS can be applied, and methods such as Molecular Docking [22], and Molecular Dynamics [6] are used. Unfortunately, the number of crystallographic structures that solved until now is not sufficient enough [12], so SBVS methods cannot always be applied. As an alternative, LBVS methods can always be used, where only known data (descriptors) (their chemical composition or expression, electrostatic potential, color) are used to derive new and improved ligands.

There are different categories [11] of LBVS methods, such as pharmacophore methods [17], similarity methods [21], QSAR [5], Machine Learning [1], etc.

This work focuses on similarity LBVS methods. In shape similarity methods or similarity of another descriptor, the starting point is a source drug whose shape or other descriptor is known. This source ligand or crystal will be the query, and the virtual screening methods try to find which other ligands or molecules in a large database or chemolibrary are similar to this query. When calculating the similarity between the query and a compound in the database, the optimal position and 3D space compound orientation must be found to allow maximum similarity. It is in this process of searching for the best orientation that optimization algorithms are used [7,20].

Protein flexibility is necessary for metabolism, transport, and function biological effects. When a single protein structure is investigated (the protein is considered as rigid), the functions of the protein considered are poor, where the intrinsic dynamics and the motions allowed by the rotatable bonds (see Fig. 1) are not taken into account, resulting in inefficient results [4,8]. For this reason, to identify new inhibitory compounds, it is necessary to consider conformational changes and thus the molecule's flexibility [3,18]. Except for simple molecules such as  $O_2$ , both ligands and receptors are flexible molecules. Therefore, there is no single three-dimensional representation of these molecules when they are free in solution or forming the ligand-receptor complex at the organism's temperature. On the contrary, receptors and ligands possess many thermally accessible states, which define the accessible conformational space at a given temperature, and which essentially reflect the internal flexibility due to rotations through single bonds (see Fig. 1). The conformational richness increases exponentially with its size since the more atoms (and therefore bonds, angles, and twists) it possesses, the more degrees of freedom there are. These degrees of freedom are not additive but multiply, giving rise to many possible conformational states.

The Virtual Screening studies based on ligands in which ligand flexibility is considered [2,9,10], consider the protein to be almost rigid or with partial flexibility, so that of all the possible rotatable single bonds, they only allow a maximum of 5 of them to rotate. In some cases, the solution is to perform the Virtual Screening considering the molecule to be rigid, and then apply a process in which the flexibility is studied for the number of rotatable links allowed by the algorithm [9]. This process sometimes consists of varying in a discrete way each of the angles of the rotatable bonds to find the best solution. Other authors include gradient algorithms [2] to find the best rotation of these simple links. In [10], apart from studying flexibility as a refinement process after Virtual Screening with rigid molecules, he also studies flexibility from the beginning of the Virtual Screening process, considerably increasing the number of parameters to be optimized.



(a) A molecule of the target FPPS that has some rotatable bonds. (b) A set of conformations generated from the rigid FPPS molecule.

**Fig. 1.** A rigid molecule (a) can generate different conformations (b). An example for the farnesyl diphosphate synthase (FPPS) target from the DUD-E database is shown here.

## 2 Methods

### 2.1 Shape Similarity Scoring Function

The shape similarity value of two molecules A and B is measured as the overlapping volume of their atoms. In this work, the similarity function is implemented as in WEGA [20] where the function is expressed in the following form:

$$V_{AB}^g = \sum_{i \in A, j \in B} w_i w_j v_{ij}^g \quad (1)$$

Considering that the different sizes of distinct molecules can imply significant overlapping variations, it is essential to normalize this overlapping. For this purpose, different metrics exist in the literature such as Tversky [19] or Tanimoto [16] coefficients. In this work, it has been decided to use the later because it is used in well-known virtual screening software like ROCS [15], WEGA [20], and Align-it [9]. Consequently, the score of Eq. 1 is normalized using the Tanimoto coefficient [16], which is calculated as follows:

$$Tc = \frac{V_{AB}}{V_{AA} + V_{BB} - V_{AB}} \quad (2)$$

where  $V_{AA}$  and  $V_{BB}$  is the self-overlap volume of molecules A and B, respectively. It returns a value between [0, 1] where two molecules are more similar the closer the value is to 1, and vice versa.

### 2.2 OptiPharm Algorithm

This section briefly describes OptiPharm, the software used to perform the experiments in this work. For further reading, it is recommended to read their original work [14]. However, given the increased complexity of the problem addressed



here, new mechanisms implemented in OptiPharm to reduce the computational cost will be detailed below.

OptiPharm is a recent software designed explicitly for LBVS problems. It implements a global evolutionary optimizer capable of calculating the similarity between two compounds, a target, and a query. To do so, it uses different methods in the optimization process to gradually adjust the position of the query while the target fixes its position.

To explore the solution space, OptiPharm works with a user-defined population of size  $M$ , which applies reproduction, selection, and improvement methods to each member of the population. A member or solution of this population represents the rotation and translation of the query molecule. Originally ten parameters were used to represent this modification, which means to work in a 10-dimensional search space. This paper presents a new version of Optipharm, where the search space is reduced to 6 dimensions. The main change consists of replacing the use of quaternions with a semi-sphere parametrization, which simplifies the definition of the rotation axis. Consequently, searchability is enhanced due to the reduction of the search space dimension. Nevertheless, not only that, this new system avoids the repetition of the same rotation axis already explored.

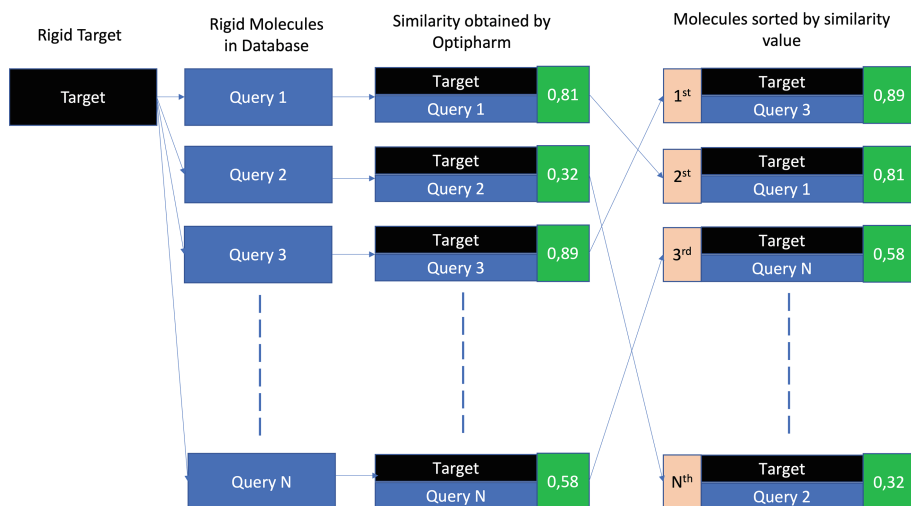
This new mechanism provides improved freedom of exploration. In addition to reducing input parameters, the new version incorporates some problem knowledge, such as a mechanism to keep the angular variables between 0 and  $2\pi$  in a continuous circular. So, if during the search an angle  $\alpha$  takes a value greater than  $2\pi$ , it is updated to the  $\alpha - 2\pi$  value. In the previous version of Optipharm, this value was updated to the maximum value  $2\pi$ .

One of the strengths of Optipharm is the concept of annealing the search space by decreasing at each iteration of the algorithm the area where each solution can search, i.e., can perform reproduction and improvements operations. The radius's value associated with a solution search area depends on the iteration in which the solution was created and decreases with each iteration. So initially, a solution may explore the entire search space. However, in the later stages of the algorithm, it focuses on a very localized area of the space to refine the solutions. It allows the coexistence of a population of solutions with different radii that search on different sub-areas. This mix of coexisting solutions implies an equilibrium between exploitation and exploration of the search space. The radius associated to search areas decreases to a user-defined value called  $R_{t_{max}}$  and has this value in the last iteration, whose maximum number is also defined as parameter  $t_{max}$ .

## 2.3 Methodology

### Procedure for Rigid Molecules

As indicated in the previous section, with Optipharm, after an optimization process, the maximum similarity between a target and a query can be obtained. This optimization process is repeated for each query in the database. Then, after finalizing the process, the maximum similarity for molecule in the database



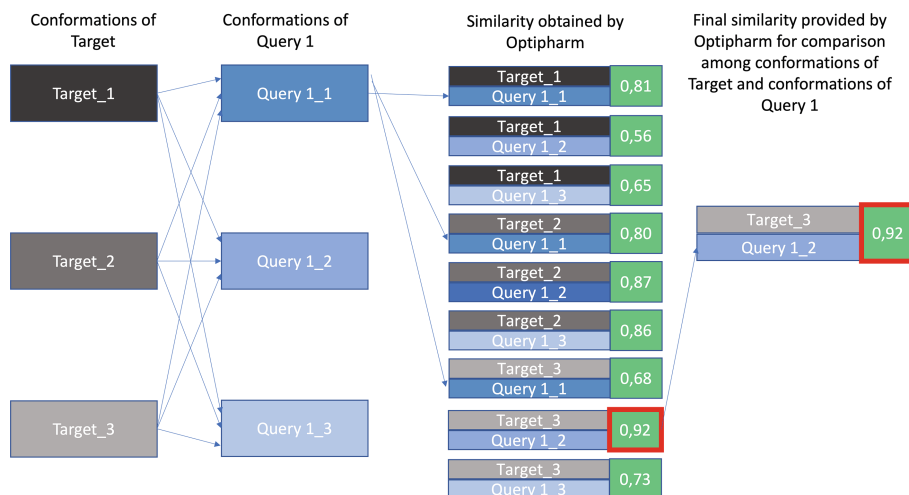
**Fig. 2.** Procedure to rank rigid molecules.

is known. Consequently, a ranked molecules list is obtained by sorting them according to their similarity. The first compounds are more likely to be successful potential drugs because they are the most similar to the target. Figure 2 shows this process to obtain a ranked list of compounds.

### Procedure for Molecules Conformations

When working with flexible molecules with some rotatable bonds, it is necessary to modify the methodology to obtain the similarity between a target and a query molecule in the database. Therefore, to simulate the flexibility at the rotatable bonds of a given molecule, multiple alternative conformers of this molecule have been constructed by modifying the rotatable bonds with various rotation angles. Once multiple conformations of both the target crystal or molecule and the query molecule have been obtained, a procedure is executed to obtain the maximum similarity of the two flexible compounds. Figure 3 shows such a procedure by means of a toy example where only three conformations have been generated for both the target and for the query.

As can be seen in the figure, an extensive comparison is carried out, which involves running  $ntnq$  times Optipharm algorithm instead of just one for rigid molecules. In this exhaustive comparison  $nt$  represents the number of conformations of the target molecule and  $nq$  is the number of conformations of the query molecule. Once the maximum similarity of each of the comparisons has been calculated, the algorithm searches for the highest value and provides it as the final similarity result between the two flexible molecules. In most cases, the obtained highest value is greater than the similarity value achieved when trading the molecules as rigid. In this toy example, ( $nt = nq = 3$ ) hence nine comparisons are made, and the similarity values are obtained. Finally, it can be



**Fig. 3.** Procedure for obtaining maximum similarity when working with conformations of molecules.

seen that the highest value (0.92) is obtained when conformation 3 of the target is compared to conformation 2 of the query, this value being, therefore, the one selected by the proposed method.

Once the flexible target has been compared with all molecules in the database, they are ordered based on their similarity value. This value, as previously mentioned, is usually higher than in the case of rigid molecules. This variation in the similarity values may imply that there will be a variation in the order of the compounds compared with the ordered list obtained when rigid molecules are considered. Consequently, new query compounds with a high similarity value can be identified while they are not detected when working with rigid molecules.

An example of how using the flexibility of the molecules yields better results than with rigid molecules (Fig. 2) is shown in Fig. 4. In this case, molecule 1, which was in second place when considered rigid, appears in the first place when it is considered flexible.

### 3 Materials

**Hardware.** The experiments of this work have been carried out using a cluster of 18x Bullx R424-E3: 2 Intel Xeon E5 2650 (16 cores), 64 GB of RAM memory and 128 GB SSD.

**Database DUD-E.** The DUD-E is a well-known benchmark for structure-based virtual screening methods from the Shoichet Lab at UCSF. The methodology of the DUD-E benchmark is fully described in its original work. In this

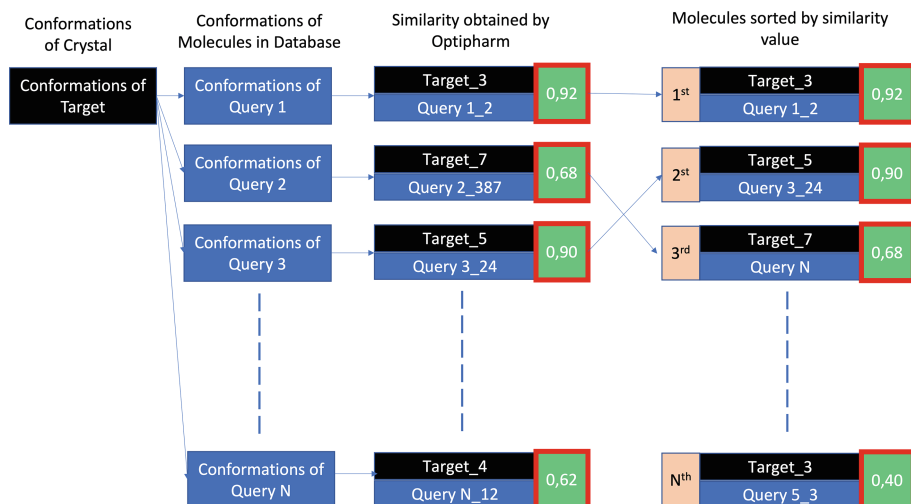


Fig. 4. Procedure to rank Molecules with conformations.

work, we only consider that the DUD-E database consists of 102 targets and 1.477.909 molecules that are the queries of our experiments.

The original DUD-E database downloaded from <http://dude.docking.org/> has been used in this work.

**Software.** The new version of Optipharm, described in Sect. 2.2 is the optimization algorithm used to find the maximum similarity between two compounds. It has been configurated to consider the hydrogen atoms of each molecule. In addition, all the heavy atom radii have been set to 1.7 Å. Furthermore, all compound pairs are centered and aligned. Consequently, the molecule centroids have been located at the coordinates center of the search space. Finally, each target molecule has been aligned so that its longest axis has been oriented at X-axis and the shortest along the Z-axis. The input parameter set used in Optipharm is the one proposed by the authors to make the algorithm reliable and robust. In particular, the following values were considered:  $N = 200000$  function evaluations,  $M = 5$  starting poses,  $t_{max} = 5$  iterations, and  $R = 1$  as the smallest possible radius.

Additionally, software OMEGA [13] has been the generator selected to obtain the conformations of targets and queries in the database. The maximum number of conformations for a given compound was limited to 500, though the obtained number was smaller in many compounds due to a small number of rotatable bonds.

## 4 Results

In this work, an experimental study has been performed with some targets of the DUD-E database. The results are shown for the two targets, FFPS and PTN1.

Both rigid and flexible molecules have been considered to establish a comparison of results.

Tables 1 and 2 show the results obtained with FPPS and PTN1 targets, respectively. These tables show in decreasing order of similarity the best 10 queries both when considering rigid and flexible molecules. In each table, on the left-hand side, the first three columns refer to the experiment results with rigid molecules. The *Query* column shows the name of the query, the *Tc* column shows the similarity value obtained, and the *Rk<sub>C</sub>* column shows the position of the compound in the ordered list obtained from the experiments with flexible molecules. The remaining four columns correspond to the experiments performed with flexibility. The *Target conformation* column identifies the conformation number that has obtained the best similarity value with the corresponding conformation of the query indicated in the *Query conformation* column. The *Tc* column indicates the similarity value obtained, and the *Rk<sub>R</sub>* column shows in which position that molecule is in the ordered list with the results of rigid molecules.

**Table 1.** Top-10 most similar compounds in shape to the target ZOL\_901\_1ZW5 (FPPS, Farnesyl diphosphate synthase).

Rigid			Flexible			
Query	<i>Tc</i>	<i>Rk<sub>C</sub></i>	Target conformation	Query conformation	<i>Tc</i>	<i>Rk<sub>R</sub></i>
CHEMBL301247	0.887	1	Target_7	CHEMBL301247_16	0.963	1
ZINC05368839	0.883	47	Target_5	CHEMBL924_3	0.963	175
CHEMBL299717	0.882	40	Target_10	CHEMBL446734_6	0.955	133
CHEMBL55358	0.881	28	Target_4	CHEMBL340034_4	0.952	837
CHEMBL434024	0.868	32	Target_6	CHEMBL394758_2	0.946	705
CHEMBL322551	0.866	42	Target_2	CHEMBL923_1	0.940	139
CHEMBL301065	0.863	96	Target_4	CHEMBL437758_11	0.939	18
CHEMBL99369	0.857	29	Target_9	CHEMBL394759_3	0.937	778
ZINC42040652	0.857	53	Target_2	CHEMBL55886_1	0.934	188
ZINC45202189	0.852	472	Target_3	CHEMBL98110_7	0.929	103

As seen in the tables, the similarity value obtained when flexibility is considered, as compared to the values obtained when only rigid molecules are taken into account. Thus, for example, in the Table 1 the similarity values are in the range [0.887–0.852] for rigid molecules and [0.963–0.929] for flexible molecules. This increment corroborates that it is more accurate to compare molecules considering them flexible.

This increase in similarity is not a fixed quantity for all the compounds. This variable increment implies that rigid molecules can modify their position and move from the latest positions in the list to the first positions when flexibility is taken into account.

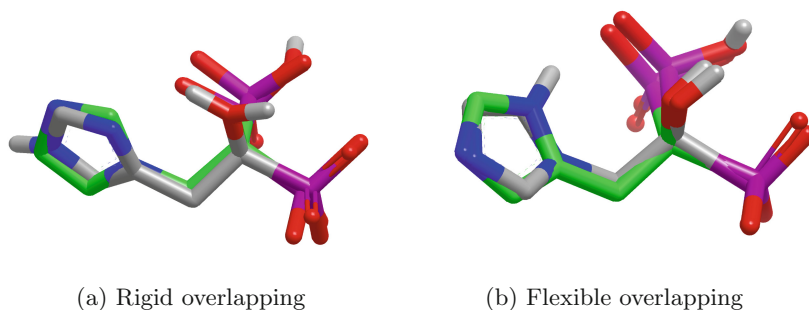
This fact can be verified by analyzing the two tables’ *Rk<sub>C</sub>* and *Rk<sub>R</sub>* columns. In the case of the *FPPS* target, it can be seen that the query that appears in

**Table 2.** Top-10 most similar compounds in shape to the target 982.301.2AZR (PTN1, Protein-tyrosine phosphatase 1B).

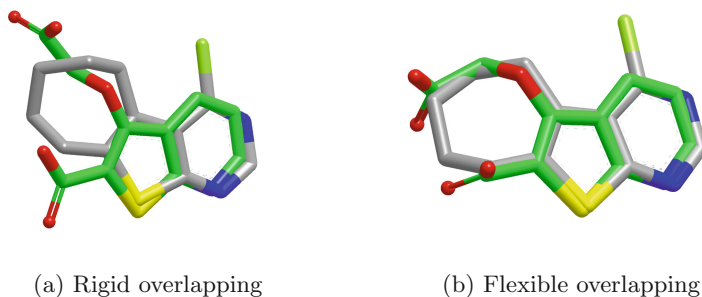
Rigid			Flexible			
Query	$T_c$	$Rk_C$	Target conformation	Query conformation	$T_c$	$Rk_R$
ZINC05945704	0.812	2	Target_12	ZINC03887548_2	0.899	15
ZINC39460069	0.811	7	Target_2	ZINC05945704_7	0.870	1
ZINC49581318	0.805	20	Target_12	ZINC62614886_9	0.863	103
ZINC50396795	0.802	205	Target_2	ZINC39612940_84	0.863	350
CHEMBL601290	0.797	29	Target_5	ZINC36909894_3	0.842	13
ZINC42396974	0.797	44	Target_8	ZINC49819538_3	0.840	8
ZINC44419884	0.794	10	Target_2	ZINC39460069_3	0.832	2
ZINC49819538	0.789	6	Target_2	ZINC39133313_11	0.832	282
ZINC16698354	0.788	28	Target_12	ZINC42708789_105	0.832	55
CHEMBL601290	0.782	37	Target_9	ZINC44419884_43	0.830	7

first place in the case of rigid molecules coincides with the one that appears in first place in the case of flexible molecules, although their similarity increases. However, for the rest of the queries, it can be seen that their position within the ordered list of flexible molecules is higher than 25. In particular, the tenth query (ZINC45202189) appears in position 472. On the other hand, if we analyze the first positions of the flexible molecules, we can see that most of them were in positions higher than 100, which implies that they would never be selected as drug candidates. Thus, for example, the compound appearing in second place (CHEMBL924\_3) in the flexible list occupies position 175 in the rigid list. These changes of positions in the respective lists can also be seen in Table 2, where the best query considering flexibility occupies position 15 in the list of rigid molecules.

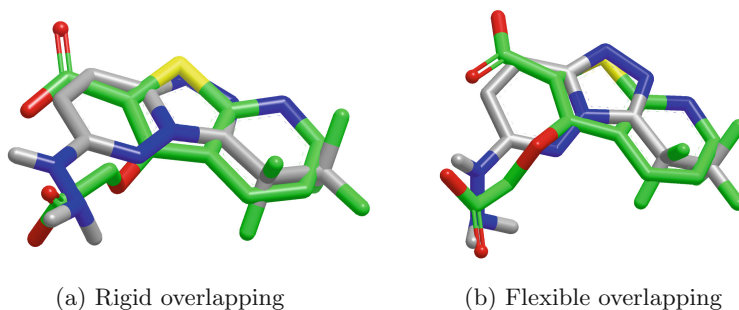
Figures 5 and 6 show the maximum similarity obtained respectively between the targets *FPPS* and *PTN1* and the queries appearing in the first position of the ordered lists. Thus, in the case of the target *FPPS*, where the best query obtained is the same (CHEMBL301247), whether rigid or flexible molecules are considered, it can be seen in Fig. 5 how the similarity in shape improves when the flexibility of the molecule is included. However, in the case of the target *PTN1*, the query with the highest similarity (0.812) when rigid molecules are considered is ZINC05945704. This query occupies the second position in the ordered list according to the similarity of flexible molecules. Figure 6 shows their overlapping with the Target. Regarding the flexible rank, conformation 2 of molecule ZINC03887548 has the best similarity value (0.899) though this molecule occupied position 15 when considered rigid. Figure 7 shows the overlapping of this query when it is considered rigid and flexible.



**Fig. 5.** Target *FPPS* or *ZOL.901.1ZW5* (in green) and Query *CHEMBL301247* are depicted in this Figure. Figure (a) shows the best overlapping between them when molecules are considered rigid while Figure (b) shows the overlapping after applying to them a conformational process as indicated in the first row of Table 1. (Color figure online)



**Fig. 6.** Target *PTN1* or *982.301.2AZR* (in green) and Query *ZINC03887548* are depicted in this Figure. Figure (a) shows the best overlapping between them when molecules are considered rigid while Figure (b) shows the overlapping after applying to them a conformational process as indicated in the first row of Table 2. (Color figure online)



**Fig. 7.** Target *PTN1* or *982.301.2AZR* (in green) and Query *ZINC05945704* are depicted in this Figure. Figure (a) shows the best overlapping between them when molecules are considered rigid while Figure (b) shows the overlapping after applying to them a conformational process as indicated in the second row of Table 2. (Color figure online)

## 5 Conclusion and Future Work

In this paper, we have addressed the LBVS similarity search problem, including flexibility. For this purpose, a new and more efficient version of OptiPharm has been presented. Among its improvements is the reduction of the optimization parameters by reducing the search space and a new freedom range in some of them. Consequently, the number of function evaluations required to find the optimal similarity is decreased. The experiments have been performed using the well-known DUD-E database, and flexibility has been implemented using OMEGA, where a maximum of 500 conformations was generated for each molecule.

The results obtained show an increase in the scoring function value for flexible molecules compared to the traditional rigid-molecule procedure. This better similarity optimization leads to new query compounds with a high similarity value that can be identified while not detected when working with rigid molecules.

As future work, it is proposed to implement a conformation generation algorithm as an internal procedure of OptiPharm, allowing the reduction of the experiments' time and offering a new solution to this complex LBVS problem.

**Acknowledgments.** This work was supported by the Spanish Ministry of Economy and Competitiveness through the CTQ2017-87974-R, RTI2018-095993-B-I00 and EQC2019-006418-P grants; by the Junta de Andalucía through the grant Proyectos de excelencia (P18-RT-1193), by the Programa Regional de Fomento de la Investigación (Plan de Actuación 2018, Región de Murcia, Spain) through the: "Ayudas a la realización de proyectos para el desarrollo de investigación científica y técnica por grupos competitivos (20988/PI/18)" grant; by the University of Almeria through the grant: Ayudas a proyectos de investigación I+D+I en el marco del Programa Operativo FEDER 2014-20" (UAL18-TIC-A020-B). Savíns Puertas Martín is a fellow of the 'Margarita Salas' grant (RR.A.2021.21), financed by the European Union (NextGenerationEU).

## References

1. Ain, Q.U., Aleksandrova, A., Roessler, F.D., Ballester, P.J.: Machine-learning scoring functions to improve structure-based binding affinity prediction and virtual screening. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **5**(6), 405–424 (2015)
2. Axenopoulos, A., Rafailidis, D., Papadopoulos, G., Houstis, E.N., Daras, P.: Similarity search of flexible 3D molecules combining local and global shape descriptors. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **13**(5), 954–970 (2016)
3. Cano-Muñoz, M., Jurado, S., Morel, B., Conejero-Lara, F.: Conformational flexibility of the conserved hydrophobic pocket of HIV-1 gp41. Implications for the discovery of small-molecule fusion inhibitors. *Int. J. Biol. Macromol.* **192**, 90–99 (2021)
4. Carlson, H.A.: Protein flexibility and drug design: how to hit a moving target. *Curr. Opin. Chem. Biol.* **6**(4), 447–452 (2002)
5. Debnath, S., Debnath, T., Majumdar, S., Arunasree, M.K., Aparna, V.: A combined pharmacophore modeling, 3D QSAR, virtual screening, molecular docking, and ADME studies to identify potential HDAC8 inhibitors. *Med. Chem. Res.* **25**(11), 2434–2450 (2016). <https://doi.org/10.1007/s00044-016-1652-5>



6. Ganesan, A., Coote, M.L., Barakat, K.: Molecular dynamics-driven drug discovery: leaping forward with confidence. *Drug Discov. Today* **22**(2), 249–269 (2017)
7. Ge, H., Wang, Y., Zhao, W., Lin, W., Yan, X., Xu, J.: Scaffold hopping of potential anti-tumor agents by WEGA: a shape-based approach. *Med. Chem. Commun.* **5**(6), 737–741 (2014)
8. Han, R., Zhang, F., Wan, X., Fernández, J.J., Sun, F., Liu, Z.: A marker-free automatic alignment method based on scale-invariant features. *J. Struct. Biol.* **186**(1), 167–180 (2014)
9. Hu, J., Liu, Z., Yu, D.J., Zhang, Y.: LS-align: an atom-level, flexible ligand structural alignment algorithm for high-throughput virtual screening. In: *Bioinformatics*, vol. 34, pp. 2209–2218. Oxford University Press (2018)
10. Kalászi, A., Szisz, D., Imre, G., Polgár, T.: Screen3D: a novel fully flexible high-throughput shape-similarity search method. *J. Chem. Inf. Model.* **54**(4), 1036–1049 (2014)
11. Leelananda, S.P., Lindert, S.: Computational methods in drug discovery. *Beilstein J. Org. Chem.* **12**, 2694–2718 (2016)
12. Lipinski, C.A.: Rule of five in 2015 and beyond: target and ligand structural limitations, ligand chemistry structure and drug discovery project decisions. *Adv. Drug Deliv. Rev.* **101**, 34–41 (2016)
13. OMEGA 4.1.0.2: OpenEye Scientific Software: Santa Fe, NM, USA (2019). <http://www.eyesopen.com>
14. Puertas-Martín, S., Redondo, J.L., Ortigosa, P.M., Pérez-Sánchez, H.: OptiPharm: an evolutionary algorithm to compare shape similarity. *Sci. Rep.* **9**(1), 1398 (2019)
15. ROCS: OpenEye Scientific Software: Santa Fe, NM. <http://www.eyesopen.com>
16. Rogers, D.J., Tanimoto, T.T.: A computer program for classifying plants. *Science* **132**(3434), 1115–1118 (1960)
17. Seidel, T., Bryant, S.D., Ibis, G., Poli, G., Langer, T.: 3D pharmacophore modeling techniques in computer-aided molecular design using LigandScout. Wiley (2017)
18. Selvaraj, C., et al.: Microsecond MD simulation and multiple-conformation virtual screening to identify potential anti-COVID-19 inhibitors against SARS-CoV-2 main protease. *Front. Chem.* **8**, 1–15, 595273 (2021). <https://doi.org/10.3389/fchem.2020.595273>
19. Tversky, A.: Features of similarity. *Psychol. Rev.* **84**(4), 327 (1977)
20. Yan, X., Li, J., Liu, Z., Zheng, M., Ge, H., Xu, J.: Enhancing molecular shape comparison by weighted gaussian functions. *J. Chem. Inf. Model.* **53**(8), 1967–1978 (2013)
21. Yan, X., Liao, C., Liu, Z., Hagler, A.T., Gu, Q., Xu, J.: Chemical structure similarity search for ligand-based virtual screening: methods and computational resources. *Curr. Drug Targets* **17**(14), 1580–1585 (2016)
22. Yuriev, E., Ramsland, P.A.: Latest developments in molecular docking: 2010–2011 in review. *J. Mol. Recogn.* **26**(5), 215–239 (2013)

# **Feature Selection, Extraction, and Data Mining in Bioinformatics: Approaches, Methods and Adaptations**



# Comparisons of Knowledge Graphs and Entity Extraction in Breast Cancer Subtyping Biomedical Text Analysis

Jean Davidson<sup>(✉)</sup>, Grif Hawblitzel, McClain Kressman, Andrew Doud,  
Harsha Lakshman Kumar, Ella Thomas, Paul Kim, Ava Jakusovszky,  
and Paul Anderson

Department of Computer Science and Department of Biological Sciences,  
California Polytechnic State University, San Luis Obispo, San Luis Obispo, CA, USA  
j david06@calpoly.edu

**Abstract.** In order to capitalize on the extensive biological research publications and databases, knowledge graphs can help extract clinically useful details from large and complicated resources. Here, we compare utility of knowledge graphs and named entity extraction for identifying biologically appropriate results from breast cancer subtyping publications. This biomedical field is an excellent representative test set - the biological mechanisms are well studied but complex, while the clinical applications of identifying breast cancer subtypes are critical to making appropriate diagnostic and therapeutic considerations. Optimizing knowledge graphs to extract actionable biological details rapidly and accurately could have huge implications in translating biological data into clinical care responses. Our research suggests that limitations exist in current knowledge graph pipelines in biomedical data analysis, primarily related to named entity extraction issues.

**Keywords:** Knowledge graphs · Breast cancer subtyping ·  
Reproducibility · Precision medicine

## 1 Introduction

The complexity of biological systems requires graphical representation in order to make any sense of the seemingly chaotic tangles of molecules and interactions. Metabolic pathways can grow to hundreds and thousands of intersecting proteins, carbohydrates and lipids - all simultaneously activating and interacting with partners to complete their tasks [9]. Graphs have become increasingly critical as biological datasets grow in complexity with the expansion of genomic, transcriptomic, and proteomic datasets capturing more and more biological phenomenon [16]. While broadly utilized in bioinformatics and biomedical research, only a subset can be considered knowledge graphs. We define knowledge graphs as a graph where nodes represent biological entities (e.g., gene, drug) and edges

represent relationships between the entities. While conflicting definitions exist, this definition is consistent with the predominant researchers in this field [24].

While knowledge graphs have been shown to be beneficial in many studies, they have not seen widespread adoption in the fields of biomedical sciences, the very topic which could most benefit from clear graph modeling of complex pathways, wherein one subtle shift could have profound patient outcomes [4]. The goal of this work is to study and assess the performance of state-of-the-art knowledge graph pipelines in biomedical research. To provide context for evaluation, we focus our analysis on the complex and heterogeneous disease of breast cancer where increased specificity of related knowledge allows for greater specificity of patient care [20]. We perform this analysis by analyzing the benefits of using a knowledge graph generated from relevant literature versus a pure manual analysis of the same body of literature.

## 2 Background

Breast cancer is a disease which currently affects over 2.3 million people per year worldwide and is responsible for over 600,000 deaths [21]. It is also a remarkably heterogeneous disease, with a variety of subtypes which have a wide range of prognostic outcomes [25]. These subtypes are defined by gene expression profiles, cellular histology, and tissue of origin. Currently the field describes five “intrinsic subtypes,” initially identified by the PAM50 subtype classifier: luminal A, luminal B, HER-2 enriched, basal-like (triple-negative), and normal [5]. It is clinically important to identify the subtype as early as possible in the diagnostic process in order to identify the most optimal treatment options to target the specific cellular characteristics of each subtypes. Luminal A and B tumors both grow from cells in the inner mammary ducts and similar gene expression profiles. In general, they have more promising prognoses, though luminal B tumors tend to be larger and more invasive than luminal A, resulting in a poorer clinical outcomes [11]. Basal-like tumors develop from the outer cells of the mammary duct and exhibit the poorest prognoses [8]. HER-2 enriched tumors are so-named for their abundance of HER-2 receptors on the cell surface, leading to their sensitivity to estrogen-blocking therapeutics. Their prognoses are often poorer than luminal subtypes and better than basal-like [3]. In general, subtype classification allows researchers to identify correlating differences both in disease etiology and clinical outcome; therefore, the discovery of more distinct subtypes provides a critical impetus for breast cancer research.

ScispaCy is a Python-based natural language processing (NLP) package for biomedical literature that is built off of the more general Python-based NLP package called spaCy [15]. The goal of scispaCy is to be a robust NLP library that provides the text-processing needs of the biomedical domain. The subset of the package we are focusing on is the “en\_ner\_bionlp13cg\_md” pre-trained model. En\_ner\_bionlp13cg\_md is a scispaCy model specifically designed for named entity

recognition (NER) on literature in the domain of cancer research. NER is a form of information extraction that involves extracting and classifying named entities in unstructured text. En\_ner\_bionlp13cg\_md largely uses a neural network model to classify named entities such as amino acid, cancer, tissue, gene, cell, disease, etc.

BERN2 is another NER model that uses a neural network and is specifically built for biomedical literature [22]. BERN2 was designed for any research paper that could be on Pubmed. BERN2 works by using cached annotations if possible, and if not uses a neural network NER model to find entities. The classes of named entities BERN2 can extract are “Disease”, “Chemical”, “Gene/Protein”, “Species”, “Cell Line”, “Cell Type”, “DNA”, “RNA”.

KGen is a pipeline built from state-of-the-art natural language processing libraries and publicly available ontologies. It is designed to generate knowledge graphs in a semi-automatic fashion [18]. It assists in graphically representing knowledge from the vast amount of scientific information available in articles. It does so by identifying triples for Resource Description Framework (RDF) graphs. A triple in an RDF graph consists of a subject, predicate, and object, which essentially describes a node, and the nature of its relation to another node. It can be used on unstructured text from biomedical papers and produces a ttl file, a file type designed to represent triples, which can then be queried by SPARQL - a semantic query language.

Knowing that KGen would be unable to create a concise, or accurate graph, the working goal for a partial output of a knowledge graph from a paper on breast cancer sub-typing would look similar to Fig. 1.

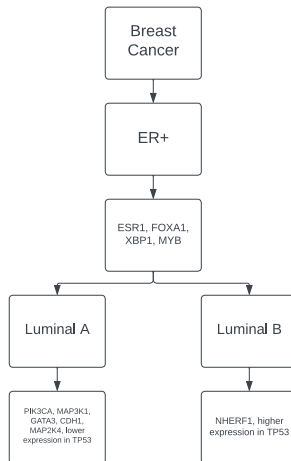


Fig. 1. Ideal knowledge graph output.

### 3 Methods

Individual biologists manually performed a systematic literature review to identify breast cancer subtyping genomic articles. Article title, key words, and abstracts were searched using Google Scholar using the following search condition: “breast cancer subtype molecular classification”. The resulting articles were filtered to only those containing primary genomic and transcriptomic data on breast cancer subtyping. These factors greatly influenced the article filtering due to the availability of these large data sets for farther work on cancer subtyping. We considered journal articles published prior to January 1, 2022. This resulted in a final set of 9 articles [1, 2, 2, 7, 10, 12, 13, 17, 19].

Each of the articles was reviewed by at least two biologists who prepared a summary of major genomic findings. When reviewing articles, the criteria to determine utility of the paper was dependent on the ability to use molecular markers for more accurate sub-typing of breast cancer. Key information gathered in the process was the factors for sub-typing, the number of sub-types and contrasting names for each sub-type. A meta-analysis was then performed on all the findings to identify open questions and gaps in the findings. For this study, we selected two of the questions for comparative analysis:

1. Blows *et al.* found that low TP53 mutation frequency in luminal A (12%) and a higher frequency in luminal B (29%) cancers [2]. Have other researchers found the interesting change in mutation rate in TP53 between luminal A and luminal B?
2. Koboldt *et al.* found a luminal expression signature of ESR1, GATA3, FOXA1, XBP1 and MYB [12]. Have other researchers found similar signatures?

Utilizing these papers, we applied our selected entity recognition technologies for comparison to manually curated information performed by biologists. Figures 2, 3 and 4 below show example annotations done on the abstract of the Koboldt *et al.* paper [12] by ScispaCy, BERN2, and manually extracted information.

#### 3.1 Knowledge Graph Pipeline

KGen is a pipeline of numerous NLP tools in order to generate the aforementioned triples. It begins with preprocessing in four major steps.

1. Split raw text into sentences
2. Co-reference Identification
3. Identify abbreviations
4. Simplify sentences to groups of phrases. (Noun phrase, phrase, etc.).

The first step of splitting into sentences is done by tokenizing the text and identifying end of sentence punctuation, in order to prepare for future processing.

We analysed primary breast cancers by genomic DNA CELLULAR\_COMPONENT copy number arrays, DNA CELLULAR\_COMPONENT methylation, exome sequencing, messenger RNA arrays, microRNA sequencing and reverse-phase protein arrays. Our ability to integrate information across platforms provided key insights into previously defined gene expression subtypes and demonstrated the existence of four main breast cancer CANCER classes when combining data from five platforms, each of which shows significant molecular heterogeneity. Somatic mutations in only three genes ( TP53 GENE\_OR\_GENE\_PRODUCT , PIK3CA GENE\_OR\_GENE\_PRODUCT and GATA3 GENE\_OR\_GENE\_PRODUCT ) occurred at >10% incidence across all breast cancers CANCER ; however, there were numerous subtype-associated and novel gene mutations including the enrichment of specific mutations in GATA3 GENE\_OR\_GENE\_PRODUCT , PIK3CA GENE\_OR\_GENE\_PRODUCT and MAP3K1 GENE\_OR\_GENE\_PRODUCT with the luminal A subtype. We identified two novel protein-expression-defined subgroups, possibly produced by stromal/microenvironmental SIMPLE\_CHEMICAL elements, and integrated analyses identified specific signalling pathways dominant in each molecular subtype including a HER2/phosphorylated HER2/EGFR/phosphorylated EGFR GENE\_OR\_GENE\_PRODUCT signature within the HER2-enriched CANCER expression subtype. Comparison of basal-like breast tumours CANCER with high-grade serous ovarian tumours CANCER showed many molecular commonalities, indicating a related aetiology and similar therapeutic opportunities. The biological finding of the four main breast cancer CANCER subtypes caused by different subsets of genetic and epigenetic abnormalities raises the hypothesis that much of the clinically observable plasticity and heterogeneity occurs within, and not across, these major biological subtypes of breast cancer CANCER .

Fig. 2. ScispaCy’s annotation of the Koboldt *et al.* paper [12]

● Cell type ● Gene/Protein ● Cell line ● RNA ● DNA ● Disease

We analysed primary breast cancers by genomic DNA copy number arrays, DNA methylation, exome sequencing, messenger RNA arrays, microRNA sequencing and reverse-phase protein arrays. Our ability to integrate information across platforms provided key insights into previously defined gene expression subtypes and demonstrated the existence of four main breast cancer classes when combining data from five platforms, each of which shows significant molecular heterogeneity. Somatic mutations in only three genes (TP53, PIK3CA and GATA3) occurred at >10% incidence across all breast cancers; however, there were numerous subtype-associated and novel gene mutations including the enrichment of specific mutations in GATA3, PIK3CA and MAP3K1 with the luminal A subtype. We identified two novel protein-expression-defined subgroups, possibly produced by stromal/microenvironmental elements, and integrated analyses identified specific signalling pathways dominant in each molecular subtype including a HER2/phosphorylated HER2/EGFR/phosphorylated EGFR signature within the HER2-enriched expression subtype. Comparison of basal-like breast tumours with high-grade serous ovarian tumours showed many molecular commonalities, indicating a related aetiology and similar therapeutic opportunities. The biological finding of the four main breast cancer subtypes caused by different subsets of genetic and epigenetic abnormalities raises the hypothesis that much of the clinically observable plasticity and heterogeneity occurs within, and not across, these major biological subtypes of breast cancer.

Fig. 3. BERN2’s annotation of the Koboldt *et al.* paper [12]

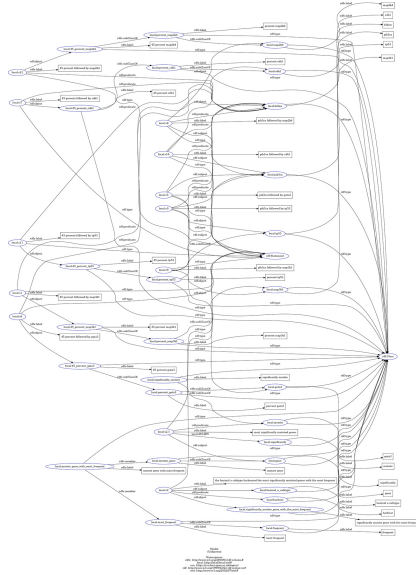
Co-reference identification looks to link objects to its implicit references nearby. For example, in a sentence that says, “this study, ... it ...” with it referring to this study, the program attempts to replace the it with “this study” so that it is explicitly related. It also identifies abbreviations by looking for the common pattern of “part of speech (POS)” with parenthesis to identify an abbreviation for later use. Finally it subdivides sentences into smaller phrases, such as noun phrases and verb phrases.

It then attempts to extract primary and secondary triples from the structures using semantic labelling and dependency parsing. An example is that a sentence of the form {NounPhrase<sub>1</sub>, VerbPhrase} with the verb phrase being of the structure {Verb, NounPhrase<sub>2</sub>}, would be processed into the rdf triple {NounPhrase<sub>1</sub>, verb, NounPhrase<sub>2</sub>}. Then, using entity recognition models, such as ScispaCy, it attempts to link with an ontology, and creates a graph using the triples as defining a node and edge pair. The result will look something like this (Fig. 5).

To evaluate the knowledge graphs, we browsed and developed SPARQL queries to extract partial and relevant information for these questions. In addition to running the queries on the entire knowledge graph derived from all







**Fig. 5.** KGen graph output for a single sentence.

en\_ner\_bionlp13cg\_md NER model and BERN2 were applied to the titles and abstracts to detect genes in the texts. The pieces of text the NER models labelled as genes were preprocessed by converting the text to lowercase and removing unnecessary whitespace, and punctuation. The detected genes were matched with genes existing in our 20,000 gene dataset and only the matching genes were used in the final dataset fed to the model. ScispaCy ended up finding 176 genes in common and BERN2 ended up finding 172.

SciKit Learn’s Multi-Layer Perceptron Classifier (MLPClassifier) was used to generate performance results for comparative analysis. The classifier uses a basic neural network with the default SciKit Learn parameters and architecture. We chose to use the MLPClassifier over more sophisticated classifiers due to its balance between a short training time and reasonable performance. In training and testing a random 80/20 train/test split was always used, and the final results were determined by averaging the performance metrics of 10 independent runs.

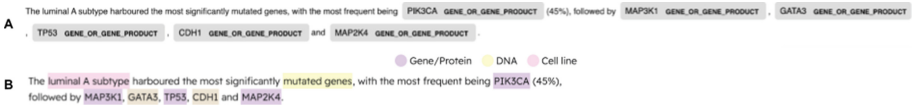
## 4 Results and Discussion

A qualitative self-evaluation of Question 1 indicates that the quality of the entities extracted is not sufficient for wide-scale deployment of the KGen pipeline. For example, when a graph made from the sentence “The luminal A subtype harboured the most significantly mutated genes, with the most frequent being PIK3CA (45%), followed by MAP3K1, GATA3, TP53, CDH1 and MAP2K4” from Blows *et al.* [2]. This sentence originates the idea in Question 1. This text

is easily interpreted and understood by biologists as an association of those genes with luminal A cancer. The graph generated for the sentence contains 0 links from a luminal A node to any of the genes of interest. For example a query for relations to the node representing gata3 returns only one link:

```
http://www.local/local.owl →
http://www.w3.org/2000/01/rdf\discretionary-schema#label → gata3.
```

Further analysis and browsing of the knowledge graph indicates that the entities are not extracted with the precision and recall necessary to execute the required queries. To explore this finding, we manually compared two leading entity extraction methods: ScispaCy and BERN2. The results of this analysis are shown in Fig. 6. We observe that both methods correctly identify genes and that BERN2 incorrectly classifies subtype, while ScispaCy omits it. We also observe a misidentification of gata3 and cdh1, and finally, we note that ScispaCy does not identify luminal A.



**Fig. 6.** Named entity extraction results for Question 2 shown for (A) ScispaCy and (B) BERN2.

Similar results are observed for Question 2. The sentence that originates this question is “One of the most dominant features is high mRNA and protein expression of the luminal expression signature, which contains ESR1, GATA3, FOXA1, XBP1 and MYB; the luminal/ER+ cluster also contained the largest number of significantly mutated genes.” For this sentence, our analysis showed that both ScispaCy and BERN2 identified the genes, BERN2 incorrectly classifies the subtype, ScispaCy omits the subtype, and ScispaCy misidentifies luminal and ER+ as genes.

ScispaCy and BERN2 were found to be effective at improving model performance through dimensionality reduction.

The major performance difference between the unprocessed dataset and the random 1000 dataset on the same classifier (see Table 1) highlights the difficulty the MLPClassifier has handling our dataset with 20,000 features. This also highlights the necessity of feature selection or dimensionality reduction on this dataset to ensure model performance. The highest performing dimensionality reduction method was PCA with 100 components with an f-score of 0.7798. ScispaCy and BERN2 were comparable to this with f-scores of 0.7616 and 0.7656 respectively, and performed a significant amount better than the random 1000 which had an f-score of 0.7175. While ScispaCy and BERN2 did not perform the best, it is arguable that they are still the best choice for within this set of methods because they still maintain the interpretability of the data.

**Table 1.** Comparing the performance of ScispaCy and BERN2 used for dimensionality reduction

	Precision	Recall	F-score
No preprocessing	0.0894	0.2275	0.1141
Random 1000	0.7673	0.7209	0.7175
PCA 100	0.7886	0.7747	0.7798
ScispaCy 178	0.7992	0.7550	0.7616
BERN2 172	0.8091	0.7577	0.7656

## 5 Conclusion

The potential advantages of knowledge graphs and NER in biomedicine include but are not limited to identifying important and relevant entities in papers, improving predictive performance of machine learning models, and predicting unknown relationships between entities. Our results indicate that connections between possible gene and disease relations, can to some extent, be identified quickly with both automated knowledge graph pipelines (e.g., KGen), and specifically, NER methods such as BERN2 and ScispaCy. Even in the presence of omitted and incorrectly labeled entities, the preprocessing done can with these tools can help a researcher identify targets with higher efficiency. None of the extraction methods tested in this paper were able to execute with a high degree of accuracy, potentially limiting their ability to discover connections across papers and predict unknown relationships between entities.

The domain of biomedical literature is broad and diverse making it difficult for general biomedical NER models to perform on subfields such as breast cancer subtyping. For example, there is ongoing research into defining molecular subtypes [23] of breast cancer. With the development of highly accurate entity extraction methods, the ability to settle the differing nomenclature behind what is the underlying same sub-type may be possible. The implications for this, especially in a field such as cancer sub-typing where the many groups working towards identifying cancer sub-type groups do not take into account prior classifications is immense. By centralizing the data behind molecular and clinical sub-typing, better treatment can be easily reached for patients with more suitable treatment [14].

Our results show that semi-automated knowledge graph pipelines such as KGen are limited in the triples they can identify extract and thus are limited in utility for complex biomedical domains such as breast cancer subtyping. Our analysis identified named entity extraction as a primary issue with biomedical knowledge graph creation and one with significant biomedical informatics research potential.



## References

1. Bastien, R.R., et al.: Pam50 breast cancer subtyping by RT-QPCR and concordance with standard clinical molecular markers. *BMC Med. Genom.* **5**(1), 1–12 (2012)
2. Blows, F.M., et al.: Subtyping of breast cancer by immunohistochemistry to investigate a relationship between subtype and short and long term survival: a collaborative analysis of data for 10,159 cases from 12 studies. *PLoS Med.* **7**(5), e1000279 (2010)
3. Burstein, H.J.: The distinctive nature of her2-positive breast cancers. *New Engl. J. Med.* **353**(16), 1652–1654 (2005)
4. Callahan, T.J., Tripodi, I.J., Pielke-Lombardo, H., Hunter, L.E.: Knowledge-based biomedical data science. *Ann. Rev. Biomed. Data Sci.* **3**, 23–41 (2020)
5. Chia, S.K., et al.: A 50-gene intrinsic subtype classifier for prognosis and prediction of benefit from adjuvant tamoxifen. *Clin. Cancer Res.* **18**(16), 4465–4472 (2012)
6. Curtis, C., et al.: The genomic and transcriptomic architecture of 2,000 breast tumors reveals novel subgroups. *Nature* **486**, 346–352 (2012)
7. Dai, X., et al.: Breast cancer intrinsic subtype classification, clinical use and future trends. *Am. J. Cancer Res.* **5**(10), 2929 (2015)
8. Fadare, O., Tavassoli, F.A.: Clinical and pathologic aspects of basal-like breast cancers. *Nat. Clin. Pract. Oncol.* **5**(3), 149–159 (2008)
9. Gunduz, C., Yener, B., Gultekin, S.H.: The cell graphs of cancer. *Bioinformatics* **20**(suppl\_1), i145–i151 (2004)
10. Horr, C., Buechler, S.A.: Breast cancer consensus subtypes: a system for subtyping breast cancer tumors based on gene expression. *NPJ Breast Cancer* **7**(1), 1–13 (2021)
11. Ignatiadis, M., Sotiriou, C.: Luminal breast cancer: from biology to treatment. *Nat. Rev. Clin. Oncol.* **10**(9), 494–506 (2013)
12. Koboldt, D., et al.: Comprehensive molecular portraits of human breast tumours. *Nature* **490**(7418), 61–70 (2012)
13. Lakis, S., et al.: The androgen receptor as a surrogate marker for molecular apocrine breast cancer subtyping. *The Breast* **23**(3), 234–243 (2014)
14. Liu, Z., Zhang, X.S., Zhang, S.: Breast tumor subgroups reveal diverse clinical prognostic power. *Sci. Rep.* **4**, 4002 (2014). <https://europepmc.org/articles/PMC5379255>
15. Neumann, M., King, D., Beltagy, I., Ammar, W.: Scispacy: Fast and robust models for biomedical natural language processing. arXiv preprint [arXiv:1902.07669](https://arxiv.org/abs/1902.07669) (2019)
16. Nicholson, D.N., Greene, C.S.: Constructing knowledge graphs and their biomedical applications. *Comput. Struct. Biotechnol. J.* **18**, 1414–1428 (2020)
17. Rontogianni, S., et al.: Proteomic profiling of extracellular vesicles allows for human breast cancer subtyping. *Commun. Biol.* **2**(1), 1–13 (2019)
18. Rossanez, A., Dos Reis, J.C., Torres, R.D.S., de Ribaupierre, H.: Kgen: a knowledge graph generator from biomedical scientific literature. *BMC Med. Inf. Decis. Mak.* **20**(4), 1–24 (2020)
19. Shibahara, T., et al.: Deep learning generates custom-made logistic regression models for explaining how breast cancer subtypes are classified. *bioRxiv* (2021)
20. Sims, A.H., Howell, A., Howell, S.J., Clarke, R.B.: Origins of breast cancer subtypes and therapeutic implications. *Nat. Clin. Pract. Oncol.* **4**(9), 516–525 (2007)
21. Sung, H., et al.: Global cancer statistics 2020: globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA Cancer J. Clin.* **71**(3), 209–249 (2021)

22. Sung, M., Jeong, M., Choi, Y., Kim, D., Lee, J., Kang, J.: Bern2: an advanced neural biomedical named entity recognition and normalization tool. arXiv preprint [arXiv:2201.02080](https://arxiv.org/abs/2201.02080) (2022)
23. Troester, M.A., Swift-Scanlan, T.: Challenges in studying the etiology of breast cancer subtypes. *Breast Cancer Res. BCR* **11**(3), 104 (2009). <https://europepmc.org/articles/PMC2716506>
24. Yan, J., Wang, C., Cheng, W., Gao, M., Zhou, A.: A retrospective of knowledge graphs. *Front. Comput. Sci.* **12**(1), 55–74 (2018). <https://doi.org/10.1007/s11704-016-5228-9>
25. Zardavas, D., Irrthum, A., Swanton, C., Piccart, M.: Clinical management of breast cancer heterogeneity. *Nat. Rev. Clin. Oncol.* **12**(7), 381–394 (2015)



# Towards XAI: Interpretable Shallow Neural Network Used to Model HCP's fMRI Motor Paradigm Data

José Diogo Marques dos Santos<sup>1,2</sup>  and José Paulo Marques dos Santos<sup>3,4,5</sup> 

<sup>1</sup> Faculty of Engineering, University of Porto, R. Dr Roberto Frias, 4200-465 Porto, Portugal

<sup>2</sup> Abel Salazar Biomedical Sciences Institute, University of Porto, R. Jorge de Viterbo Ferreira, 4050-313 Porto, Portugal

<sup>3</sup> University of Maia, Av. Carlos de Oliveira Campos, 4475-690 Maia, Portugal  
jpsantos@umaia.pt

<sup>4</sup> LIACC - Artificial Intelligence and Computer Science Laboratory,  
University of Porto, R. Dr Roberto Frias, 4200-465 Porto, Portugal

<sup>5</sup> Unit of Experimental Biology, Faculty of Medicine, University of Porto,  
Alameda Prof. Hernâni Monteiro, 4200-319 Porto, Portugal

**Abstract.** Under the concept of explainable artificial intelligence (XAI), this study explores the usage of shallow neural networks (SNN) to model and predict motor processes in the brain. Two main goals are considered: the suitability of independent component analysis (ICA) for data dimension reduction; and the capability of the SNN to have its underlying processes explained while retaining accurate predictions.

Thirty subjects from the HCP Young Adult database are used. A traditional GLM-based data analysis is carried out aiming to establish a ground for comparison, besides founded neuroscientific knowledge. ICA is used for input data dimensionality reduction, which feeds an SNN with one hidden layer containing 10 nodes. Accuracies range from 67.5% to 92.5%, and precisions from 64.3% to 97.2%, per stimulus. The analysis of the weights yields independent components (ICs), i.e. inputs, that encompass motor areas. Even though the ICs' spatial resolution is not optimal, the SNN predicts well above the chance level.

The motor cortex-containing ICs, i.e. the main inputs, are in accordance with the founded neuroscientific knowledge and the GLM-based data analysis results, allowing for the interpretability of the SNN underlying processes.

**Keywords:** Explainable artificial intelligence (XAI) · Shallow neural networks · Backpropagation feedforward artificial neural networks · Human Connectome project · fMRI

## 1 Introduction

Although the success of deep learning methods, their “black box” nature is not transparent in what concerns how they achieve predictions [1]. The process is not comprehensible

and, therefore, is not under control. Such drawback precludes the extensive use of deep learning in the health / medical domains due to the lack of reliability [2]. The purpose of the present study is to contribute to the emerging field of explainable artificial intelligence (XAI) [3], more specifically, to develop interpretable machine learning procedures that help understand how the brain functions.

fMRI (functional magnetic resonance imaging) is a neuroscientific technique widely used to study brain functioning. The Human Connectome Project “is undertaking a systematic effort to map macroscopic human brain circuits and their relationship to behaviour in a large population of healthy adults.” [4] Further progress [5, 6] has freely disclosed large brain function-related datasets. One encompasses fMRI data acquired in a simple motor paradigm where subjects move their feet, hands, and tongue. Contrary to cognitive processes, motor processes in the brain are reasonably known. The motor and somatosensorial cortices map specific body parts. Such specificity may be used to define targets for classification. Therefore, one may have both ends of the process, inputs and outputs, which may contribute to exploring and discovering the underlying machine learning processes. Thus, the motor fMRI Human Connectome Project dataset offers an appropriate platform for testing and studying machine learning classifiers and understanding how they function.

ANNs (artificial neural networks) are not new for fMRI data modelling and analysis [5, 6]. However, because fMRI data is of high dimensionality, the studies have focused on parts of the brain or modelling the BOLD (blood-oxygen-level-dependent) signal. The alternative for whole-brain fMRI acquisitions passes by data dimensionality reduction [7, 8]. This step is of crucial importance as the yielded data must retain the information needed for accurate predictions and, in this case, make it possible to interpret both neural networks, the biological and the artificial. Because ICA (independent component analysis) is widely used in fMRI [9], each IC (independent component) has a spatial expression that permits anatomical comparisons, and each IC retains temporal information that may be used for feature extraction for feeding the ANN, it is used in the study for data dimensionality reduction.

Thus, the present article aims to contribute with answers to the questions:

- is ICA suitable for data dimension reduction for the purpose of artificial neural networks modelling and prediction?
- is the simplicity of shallow neural networks helpful in interpreting the underlying processes yet achieving high prediction accuracies?

## 2 Method

### 2.1 Data Source: The Motor Paradigm in the Human Connectome Project

The data used were the 30 subjects in the HCP (Human Connectome Project) Young Adult database<sup>1</sup>, subjects 100307 to 124422, from the 100 Unrelated Subjects subset [4, 10, 11]. HCP’s motor paradigm is adapted from [12, 13]. It consists of two runs per subject. Each run encompasses a sequence of five stimuli where subjects were asked to

<sup>1</sup> <https://www.humanconnectome.org/study/hcp-young-adult>.

squeeze their left foot (LF), tap their left-hand fingers (LH), squeeze their right foot (RF), tap their right-hand fingers (RH), and move their tongue (T). The 12 s stimulus response time is preceded by a 3-s cue, both visually projected. There are also three periods with a fixation cross (FIX) lasting 15 s. Besides the stimuli sequences themselves, the difference between both runs was the fMRI scanner's phase encoding: one was acquired with right-to-left phase encoding (RL), and the other run with left-to-right (LR):

- RL sequence: FIX-RH-LF-T-RF-LH-FIX-T-LF-RH-FIX-LH-RF-FIX;
- LR sequence: FIX-LH-RF-FIX-T-LF-RH-FIX-LH-T-RF-RH-LF-FIX.

Subjects made no responses in this task. The TR is set to 0.72, and the run duration is 3:34. Each subject originated 284 volumes in each fMRI session, as the first fixation cross was cut in order to synchronise the data files with the other run for the same subject and between subjects. Data files were already subjected to brain extraction and registered to a standard image.

## 2.2 Neural Data Processing

The data originated in the fMRI scanning sessions is analysed in two parallel ways: on the one hand, it is a traditional GLM-based analysis considering the a priori model defined by the explanatory variables (EVs); on the other hand, the data is ultimately analysed by the mean of a backpropagation feedforward shallow neural network, but it is previously pre-processed using ICA (Independent Component Analysis). The purpose of the ICA pre-processing is to reduce data dimensionality.

**GLM-based Data Analysis.** fMRI data pre-processing is carried out with FSL (FMRIB's Software Library) v. 6.0.5<sup>2</sup> and the specific GLM-based analysis with FEAT (fMRI Expert Analysis Tool) [14].

At the subject level, each session is analysed separately, and the following methods are applied: registration to high resolution structural and standard space images is carried out using FLIRT [15, 16]; motion correction using MCFLIRT [16]; non-brain removal using BET [17]; spatial smoothing using a Gaussian kernel of FWHM 5mm; grand-mean intensity normalisation of the entire 4D dataset by a single multiplicative factor; high pass temporal filtering (Gaussian-weighted least-squares straight-line fitting, with  $\sigma = 45.0s$ ); time-series statistical analysis is carried out using FILM with local autocorrelation correction [18];  $z$  (Gaussianised T/F) statistic images is thresholded using clusters determined by  $z > 2.3$  and a (corrected) cluster significance threshold of  $p = 0.05$  [19]. This analysis considers six EVs: LF, LH, RF, RH, T, and cue. Finally, the subject level analysis has a second step where the results of both sessions are combined, which is done considering fixed effects.

The thirty individual level contrasts outputs are then analysed at the group level. In this stage, the mean across all subjects is calculated for all EVs, considering mixed-effects (FLAME stage 1; FMRIB's Local Analysis of Mixed Effects) [20, 21]. The statistical parameter maps are thresholded for  $z > 2.3$ .

<sup>2</sup> <https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FSL>.



**Data Reduction with ICA and Pre-processing.** The 30 subjects are randomly split into two groups: 20 are assigned into the train group and the remaining 10 into the test group. Because the stimuli sequence is different in each session and in order to concatenate the data files, each session is interpreted as a separate subject, i.e., for the data reduction with ICA and pre-processing, each data file is considered one subject, and, therefore, there are 40 data files in the train group, and 20 in the test group. It is important to stress that each subject keeps in the respective group, i.e., there is no data of the same subject in both groups.

The 40 fMRI data files of the train group entered the ICA analysis as implemented in MELODIC (Multivariate Exploratory Linear Optimized Decomposition into Independent Components) v. 3.15 [9], also part of FSL. Importantly, to overcome the limitations imposed by using MIGP (MELODIC's Incremental Group-PCA) in v. 3.15, which precludes the output of ICs' complete time-courses, this step is run in line command including the option `--disableMigp`. The following data pre-processing was applied to the input data: masking of non-brain voxels; voxel-wise de-meaning of the data; normalisation of the voxel-wise variance. Pre-processed data were whitened and projected into a 46-dimensional subspace using probabilistic Principal Component Analysis. The number of dimensions was estimated using the Laplace approximation to the Bayesian evidence of the model order [9, 22]. The whitened observations were decomposed into sets of vectors that describe signal variation across the temporal domain (time-courses), the session/subject domain and across the spatial domain (maps) by optimising for non-Gaussian spatial source distributions using a fixed-point iteration technique [23]. Estimated component maps were divided by the standard deviation of the residual noise and thresholded by fitting a mixture model to the histogram of intensity values [9].

Features are then extracted from each of the 46-time courses of the ICs (independent components). The strategy adopted is to average the seventh, eighth and ninth signals after the stimulus onset. The mean time difference to the stimulus onset is 5.285 s, proximal to the maximum of the canonical hemodynamic response in the brain [24], i.e., this feature maximizes the difference between task activation and the baseline. The data is standardised. At the end of this stage, the result is a matrix with 400 rows (20 subjects  $\times$  2 sessions  $\times$  5  $\times$  2 stimulus/session), each corresponding to an epoch and 46 columns corresponding to one IC. This matrix is the training set input.

The test data is obtained with a different procedure. The 46 brain activation maps obtained with the train group are used as masks to average the individual time courses in the raw NIfTI files of each subject pertaining to the test group. The same procedure for feature calculation is adopted. The seventh, eighth and ninth acquisitions after stimulus onset are averaged. The data is standardised. Finally, a similar matrix with 200 rows (10 subjects  $\times$  2 sessions  $\times$  5  $\times$  2 stimulus/session) and 46 columns is obtained. This matrix is the test set input.

### 2.3 Implementation of the Shallow Neural Network

The AMORE package v. 0.2–15 [25] implemented in R<sup>3</sup> v. 4.1.2 [26] and RStudio<sup>4</sup> v. 2021.09.01 Build 372 is used to design and perform the necessary calculations of the backpropagation feedforward shallow neural network.

Exploratory analysis searched for the best tuning parameters. Firstly, the global learning rate varied from 0.05 to 0.50 in 0.05 steps and then, more finely, from 0.03 to 0.10 in 0.01 steps, and the momentum ranged from 0.3 to 0.9 in 0.1 steps. The best combination yields a global learning rate of 0.10 and a global momentum of 0.8. Because the purpose of the study is to deliver interpretable shallow neural networks, it is considered a single hidden layer with 10 nodes fully connected with the inputs (46) and outputs (5). The selected activation function for the hidden nodes is “tansig”, while “sigmoid” is for output neurons.

### 2.4 Neural Network Interpretation

To aid in the neural network interpretation, especially to understand which inputs have a higher impact on the correct hits, the “path weight” is calculated according to:

$$path\ weight_{ijk} = |w_{I_i H_j} \times w_{H_j O_k}| \quad (1)$$

where  $w_{I_i H_j}$  is the weight between the input node  $I_i$  and the hidden node  $H_j$ , and  $w_{H_j O_k}$  is the weight between the hidden node  $H_j$  and the output node  $O_k$ . Therefore,  $path\ weight_{ijk}$  is the module of the product of the weights found in the path from input  $I_i$  to output  $O_k$ , passing by the hidden node  $H_j$ . The analysis of the path weights aims to identify which magnitudes are further from zero. These “heavier” path weights contribute more to the perceptron equation than underweighted paths, close to zero, although signal magnitude also has a role here. Therefore, such path weights may identify the inputs (ICs) that hold important information for correct predictions.

The test data is used to evaluate the neural network’s performance depleted of its “lighter” weights. This is done twice, with the top 10 path weights per output and 46 (corresponding to the top 10% “heavier”). Finally, the inputs of the top 10 are compared to the contrasts that result from the GLM-based analysis and interpreted according to the neuroscience literature.

### 2.5 Neural Network and Procedure Quality Analyses

To identify possible biases in the neural network’s structure, the network is 10,000 times fed with two data sets: random values from a uniform distribution ranging two standard deviations (above e below) from the test data mean; random values from a normal distribution with the same mean and standard variation of the test data mean (as data is previously standardised,  $\mu = 0$ ,  $sd = 1$ ).

The neural network’s train and test input data are obtained by different procedures. While the train inputs derive from the process of data reduction with ICA, the test

<sup>3</sup> <https://www.r-project.org/>.

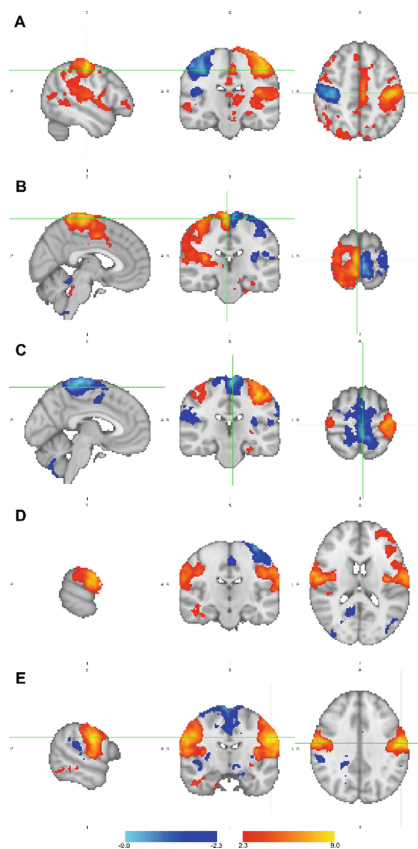
<sup>4</sup> <https://www.rstudio.com/>.

inputs result from averaging the raw data files screened with masks of the activations in each IC's statistical parameter map. Because such difference may influence the testing stage and have a magnitude of such impact, further analysis is done with the train data processed the same way as the test data, i.e., the train raw files are screened with masks obtained from the activated voxels in each IC, and the surviving voxels' time courses are averaged.

### 3 Results

#### 3.1 GLM-Based Analysis

Figure 1 depicts the activations that resulted from the GLM-based data analysis for the five types of stimuli contrasted among them. All the cases exhibit extensive activations in



**Fig. 1.** Selected sagittal, coronal, and axial views of the main statistical parameter maps in the GLM analysis. A: RH > LH ( $x = -50, y = -20, z = 48$ ); B: LF > RF ( $x = 4, y = -24, z = 70$ ); C: RH > RF ( $x = -4, y = -24, z = 64$ ); D: T > RH ( $x = -66, y = -18, z = 16$ ); E: T > LF ( $x = -58, y = -10, z = 34$ ). MNI152 coordinates. Radiological convention: right hemisphere on the left.

the motor cortex and cerebellum. In panel A, which contrasts right hand > left hand, the coronal and axial slices show the activation of the right hand in the medial section of the contralateral motor cortex and the deactivation of the left hand in the right motor cortex, also in its medial section. The two feet are contrasted in panel B, left foot > right foot, and both are contralaterally represented in the dorsal motor cortex in the interhemispheric fissure. Panel C addresses the right member, contrasting right hand > right foot; the activation is the medial motor cortex, which corresponds to hand, and the deactivation is in the interhemispheric fissure section of the motor cortex, both contralateral. Panels D and E contrast tongue > right hand and tongue > left foot, respectively. Hand and foot deactivations are in the expected places, as addressed in the previous panels, and the activation corresponding to the tongue appears in both hemispheres in the ventral section of the motor cortex, bordering the Sylvian fissure.

### 3.2 Performance of the Neural Network

The results of the shallow neural network with the best performance (more global correct hits = 166) are represented in Table 1, with the respective accuracies (global and partial) and precisions. Partial accuracies start at 67.5% for the left foot and increase to 77.5% for the right foot. The accuracies are higher for hand movements, 90.0% for the left and 92.5% for the right. For tongue, the accuracy is similar, 87.5%. The global accuracy is 83.0%. All of them are well above the chance level. Precision values are similar to accuracies. The left side has 63 correct hits, while the right side has 68, both out of 80. Feet has 58 correct hits versus 73 in hands (out of 80).

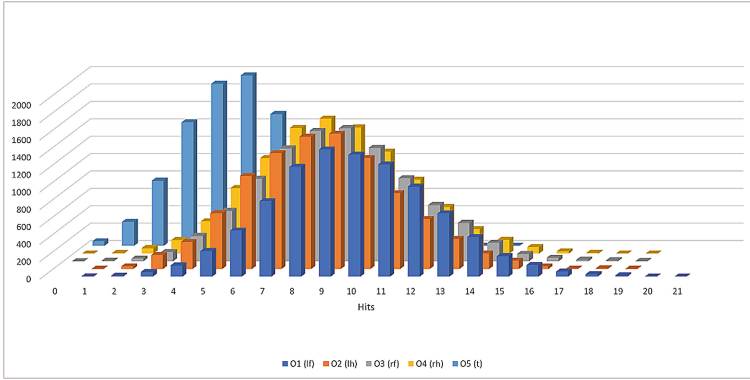
### 3.3 Procedure Quality Analysis

The first part consists in feeding the neural network with random values from a uniform and normal distributions (cf. Sect. 2.4). Figures 2 and 3 respectively represent the hits

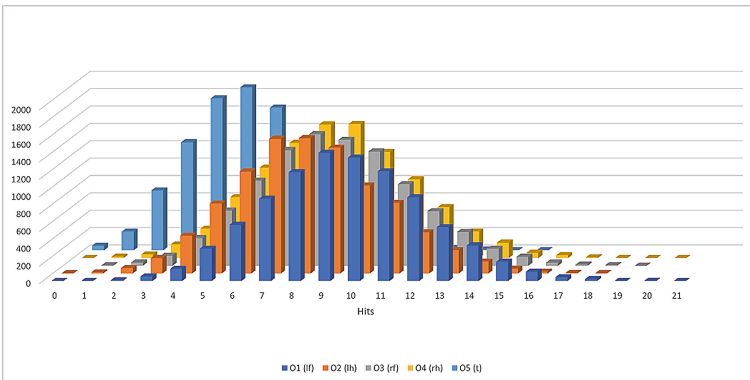
**Table 1.** Confusion matrix of the predictions of the neural network based on the test data, including the partial and global accuracies and precisions (LF: left foot; LH: left hand; RF: right foot; RH: right hand; T: tongue).

Stimulus		Prediction					Total
		LF	LH	RF	RH	T	
Input	LF	27	1	6	5	1	40
	LH	3	36	0	1	0	40
	RF	8	0	31	1	0	40
	RH	0	2	1	37	0	40
	T	4	0	1	0	35	40
Total		42	39	39	44	36	
Accuracy (%)		67.5	90.0	77.5	92.5	87.5	83.0
Precision (%)		64.3	92.3	79.5	84.1	97.2	

rate for each type of stimulus in frequency bar graphs. Compared with the values in the matrix diagonal in Table 1, both the peaks in Figs. 2 and 3 are well below (between 5 and 10 in the graphs, and between 27 and 37 in the table), which means that the structure of the network does not introduce biases that could inflate the hit rate.



**Fig. 2.** Bar graph representing the frequency for the five outputs (LF: left foot; LH: left hand; RF: right foot; RH: right hand; T: tongue) after feeding the neural network with random values from a uniform distribution (10,000 times).



**Fig. 3.** Bar graph representing the frequency for the five outputs (LF: left foot; LH: left hand; RF: right foot; RH: right hand; T: tongue) after feeding the neural network with random values from a normal distribution (10,000 times).

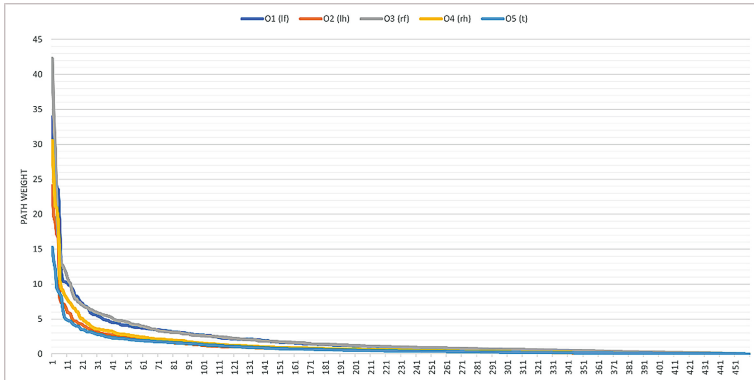
In the second part of the analysis of the quality of the procedure, the neural network is fed with the train data processed the same way as the test data. Table 2 reports the results, i.e. the hit rate for each stimulus, in the matrix diagonal, and both global and partial accuracies and precision. Except for left hand (accuracy) and tongue (precision), all the other values are higher in Table 2, although slightly.

### 3.4 Identification of the Inputs that Most Contribute to Correct Hits

The 460 path weights (46 inputs × 10 hidden nodes) for each stimulus (output) are depicted in Fig. 4 sorted in decreasing order. Visually, it is evident that about 10% of the path weights are “heavier”. Two analyses are run, one considering the 46 “heavier” path weights (which is not reported here for the sake of space) and the other regarding the top 10 “heavier” path weights.

**Table 2.** Confusion matrix of the predictions of the neural network based on the test with train data processed the same as the test data, including the partial and global accuracies and precisions (LF: left foot; LH: left hand; RF: right foot; RH: right hand; T: tongue).

Stimulus		Prediction					Total
		LF	LH	RF	RH	T	
Input	LF	62	1	12	2	2	79
	LH	5	69	1	4	1	80
	RF	11	0	68	1	0	80
	RH	0	3	2	75	0	80
	T	4	0	2	0	74	80
Total		82	73	85	82	77	
Accuracy (%)		78.5	86.2	85.0	93.8	92.5	87.2
Precision (%)		75.6	94.5	80.0	91.5	96.1	



**Fig. 4.** Path weights for each output (LF: left foot; LH: left hand; RF: right foot; RH: right hand; T: tongue) sorted in decreasing order.

The results of the neural network test depleted from all the “lighter” path weights, i.e. considering the top 10 paths per output only, are reported in Table 3. Compared with Table 1, with the complete set of weights, the global accuracy drops from 83.0% to 64.0%. Acknowledging that the chance level is 20%, this top 10 frugal neural network

still predicts well above. The stimuli that mostly drops are feet, the left dropping from 67.5% to 47.5% and the right from 77.5% to 42.5%. Nonetheless, the drop in accuracy is of lesser magnitude in hands: left drops from 90.0% to 77.5%, and right drops from 92.5% to 90.0%, the latter retaining high accuracy, though. The tongue is similar to feet, and the analysis with precision is in the same line.

The reduced neural network, containing the top 10 path weights, is further explored. Table 4 presents the ICs that belong to the “heavier” path weights. Remarkably, these ICs compose a restricted group. ICs 5, 7, 11, and 12 figure in all the paths, although with different influences. ICs 43 and 44 also figure for the left hand, right foot, and right hand, but with lesser weights.

**Table 3.** Confusion matrix of the predictions of the top 10 path weights per stimulus (output) of the neural network, including the partial and global accuracies and precisions (LF: left foot; LH: left hand; RF: right foot; RH: right hand; T: tongue).

Stimulus		Prediction					Total
		LF	LH	rf	RH	T	
Input	LF	19	1	9	2	9	40
	LH	1	31	4	4	0	40
	RF	5	0	17	8	10	40
	RH	0	1	2	36	1	40
	T	7	1	3	4	25	40
Total		32	34	35	54	45	
Accuracy (%)		47.5	77.5	42.5	90.0	62.5	64.0
Precision (%)		59.4	91.2	48.6	66.7	55.6	

**Table 4.** Identification of the ICs (inputs) that influence more in the frugal top 10 path weights neural network (LF: left foot; LH: left hand; RF: right foot; RH: right hand; T: tongue).

Output	ICs	Path weight sum	Output	ICs	Path weight sum
#1 LF	5	28.5290	#2 LH	5	14.3806
	7	58.9993		7	41.7794
	11	64.3555		11	38.5726
	12	44.3519		12	24.0543
		43		7.2116	
		44		6.6728	

(continued)

**Table 4.** (continued)

Output	ICs	Path weight sum	Output	ICs	Path weight sum
#3 RF	5	34.3084	#4 RH	5	20.0228
	7	44.7067		7	40.1865
	11	63.8054		11	54.1943
	12	42.3072		12	30.5627
	43	12.6839		43	9.1628
	44	11.7363		44	8.4783
#5 T	5	28.4708			
	7	21.1036			
	11	27.2684			
	12	15.8641			

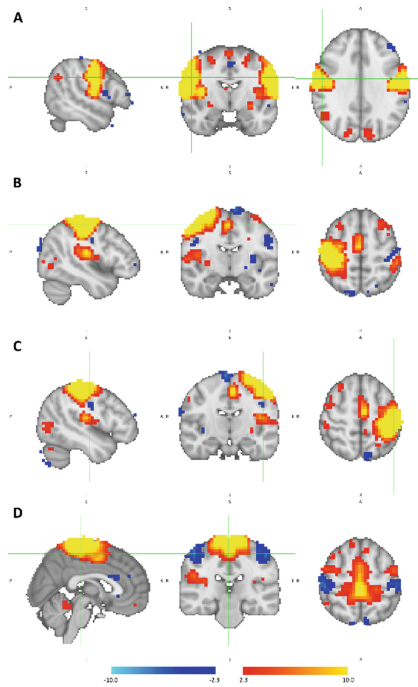
ICs 5, 7, 11, and 12 are depicted in Fig. 5. Most of the activations of these ICs are located in the precentral and postcentral gyri, which means motor and primary somatosensorial cortices. Such areas border the central sulci (all four cases) and the interhemispheric fissure (ICs 7, 11, and 12, mainly the latter).

## 4 Discussion

As reported in Table 1, the shallow neural network is predicting well above the chance level. This means that, although its frugal structure, containing one hidden layer with 10 nodes only, it extracts information from data to make predictions correctly. The mean accuracy is 83.0%, and precision ranges from 64.3% to 97.2%. These results favourably sanction all the procedures herein applied, including feature choice, the strategy for data reduction, the process of test data, and the construction of the shallow neural network and parameter tuning.

Besides predictions, the analysis of the neural network weights permits the identification of the inputs and paths into the outputs that have a greater influence on the performance. Because all the stimuli are motor-based, one would expect that the inputs that convey information of the motor-related areas in the brain would be prominent. The results support such an assumption, which is the core of the present research. Table 4 presents the inputs with higher weights in the frugal shallow neural network, i.e. the network constructed with the top 10 “heavier” path weights only. These ICs (inputs) contain motor and somatosensorial activations, as depicted in Fig. 5, bordering the central sulci and the medial part of the interhemispheric fissure. While IC 5 and IC 12 contain symmetrical activations in the motor cortex, IC 7 and IC 11 are lateralised, right and left, respectively. The motor cortex has the particularity of mapping the human body, i.e. specific sections of the motor cortex represent parts of the human body in an ordered manner. The feet are contralaterally mapped in the sections facing the interhemispheric fissure. The hands and fingers are contralaterally mapped in the medial section facing





**Fig. 5.** Selected sagittal, coronal, and axial views of the main ICs in Table 4. A: IC 5 depicted in the plans  $x = 54$ ,  $y = -6$ ,  $z = 32$ ; B: IC 7 depicted in the plans  $x = 46$ ,  $y = -10$ ,  $z = 56$ ; C: IC 11 depicted in the plans  $x = -46$ ,  $y = -14$ ,  $z = 56$ ; D: IC 12 depicted in the plans  $x = 2$ ,  $y = -26$ ,  $z = 56$ . MNI152 coordinates. Radiological convention: right hemisphere on left.

the central sulcus, and the tongue is bilaterally mapped in the ventral section, close to the Sylvian fissure [13, 27]. The results of the GLM-based analysis are consistent with these rules (cf. Fig. 1). Thus, one may conclude that there is coherence between the inputs, where the shallow neural network did the calculus to extract information to model the process and output correct predictions, and the neuroscientific knowledge, which stands that specific sections of the motor cortex participate in muscular movement processes, i.e. the output. Such coherence is equivalent to saying that the shallow neural network starts to be explainable.

Although there is a macro explanation, this procedure lacks detail due to the large activation blobs in the ICs (cf. Fig. 5). The neuroscientific knowledge is much more detailed in what concerns the organisation of the motor cortex than the features yielded by the ICA (independent component analysis) method. Probably, if the data reduction process could be more fine-grained, the neural network could individualise the contributions of distinct sections of the motor cortex. Even so, although the coarse data, the shallow neural network performs well above the chance level. Accuracies are between 67.5% and 92.5%, and precisions are between 64.3% and 97.2%. ICs other than those listed in Table 4 also contributed to the performance. Even so, the frugal shallow neural network maintains correct predictions well above the chance level.

The quality analysis of the network reveals that it has no intrinsic biases because when it is fed with random values, the predictions drop to the chance level. Another challenging aspect of the procedure is the differential processing of the train and test data. Testing the train dataset processed the same way as the test data set reveals that, although different, both processes are equivalent (cf. Table 2).

Further work should focus on three pathways: data reduction, feature extraction, and explaining hidden nodes. In fMRI acquisitions, one volume typically corresponds to a vector with around 60,000 elements, which is intractable for artificial neural networks, as the number of training epochs is several magnitudes below. Thus, data reduction imposes, and methods other than ICA may be explored to improve detail, which is needed for the sake of better explainable networks. The feature extraction here applied is comfortable for block designs, but it is not extensible to event-related. For such purpose, the particularities of the hemodynamic response may be modelled in order its capture is more reliable, signalling the process in the brain. Finally, the study may extend into the explanation to hidden nodes, revealing decision hubs.

**Acknowledgements.** This work was partially financially supported by Base Funding - UIDB/00027/2020 of the Artificial Intelligence and Computer Science Laboratory – LIACC - funded by national funds through the FCT/MCTES (PIDDAC).

## References

1. Samek, W., Müller, K.-R.: Towards explainable artificial intelligence. In: Samek, W., Montavon, G., Vedaldi, A., Hansen, L.K., Müller, K.-R. (eds.) *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence) LNCS. (LNAI)*, vol. 11700, pp. 5–22. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-28954-6\\_1](https://doi.org/10.1007/978-3-030-28954-6_1)
2. Tjoa, E., Guan, C.: A survey on explainable artificial intelligence (XAI): toward medical XAI. *IEEE Trans. Neural Netw. Learn. Syst.* **32**, 4793–4813 (2021). <https://doi.org/10.1109/TNNLS.2020.3027314>
3. Adadi, A., Berrada, M.: peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE Access* **6**, 52138–52160 (2018). <https://doi.org/10.1109/ACCESS.2018.2870052>
4. Van Essen, D.C., Smith, S.M., Barch, D.M., Behrens, T.E.J., Yacoub, E., Ugurbil, K.: The WU-minn human connectome project: an overview. *Neuroimage* **80**, 62–79 (2013). <https://doi.org/10.1016/j.neuroimage.2013.05.041>
5. Hanson, S.J., Matsuka, T., Haxby, J.V.: Combinatorial codes in ventral temporal lobe for object recognition: Haxby (2001) revisited: is there a “face” area? *Neuroimage* **23**, 156–166 (2004). <https://doi.org/10.1016/j.neuroimage.2004.05.020>
6. Misaki, M., Miyachi, S.: Application of artificial neural network to fMRI regression analysis. *Neuroimage* **29**, 396–408 (2006). <https://doi.org/10.1016/j.neuroimage.2005.08.002>
7. Santos, J.P., Moutinho, L.: Tackling the cognitive processes that underlie brands’ assessments using artificial neural networks and whole brain fMRI acquisitions. In: 2011 IEEE International Workshop on Pattern Recognition in NeuroImaging (PRNI), pp. 9–12. IEEE Computer Society, Seoul, Republic of Korea (2011)

8. Marques dos Santos, J.P., Moutinho, L., Castelo-Branco, M.: ‘Mind reading’: hitting cognition by using ANNs to analyze fMRI data in a paradigm exempted from motor responses. In: International Workshop on Artificial Neural Networks and Intelligent Information Processing (ANNIIP 2014), pp. 45–52. Scitepress (Scienceand Technology Publications, Lda.), Vienna, Austria (2014)
9. Beckmann, C.F., Smith, S.M.: Probabilistic independent component analysis for functional magnetic resonance imaging. *IEEE Trans. Med. Imaging* **23**, 137–152 (2004). <https://doi.org/10.1109/TMI.2003.822821>
10. Van Essen, D.C., Glasser, M.F.: The human connectome project: progress and prospects. *Cerebrum* 2016, cer-10–16 (2016)
11. Elam, J.S., et al.: The human connectome project: a retrospective. *Neuroimage* **244**, 118543 (2021). <https://doi.org/10.1016/j.neuroimage.2021.118543>
12. Buckner, R.L., Krienen, F.M., Castellanos, A., Diaz, J.C., Yeo, B.T.T.: The organization of the human cerebellum estimated by intrinsic functional connectivity. *J. Neurophysiol.* **106**, 2322–2345 (2011). <https://doi.org/10.1152/jn.00339.2011>
13. Yeo, B.T.T., et al.: The organization of the human cerebral cortex estimated by intrinsic functional connectivity. *J. Neurophysiol.* **106**, 1125–1165 (2011). <https://doi.org/10.1152/jn.00338.2011>
14. Jenkinson, M., Beckmann, C.F., Behrens, T.E.J., Woolrich, M.W., Smith, S.M.: FSL. *NeuroImage* **62**, 782–790 (2012). <https://doi.org/10.1016/j.neuroimage.2011.09.015>
15. Jenkinson, M., Smith, S.M.: A global optimisation method for robust affine registration of brain images. *Med. Image Anal.* **5**, 143–156 (2001). [https://doi.org/10.1016/S1361-8415\(01\)00036-6](https://doi.org/10.1016/S1361-8415(01)00036-6)
16. Jenkinson, M., Bannister, P.R., Brady, J.M., Smith, S.M.: Improved optimization for the robust and accurate linear registration and motion correction of brain images. *Neuroimage* **17**, 825–841 (2002). [https://doi.org/10.1016/S1053-8119\(02\)91132-8](https://doi.org/10.1016/S1053-8119(02)91132-8)
17. Smith, S.M.: Fast robust automated brain extraction. *Hum. Brain Mapp.* **17**, 143–155 (2002). <https://doi.org/10.1002/hbm.10062>
18. Woolrich, M.W., Ripley, B.D., Brady, J.M., Smith, S.M.: Temporal autocorrelation in univariate linear modeling of FMRI data. *Neuroimage* **14**, 1370–1386 (2001). <https://doi.org/10.1006/nimg.2001.0931>
19. Worsley, K.J.: Statistical analysis of activation images. In: Jezzard, P., Matthews, P.M., Smith, S.M. (eds.) *Functional MRI: An Introduction to Methods*, pp. 251–270. Oxford University Press, New York (2001)
20. Woolrich, M.W., Behrens, T.E.J., Beckmann, C.F., Jenkinson, M., Smith, S.M.: Multilevel linear modelling for FMRI group analysis using Bayesian inference. *Neuroimage* **21**, 1732–1747 (2004). <https://doi.org/10.1016/j.neuroimage.2003.12.023>
21. Beckmann, C.F., Jenkinson, M., Smith, S.M.: General multilevel linear modeling for group analysis in FMRI. *Neuroimage* **20**, 1052–1063 (2003). [https://doi.org/10.1016/S1053-8119\(03\)00435-X](https://doi.org/10.1016/S1053-8119(03)00435-X)
22. Minka, T.P.: Automatic choice of dimensionality for PCA. Technical Report 514. MIT Media Lab Vision and Modeling Group, MIT (2000)
23. Hyvärinen, A.: Fast and robust fixed-point algorithms for independent component analysis. *IEEE Trans. Neural Netw.* **10**, 626–634 (1999). <https://doi.org/10.1109/72.761722>
24. Buckner, R.L.: Event-related fMRI and the hemodynamic response. *Hum. Brain Mapp.* **6**, 373–377 (1998). [https://doi.org/10.1002/\(SICI\)1097-0193\(1998\)6:5/6%3c373::AID-HBM8%3e3.0.CO;2-P](https://doi.org/10.1002/(SICI)1097-0193(1998)6:5/6%3c373::AID-HBM8%3e3.0.CO;2-P)

25. Limas, M.C., Meré, J.B.O., Marcos, A.G., Ascacibar, F.J.M.d.P., Espinoza, A.V.P., Elías, F.A.: AMORE: A more flexible neural network package. León (2010)
26. R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna (2010)
27. Penfield, W., Boldrey, E.: Somatic motor and sensory representation in the cerebral cortex of man as studied by electrical stimulation. *Brain* **60**, 389–443 (1937). <https://doi.org/10.1093/brain/60.4.389>



# A Deep Learning-Based Method for Uncovering GPCR Ligand-Induced Conformational States Using Interpretability Techniques

Mario A. Gutiérrez-Mondragón<sup>(✉)</sup>, Caroline König, and Alfredo Vellido

Computer Science Department, and Intelligent Data Science and Artificial  
Intelligence (IDEAI-UPC) Research Center, Universitat Politècnica de Catalunya,  
Barcelona, Spain

{mario.alberto.gutierrez, caroline.leonore.konig}@upc.edu,  
avellido@cs.upc.edu

**Abstract.** There is increasing interest in the development of tools for investigating the protein *ligand* space. Understanding the underlying mechanisms of G protein-coupled receptors (GPCR) in the *ligand-binding* process is of particular interest due to their role in pharmacoproteomics. In this work, we propose the study of GPCR *ligand-induced* conformational variations from Molecular Dynamics (MD) simulations using Deep Learning (DL)-based methods. We devise and train a Convolutional Neural Network (CNN) for classifying the states for both ligand-free structure and the bound of agonists in the  $\beta 2$ -adrenergic receptor. We also study the transformation of MD data into an interaction network matrix to further improve and facilitate the analyses without significant loss of information. Our method introduces a framework for the study of the effect of *ligand-receptor* binding activity that includes a novel analysis based on interpretability algorithms, allowing the quantification of the contribution of individual residues to structural re-arrangements.

**Keywords:** Proteomics · GPCRs · Molecular dynamics · Residue interaction networks · Deep learning · Convolutional networks · Interpretability · Layer wise relevance

## 1 Introduction

Machine learning (ML)-based models are suitable tools for the extraction of knowledge from the data stemming from the study of biological processes [37] and also for handling, processing, and analyzing the massive amount of data often generated by different biological sources [21]. The current study deals with the analysis of protein MD simulations. Although X-ray crystallography research has boosted the study of proteins, it provides very limited information on the

---

This research is partially funded by research grant PID2019-104551RB-I00.

© Springer Nature Switzerland AG 2022

I. Rojas et al. (Eds.): IWBBIO 2022, LNBI 13347, pp. 275–287, 2022.

[https://doi.org/10.1007/978-3-031-07802-6\\_23](https://doi.org/10.1007/978-3-031-07802-6_23)

dynamic nature of their structures. In this scenario, MD data can help to incorporate the missing information regarding the dynamics of receptors.

The target of this study are G protein-coupled receptors (GPCRs) [36], primary receptors in cell membranes for signal transduction, as they can respond to plenty of signaling molecules (*ligands*) [31]. This confers them relevance as druggable targets for treating diverse diseases (Alzheimer's, cancer, and pulmonary illnesses [11,25], for instance). Broadly, the study of structural and physico-chemical properties of proteins is crucial in the drug development process [26]. Nevertheless, the endless number of atomic rearrangements that a protein can present constrains the comprehension of its function [14,19].

In this context, the large-scale analysis of protein processes through MD simulations becomes crucial to elucidate their functional properties, including the *protein-ligand* interactions and the identification of *druggable* binding pockets [4,23]. The massive generation of MD information has turned the investigation of the dynamic nature of the receptors into a Big Data problem [40], a challenging area of research. In this realm, academy and industry have already made important inroads in molecular biology problems exploiting the strengths of ML-based algorithms [27].

We focus our effort here on the development of a DL-based approach to the investigation of receptor conformations related to its function. As part of this approach, the identification of the *motifs* (residues or subsets of residues) of the protein that undergo conformational state changes is a central goal. To achieve it, we propose a CNN model to classify agonists-specific functional responses in the GPCR- $\beta_2$ -adrenergic ( $\beta_2AR$ ) receptor from MD simulations. The proposed method also involves an MD data transformation into a representation that might further improve and facilitate the analyses. For this purpose, we use residue interaction networks (RIN), an intuitive representation of the complexity of MD trajectories. Besides, we provide a way to measure the trustworthiness of the results, relying on an interpretability analysis of the model predictions. Importantly, the method allows us to reveal the *learned* molecular properties by identifying the residues that induce *ligand-dependent* conformations. From the obtained results, we expect that this method can be applied successfully to related problems.

## 2 Related Work

The exploration of the *state-of-the-art* shows that ML-based methods have boosted the drug discovery process [3,6,9,20,37]. In this work, we investigate the use of DL-based models and, particularly, CNNs. These models have been applied mostly to image analysis, including medical image, tackling problems such as cancer detection [24], or neuro-degenerative analysis [39], to name just a couple. DL methods have also been used in recent years in proteomics-related problems [28]. For instance, the study of ligand-protein interactions is discussed in [22], and [34], to name a few. Specifically, CNNs have been used of late in different applications related to the ligand influence in the protein structure. Some

examples include [8, 15–17], and [12], whose proposals stands on CNN architectures for *protein-ligand* binding prediction.

Although the results are generally encouraging, the interpretability and explainability of the predictions have been poorly investigated. The classical metrics to evaluate DL-based models are insufficient to disclose the mechanisms that induce the making-decision process for classifying a specific *ligand-dependent* protein response, as stated in [29, 30], and [38].

### 3 Materials

The data under study include the molecular structures for GPCRs simulated on the Google Exacycle platform [18]. This dataset consists of MD simulations of both inactive (PDB 2RH1) and active (PDB 3P0G) states of the  $\beta_2$ -adrenergic ( $\beta_2AR$ ) receptor and the assessment of the bounding of the inverse agonist carazolol and the full agonist BI-167107.

Broadly, the raw data is generated by the simulation of two milliseconds of the dynamics of the receptor. Each simulation state (free and induced-ligand) includes 10,000 trajectories generated with *Gromacs* molecular dynamics package [13], along with the structure file that summarizes the protein information, i.e., the sequence and the list of the atoms and their coordinates.

For our case study, we have randomly chosen 500 trajectories of the MD simulations related to the inactive (2RH1) state of the receptor. Details about the three dimensional structure of the inactive state were provided by crystallography [7]. While the study reported in [18] used several derived metrics from the receptor structure to identify distinct activation pathways related to the ligand-free structure and to the agonist binding, our experiments aim to identify relevant interactions between residues for each type of simulation. For this reason the present study analyzes the MD simulation data at the level of residues. However, to accomplish this goal, we first transform the data into a readable form that fits into the CNN model, as described in the following subsection.

#### 3.1 Protein as Residue Interaction Network Representation

To ease the learning process of conformational variations, each simulation frame from the trajectories is transformed into a RIN. This representation of the protein has been shown to facilitate the study of the structure and function of proteins. Besides, some studies suggest that this compact expression of the receptor has the advantage of capturing important elements of the global structural properties, function, folding, and stability of proteins [1, 5, 10, 35].

Furthermore, a two-dimensional representation simplifies the problem, making it more suitable for the CNN model. It also allows us to increase the training data, as several samples of the same state can be generated to analyze different conformational states. Commonly, in DL models, generalization capability improves with an increased and diverse training set.

The idea is simple: to represent the three-dimensional protein structure in a two-dimensional space where the atomic coordinates of residues represent network nodes, and the interaction strength between them, the edges in a graph network. The pairwise interaction strength of residues is evaluated using the *Protein Structure Network* (PSN) module from *Wordom* software, suitable for the analysis of MD trajectories, [33]. Roughly speaking, this module evaluates the interaction strength  $I_{ij}$  as a percentage of the interaction of distinct pairs of nodes within a given distance cutoff.

$$I_{ij} = \frac{n_{ij}}{\sqrt{N_i N_j}} 100, \quad (1)$$

where  $n_{ij}$  is the number of atoms pairs of a side-chain given within a determined cutoff distance.  $N_i$  and  $N_j$  are residue normalization factors taken from the work by Kannan and Vishveshwara, where more details on the procedure can be consulted, [5].

In general, the *Wordom* implementation of PSN analysis allows us to modify several cutoffs, perform residue selection, and achieve different network representations by probing a range of minimal interaction cutoff, etc. The analysis requires the structure file as a reference and a trajectory file to provide the coordinate sets. The module writes an output file that contains the residue-residue interaction strengths of all trajectory frames. In our case study, we pre-processed this output file to generate square and sparse matrices that were in turn fed into the CNN model.

### 3.2 Data Pre-processing

The working proposal suggests the transformation of the output data given by the PSN module in square and sparse matrix form. In our experiments, it takes 500 trajectories from the 2RH1 state of the protein and considers the analysis of the structure for both free-of-ligand and induced by inverse and full agonists.

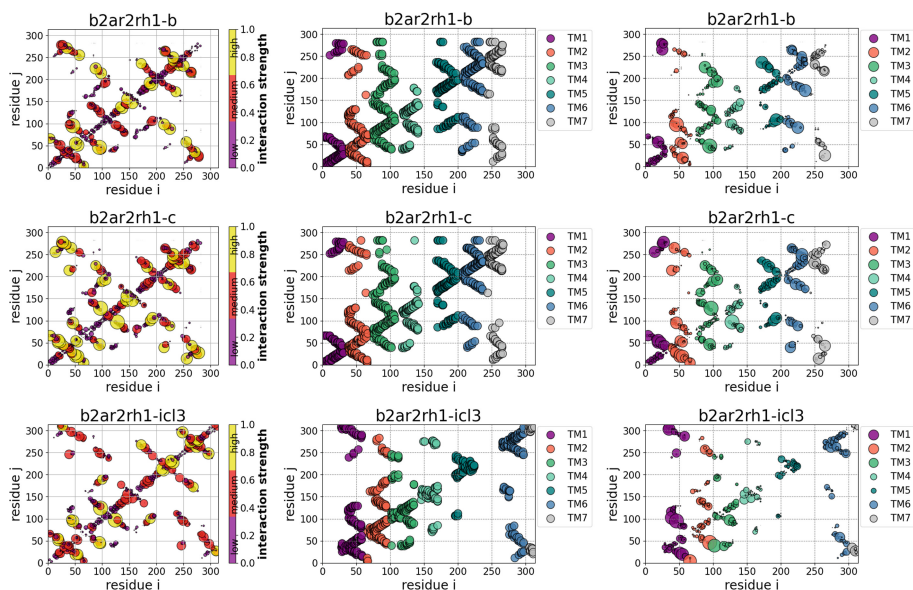
Therefore, to express each frame in the trajectories as an  $N \times N$  array of pairwise interactions, we consider the residue number as the  $XY$  position coordinate and assign the corresponding interaction strength value. Besides, the data is normalized and the resulting matrices are of  $314 \times 314$  dimension. The distribution of the data is shown in Table 1.

**Table 1.** Data distribution.

Class	Description	# trajectories	# frames
2RH1-b	$\beta_2AR$ full agonist	500	13,482
2RH1-c	$\beta_2AR$ inverse agonist	500	12,852
2RH1-icl3	$\beta_2AR$ apo (no ligand)	500	12,765
Total:		1500	39,099



To visualize the sparse matrices, the non-zero values are extracted to draw a scatter plot, color-coded according to the interaction strength. A representative sample for each class is shown in Fig. 1. Furthermore, we show the receptor transmembrane regions in the second column of the figure, according to the list of residues in [32]. For this case, each color represents a region, but we also present a visualization that merges the transmembrane regions and the interaction strength in the third column of the figure.



**Fig. 1.** Each row shows three different illustrations for expressing the RIN of each structure class. All matrices are color-coded and the axes express pairs of interacting residues ( $i, j$ ). First column: interaction strength expressed through color-coding and by the size of the marker; second column: color-coded distinct transmembrane (TM) regions; third column: interaction strength of each transmembrane region using the marker size. (Color figure online)

This interaction network representation incorporates translational and rotational invariance, which is suitable for improving the generalization of most ML models, including CNNs. Worth mentioning that the data under analysis includes plenty of matrices with small variations which make the use of generalization techniques like data augmentation unproductive.

## 4 Experimental Setup

In this study, we have first split the dataset into training and test samples. The distribution of the data is shown in Table 2. For the training and validation of our data, stratified *k-fold cross-validation* was used, with  $k = 5$ .

**Table 2.** Data split distribution.

Class	Description	# training samples	# test samples
2RH1-b	$\beta_2AR$ full agonist	8,089	5,393
2RH1-c	$\beta_2AR$ inverse agonist	7,711	5,141
2RH1-icl3	$\beta_2AR$ apo (no ligand)	7,659	5,106
Total:		23,459	15,640

CNNs are DL models designed mostly for image analysis tasks and are composed of a sequence of stacked layers that can *learn* complex representations through simple, nonlinear modules. Their main building blocks are the convolution, pooling, and fully connected layers. In classification problems, the convolution module aims to identify relevant features in the form of feature maps representing abstractions of shape, patterns, or colors. Commonly, each convolution module includes an activation function to add non-linearity to the model. Following the feature maps, a dimensionality reduction layer is set. In most cases, it uses *max* or *average pooling* layers. Finally, a fully connected layer performs the final classification over the extracted features. In contrast to shallow artificial neural networks, the learned patterns are translation invariant and have a degree of rotational invariance.

In the context of our analysis, learning spatial information is relevant, i.e., our model should detect the position of the molecular conformations. Moreover, a subsequent interpretability study of the results should allow the identification of those residues (amino acids of the protein) that induce such structural rearrangements. The best practice for defining the CNN architecture is to start from a shallow one and gradually increase its size until under-fitting vanishes. However, our early experiments showed that a larger number of filters and the increase in kernel size provided no improvement. A CNN with only two convolutional layers, *relu* activation function, and *max pooling* yielded the best results. Besides, we have established two fully connected layers to get the probabilities over the feature maps for each class. The proposed architecture for addressing the classification problem is summarized in Table 3.

**Table 3.** CNN architecture proposed.

Layer (type)	Output shape	# Param
Conv2d-1	[-1, 32, 310, 310]	832
ReLu-2	[-1, 32, 310, 310]	0
MaxPool2d-3	[-1, 32, 155, 155]	0
Conv2d-4	[-1, 32, 151, 151]	25,632
ReLu-5	[-1, 32, 151, 151]	0
MaxPool2d-6	[-1, 32, 75, 75]	0
Flatten-7	[-1, 180000]	0
Linear-8	[-1, 32]	5,760,032
ReLU-9	[-1, 32]	0
Dropout-10	[-1, 32]	0
Linear-11	[-1, 3]	99
Total params: 5,786,595		
Trainable params: 5,786,595		
Non-trainable params: 0		

The selection of this small architecture was empirical, and so was the choice of hyper-parameters. Thus, in the context of convolution layers, we set  $5 \times 5$  window size, no padding and stride value to 2. Besides, we use drop-out as a regularization method to 0.5 and we trained this architecture using *cross-entropy* as a loss function and adaptive moment estimation (ADAM) optimizer with a learning rate value to  $1 \times 10^{-4}$ . The algorithms and computations were developed using *Python* (version 3.9.7) and, *Pytorch* machine learning framework (version 1.10.1).

## 5 Results and Discussion

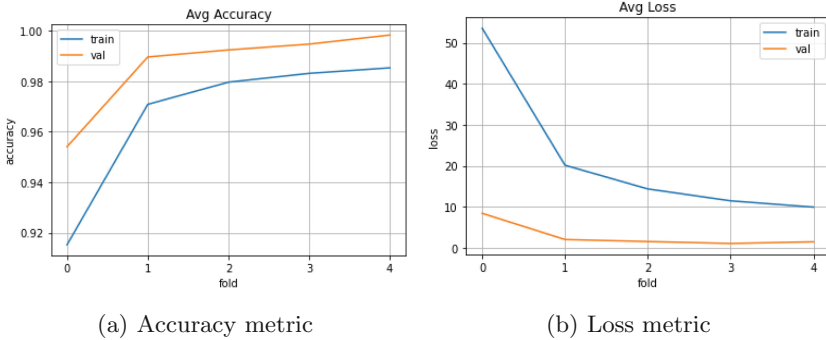
Table 4 summarizes the results of the CNN training process. The average accuracy for both training and validation sets is in the 90–100% interval and often near 99%. Furthermore, there is no evident gap between the reported training and validation accuracies in each fold, i.e., there is no obvious presence of overfitting in our training procedure as demonstrated in the learning curves reported in Fig. 2.

Importantly, in terms of assessing the generalization capabilities of the learned model, and as previously stated, we made predictions on unseen (test) data. In our experiments, the test accuracy was 99.76%.

Despite the experimental evidence about the quality of our model, there is a remaining step to perform. We should make sure that the model predictions are made considering meaningful patterns (residues in our particular case) for the classification. Therefore, the interpretation of the predictions by a sensitivity analysis is a crucial aspect of our evaluation process to succeed in our aims.

**Table 4.** The first column represents the trained model. The following two columns shows the average score for the loss function in the training and validation sets. The computed average accuracy value on training and validation sets is reported in the last two columns.

# fold	AVG Tr Loss	AVG Val Loss	AVG Tr Accuracy	AVG Val Accuracy
1	53.5658	8.3702	0.9152	0.9541
2	20.1192	1.9798	0.9709	0.9897
3	14.3261	1.4840	0.9797	0.9925
4	11.4291	1.0138	0.9832	0.9948
5	9.8927	1.4139	0.9854	0.9984
Fold AVG:	<b>21.8665</b>	<b>2.8523</b>	<b>0.9668</b>	<b>0.9859</b>

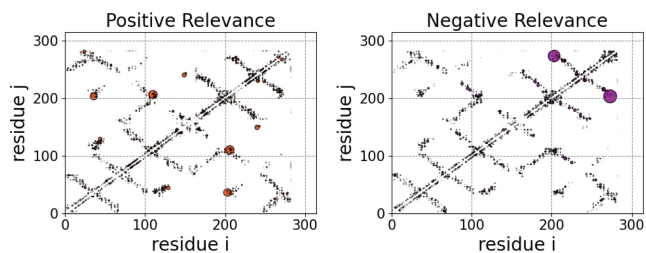


**Fig. 2.** Average accuracy/loss curves across the five *cross-validation* folds.

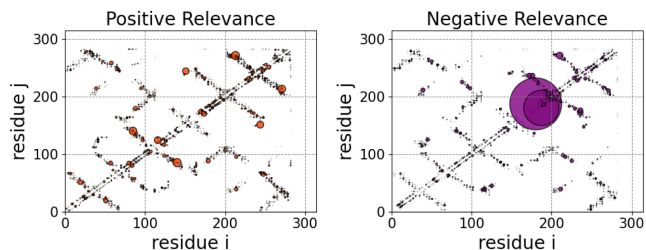
## 5.1 Interpretability of the Results

It is well known that DL methods are most successful in many scientific and industrial domains. Nevertheless, paradoxically, their complexity could also be their major constraint, because the decisions making process is notoriously lacking transparency. Trusting the results of a black-box model is not ideal (or even unacceptable in some domains), no matter the reported by the evaluations metrics (e.g., accuracy and loss). Both the model decision and its interpretation are crucial for risk assessment in many domains. Recent research has focused on the development of methods to provide these models with some level of transparency.

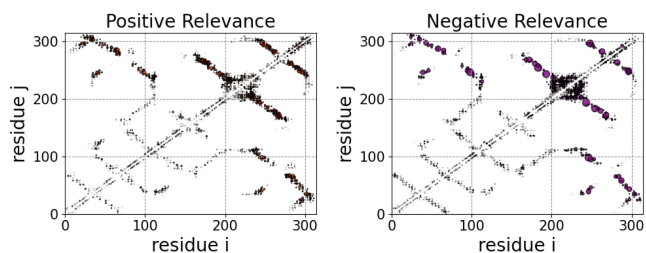
Layer-Wise Relevance Propagation (LRP) is one of the most prominent techniques to provide intuitive human-readable explanations of the model predictions, [2]. This method produces a *heatmap* of the input space, highlighting features that are relevant for the output. The core idea behind the algorithm is the identification of the contribution of the neurons by back-propagating the prediction layer by layer until reaching the input. The magnitude of the contribution is called relevance, and it is redistributed equally in each layer. The neurons that contribute the most are assigned higher relevance.



(a) Average Relevance for residues when binding the full agonist BI-167107.



(b) Average Relevance for residues when binding the inverse agonist carazolol.



(c) Average Relevance for residues for free-ligand structure.

**Fig. 3.** Contribution of pairs residues for the structure conformation of each class. Their relevance is color-coded (red and blue for positive and negative influence, respectively), and their magnitude is represented by the size of the marker. (Color figure online)

The proposed interpretability study for our experiments computes the map of relevance for each instance of a class in the test set by summing the contribution of pairs of residues and reporting the average relevance. Therefore, we generate relevance maps to identify residues that influence positively and negatively structural conformations when the receptor is either *ligand-free* or *agonists-induced*. The results are shown in Fig. 3 where it is noticeable that the trained model has succeeded in identifying relevant motives in the classification of protein structural re-arrangements.

Furthermore, from the relevance maps, we have computed the influence of the transmembrane regions on the conformation of the structure when binds

by *ligands* and when is *ligand-free*. To accomplish this task, our strategy is to identify the set of residues that positively influence a transmembrane and compute the average relevance for each region. The results are shown in Table 5.

**Table 5.** First column: TM region; second column: number of pairs of residues interacting (RI); third column: average relevance of the transmembrane.

(a) Reported average influence to structural rearrangements when the receptor binds the full agonist BI-167107.

region	# RI	AVG R
TM1	178	0.002775
TM2	219	0.003977
TM3	236	0.003365
TM4	126	0.005040
TM5	211	0.004188
<b>TM6</b>	<b>269</b>	<b>0.005906</b>
TM7	186	0.003853

(b) Reported average influence to structural rearrangements when the inverse agonist carazolol is present.

region	# RI	AVG R
TM1	183	0.006238
TM2	239	0.007591
TM3	260	0.013146
<b>TM4</b>	<b>116</b>	<b>0.018101</b>
TM5	219	0.011499
TM6	291	0.011066
TM7	213	0.012098

(c) Reported average influence to structural rearrangements when the receptor is ligand-free.

region	# RI	AVG R
TM1	214	0.005606
TM2	179	0.002913
TM3	251	0.002294
TM4	164	0.002551
TM5	445	0.003197
<b>TM6</b>	<b>257</b>	<b>0.012646</b>
TM7	66	0.002494

Overall, we are providing a tool that gives insights into the residue's influence when the receptor binds different agonists. Furthermore, this analysis enables recognizing conformational variations that can be subtle but relevant for the functional properties of the protein. The proposed method is thus important for the analysis of MD simulations in terms of identifying and distinguishing molecular conformations induced by the *ligand-binding* process.

## 6 Conclusions

The opportunities provided by the analysis of MD simulations through ML-based methods in disciplines such as molecular biology and biochemistry have provided relevant insights in understanding the function, dynamics, and molecular processes of protein structures. In this realm, our study contributes an exploration of receptor conformational activity, using a GPCR subtype as an example.

We have proposed a methodology to train a CNN-based model from MD trajectories to analyze the activation of the  $\beta_2$ -adrenergic ( $\beta_2AR$ ) receptor when bound to the inverse agonist carazolol and the full agonist BI-167107. This methodology includes the transformation of MD data into a more suitable format for analyzing and computationally using interaction networks. The results of our experiments show that this transformation provides a simplified version of the structure (from the atomic to the residue level) that still preserves the most relevant features for investigating the ligand space.

In general terms, the results provide evidence that our CNN model can recognize the relationship of the ligand-dependent conformational details to generate knowledge that could be useful to elucidate the dynamic processes of the receptor. One of the most relevant outcomes is the identification of *motifs* (groups of residues) of the receptor through an interpretability study. Consequently, we are providing a framework for understanding and assessing the mechanisms that underlie conformational rearrangements involved in the functional states of the protein. It is worth stressing that few works currently address the application of interpretability techniques in MD problems. Our approach provides an advantage by improving the trustworthiness of the CNN model and a method to further assess its predictions in this domain.

## References

1. Amitai, G., et al.: Network analysis of protein structures identifies functional residues. *J. Mol. Biol.* **344**(4), 1135–1146 (2004)
2. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE* **10**(7), e0130140 (2015)
3. Bajorath, J., et al.: Artificial intelligence in drug discovery: Into the great wide open. *J. Med. Chem.* **63**(16), 8651–8652 (2020)
4. Bera, I., Payghan, P.V.: Use of molecular dynamics simulations in structure-based drug discovery. *Curr. Pharm. Des.* **25**(31), 3339–3349 (2019)
5. Brinda, K., Vishveshwara, S.: A network representation of protein structures: implications for protein stability. *Biophys. J.* **89**(6), 4159–4170 (2005)
6. Chan, H.S., Shan, H., Dahoun, T., Vogel, H., Yuan, S.: Advancing drug discovery via artificial intelligence. *Trends Pharmacol. Sci.* **40**(8), 592–604 (2019)
7. Cherezov, V., et al.: High-resolution crystal structure of an engineered human  $\beta_2$ -adrenergic G protein-coupled receptor. *Science* **318**(5854), 1258–1265 (2007)
8. Cui, Y., Dong, Q., Hong, D., Wang, X.: Predicting protein-ligand binding residues with deep convolutional neural networks. *BMC Bioinform.* **20**(1), 1–12 (2019)

9. Fleming, N.: How artificial intelligence is changing drug discovery. *Nature* **557**(7706), S55–S55 (2018)
10. Greene, L.H.: Protein structure networks. *Brief. Funct. Genomics* **11**(6), 469–478 (2012)
11. Gutierrez, A.N., McDonald, P.H.: GPCRs: emerging anti-cancer drug targets. *Cell. Signal.* **41**, 65–74 (2018)
12. Hassan-Harriou, H., Zhang, C., Lemmin, T.: RosENet: improving binding affinity prediction by leveraging molecular mechanics energies with an ensemble of 3D convolutional neural networks. *J. Chem. Inf. Model.* **60**(6), 2791–2802 (2020)
13. Hess, B., Kutzner, C., Van Der Spoel, D., Lindahl, E.: GROMACS 4: algorithms for highly efficient, load-balanced, and scalable molecular simulation. *J. Chem. Theory Comput.* **4**(3), 435–447 (2008)
14. Hollingsworth, S.A., Dror, R.O.: Molecular dynamics simulation for all. *Neuron* **99**(6), 1129–1143 (2018)
15. Hu, S., Zhang, C., Chen, P., Gu, P., Zhang, J., Wang, B.: Predicting drug-target interactions from drug structure and protein sequence using novel convolutional neural networks. *BMC Bioinform.* **20**(25), 1–12 (2019)
16. Jiang, H., et al.: Guiding conventional protein-ligand docking software with convolutional neural networks. *J. Chem. Inf. Model.* **60**(10), 4594–4602 (2020)
17. Jiménez, J., Doerr, S., Martínez-Rosell, G., Rose, A.S., De Fabritiis, G.: DeepSite: protein-binding site predictor using 3d-convolutional neural networks. *Bioinformatics* **33**(19), 3036–3042 (2017)
18. Kohlhoff, K.J., et al.: Cloud-based simulations on google exacycle reveal ligand modulation of GPCR activation pathways. *Nat. Chem.* **6**(1), 15–21 (2014)
19. Latorraca, N.R., Venkatakrishnan, A., Dror, R.O.: GPCR dynamics: structures in motion. *Chem. Rev.* **117**(1), 139–155 (2017)
20. Lavecchia, A.: Deep learning in drug discovery: opportunities, challenges and future prospects. *Drug Discov. Today* **24**(10), 2017–2032 (2019)
21. Libbrecht, M.W., Noble, W.S.: Machine learning applications in genetics and genomics. *Nat. Rev. Genet.* **16**(6), 321–332 (2015)
22. Lim, J., Ryu, S., Park, K., Choe, Y.J., Ham, J., Kim, W.Y.: Predicting drug-target interaction using 3D structure-embedded graph representations from graph neural networks. *arXiv preprint [arXiv:1904.08144](https://arxiv.org/abs/1904.08144)* (2019)
23. Liu, X., Shi, D., Zhou, S., Liu, H., Liu, H., Yao, X.: Molecular dynamics simulations and novel drug discovery. *Expert Opin. Drug Discov.* **13**(1), 23–37 (2018)
24. Lundervold, A.S., Lundervold, A.: An overview of deep learning in medical imaging focusing on MRI. *Z. Med. Phys.* **29**(2), 102–127 (2019)
25. Lundstrom, K.: An overview on GPCRs and drug discovery: structure-based drug design and structural biology on GPCRs. *G Protein-Coupled Receptors Drug Discov.* **552**, 51–66 (2009)
26. Maurice, P., Guillaume, J.L., Benleulmi-Chaachoua, A., Daulat, A.M., Kamal, M., Jockers, R.: GPCR-interacting proteins, major players of GPCR function. *Adv. Pharmacol.* **62**, 349–380 (2011)
27. Meyer, J.G.: Deep learning neural network tools for proteomics. *Cell Rep. Methods* **1**(2), 100003 (2021)
28. Paliwal, K., Lyons, J., Heffernan, R.: A short review of deep learning neural networks in protein structure prediction problems. *Adv. Tech. Biol. Med.* **3**(3), 1–2 (2015)
29. Plante, A., Shore, D.M., Morra, G., Khelashvili, G., Weinstein, H.: A machine learning approach for the discovery of ligand-specific functional mechanisms of GPCRs. *Molecules* **24**(11), 2097 (2019)



30. Preuer, K., Klambauer, G., Rippmann, F., Hochreiter, S., Unterthiner, T.: Interpretable deep learning in drug discovery. In: Samek, W., Montavon, G., Vedaldi, A., Hansen, L.K., Müller, K.-R. (eds.) *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. LNCS (LNAI), vol. 11700, pp. 331–345. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-28954-6\\_18](https://doi.org/10.1007/978-3-030-28954-6_18)
31. Rosenbaum, D.M., Rasmussen, S.G., Kobilka, B.K.: The structure and function of G-protein-coupled receptors. *Nature* **459**(7245), 356–363 (2009)
32. Rosenbaum, D.M., et al.: Structure and function of an irreversible agonist- $\beta$ 2 adrenoceptor complex. *Nature* **469**(7329), 236–240 (2011)
33. Seeber, M., Cecchini, M., Rao, F., Settanni, G., Cafilisch, A.: Wordom: a program for efficient analysis of molecular dynamics simulations. *Bioinformatics* **23**(19), 2625–2627 (2007)
34. Shen, C., Ding, J., Wang, Z., Cao, D., Ding, X., Hou, T.: From machine learning to deep learning: advances in scoring functions for protein-ligand docking. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **10**(1), e1429 (2020)
35. Snider, J., Kotlyar, M., Saraon, P., Yao, Z., Jurisica, I., Stajlgjar, I.: Fundamentals of protein interaction network mapping. *Mol. Syst. Biol.* **11**(12), 848 (2015)
36. Torrens-Fontanals, M., Stepniewski, T.M., Aranda-García, D., Morales-Pastor, A., Medel-Lacruz, B., Selent, J.: How do molecular dynamics data complement static structural data of GPCRs. *Int. J. Mol. Sci.* **21**(16), 5933 (2020)
37. Vamathevan, J., et al.: Applications of machine learning in drug discovery and development. *Nat. Rev. Drug Discov.* **18**(6), 463–477 (2019)
38. Vellido, A.: The importance of interpretability and visualization in machine learning for applications in medicine and health care. *Neural Comput. Appl.* **32**(24), 18069–18083 (2019). <https://doi.org/10.1007/s00521-019-04051-w>
39. Vieira, S., Pinaya, W.H., Mechelli, A.: Using deep learning to investigate the neuroimaging correlates of psychiatric and neurological disorders: methods and applications. *Neurosci. Biobehav. Rev.* **74**, 58–75 (2017)
40. Zhu, H.: Big data and artificial intelligence modeling for drug discovery. *Annu. Rev. Pharmacol. Toxicol.* **60**, 573–589 (2020)



# Data Transformation for Clustering Utilization for Feature Detection in Mass Spectrometry

Vojtech Barton<sup>1,2</sup>  and Helena Skutkova<sup>1</sup> 

<sup>1</sup> Department of Biomedical Engineering, Faculty of Electrical Engineering,  
Brno University of Technology, Brno 61600, Czech Republic  
[barton@vutbr.cz](mailto:barton@vutbr.cz)

<sup>2</sup> RECETOX, Faculty of Science, Masaryk University,  
Kotlarska 2, Brno, Czech Republic

**Abstract.** Feature detection and peak detection are one of the first steps of mass spectrometry data processing. This data comes in large volumes; thus, the processing needs to be optimized, not overloaded. State-of-the-art clustering algorithms can not perform feature detection for several reasons. First issue is the volume of the data, second is the disparity of the sampling frequency in the  $MZ$  and  $RT$  axis. Here we show the data transformation to utilize the clustering algorithms without the need to redefine its kernel. Data are first pre-clustered to obtain regions that can be processed independently. Then we transform the data so that the numerical differences between consecutive points should be the same in both space axes. We applied a set of clustering algorithms for each region to find the features, and we compared the result with the Gridmass peak detector. These findings may facilitate better utilization of the 2D clustering method as feature detectors for mass spectra.

**Keywords:** Mass spectrometry · Clustering · Feature identification

## 1 Introduction

Gas chromatography-Mass Spectrometry (GC-MS) is a widely used analytical method for volatile compounds in complex mixtures. Technologies in GC-MS experiments yield a large volume of data, and correct handling of the data processing is a crucial step in processing these experiments. There is a lot of processing steps to be careful of. It greatly impacts the extent and quality to which the ions can be identified a quantized [9].

After the data acquisition, we obtained a large set of consecutively measured spectra. Three values characterize each point in spectra.  $RT$  is the measure of retention time. It is the time section for the analyst to be evaluated from the chromatographic column.  $MZ$  is the measured ratio of the mass-to-charge of the ion. The last descriptive feature of each point is the intensity of the measured

signal. Each data point in space is defined by the intensity function  $\psi(RT, MZ)$  [2, 6].

One of the first processing steps is to identify features in the data. By features, we identify the peaks and their properties. The usual method includes the centroiding step and several signal processing methods. This approach deals with data reduction so the large volume of the data can be effectively processed. Some methods, like *GridMass* [12], can work in 2D directly identifying the local maxima in the planar data.

In our approach, we will be showing the utilization of the clustering algorithms to identify the features, peaks, and their properties by segmenting the data plane. Several issues need to be solved. First, there is a need to deal with the volume of the data. The data needs to be divided into independent batches to be easily processed without the robust computing demands. Another issue is with a different dimension of RT and MZ. Each dimension has a different size, point density, and sampling frequency. There is a need for space transformation to utilize the clustering or define the distance in each dimension separately [4, 7].

We introduce the effective method for region segmentation, providing regions that can be processed independently of the others without splitting the features between the regions. To utilize the clustering, we propose data space transformation to be able to define 1D distance measure applicable in both dimensions. Last, we provide a proof-of-concept that clustering algorithms can identify the features in the data, and we test the method on a real dataset of the mass spectrometry experiment.

## 2 Materials and Methods

### 2.1 Dataset

To test the proposed method, we use an in-house generated dataset of a mixture of flame retardants at known proportions. The concentrations of the compounds were predefined at 1000 ng/ml, 500 ng/ml and 100 ng/ml. This dataset was obtained during the INTERFLAB experiment [8]. The experiment was measured by Thermofisher Scientific Q Exactive GC Orbitrap GC-MS machine. The measurement was performed in profile mode.

The composition of the dataset is shown in Table 1. For each of the compounds, the expected values of MZ and RT of their fragments were defined. This values was then projected to the raw data and manually corrected to the exact values of the local maxima in the corresponding regions. We then obtained a dataset containing 1057 known and annotated peaks. This peaks was set as ground truth to evaluate the performance of the method.

**Table 1.** List of compounds of the test mixture.

Compound name	Expected RT	Corrected RT
2IPDPDP	640.80	635.364
4IPDPDP	688.80	682.372
a-DP	945.00	939.089
Acenaphthylene d8	98.40	98.662
aTBCO	478.20	472.720
aTBECH	419.40	413.740
ATE	293.40	290.486
B4IPPPP	774.00	768.052
BATE	443.40	438.707
BDE 100 (2,2',4,4',6-Pentabromodiphenyl ether)	712.20	711.895
BDE 128 (2,2',3,3',4,4'-Hexabromodiphenyl ether)	856.80	861.607
BDE 153 (2,2',4,4',5,5'-Hexabromodiphenyl ether)	792.60	792.692
BDE 154 (2,2',4,4',5,6'-Hexabromodiphenyl ether)	763.80	763.544
BDE 183 (2,2',3,4,4',5',6-Heptabromodiphenyl ether)	861.60	861.337
BDE 28 (2,4,4'-Tribromodiphenyl ether)	528.60	528.965
BDE 47 (2,2',4,4'-Tetrabromodiphenyl ether)	637.80	638.250
BDE 71 (2,3',4',6-Tetrabromodiphenyl ether)	623.40	623.303
BDE 85 (2,2',3,4,4'-Pentabromodiphenyl ether)	745.80	745.393
BDE 99 (2,2',4,4',5-Pentabromodiphenyl ether)	690.00	689.565
BEHTBP	921.60	916.392
Benzoapyrene-d12	721.20	720.945
bTBCO	455.40	450.016
bTBECH	424.80	419.660
BTBPE	886.80	881.649
DBDPE	1246.80	1228.891
DPTE	592.80	588.993
EHDP	606.00	601.436
EHTBB	724.80	719.059
HBBZ	597.00	591.149

*(continued)*

**Table 1.** (*continued*)

Compound name	Expected RT	Corrected RT
HBCD	811.80	804.520
HCDBCO	721.20	715.044
OBIND	1099.80	1089.182
p-Terphenyl-d14	501.60	501.571
PBBA	683.40	678.061
PBBZ	450.00	444.897
PBEB	546.60	540.857
PBT	524.40	519.167
PCB121	451.80	452.172
pTBX	439.20	433.899
s-DP	925.20	918.766
T21PPP	720.00	713.967
T23BPIC	910.20	904.338
T35DMPP	759.60	753.379
TBBPA	799.80	792.692
TBBPA	799.20	792.692
TBCT	473.40	468.449
TBEP	610.80	604.131
TBEP	609.60	604.131
TBP	238.80	234.783
TCEP	244.20	244.992
T CPP	249.60	244.992
TDBPP	806.40	799.967
TDCPP	566.40	561.014
TDCPP	567.00	561.014
TEHP	567.00	561.014
TMTP	684.60	679.139
TOTP	657.60	651.241
TPP	592.20	587.190
TPTP	714.60	709.200

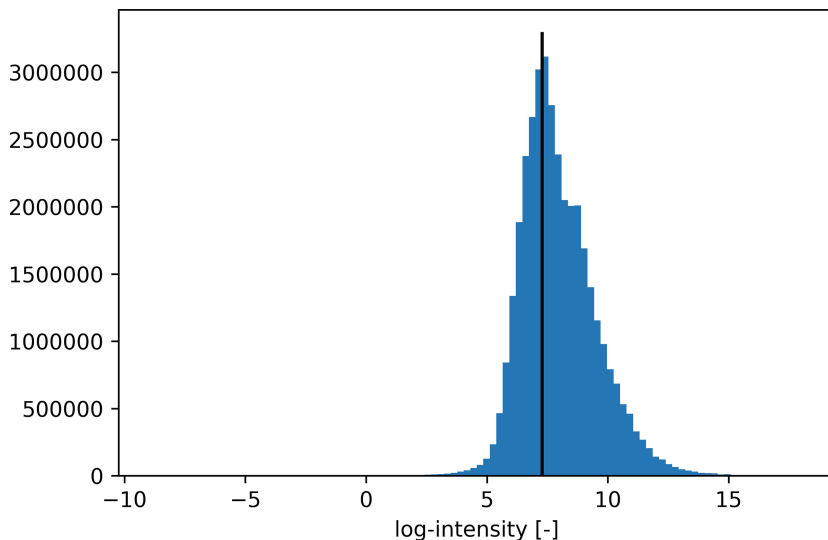
## 2.2 Space Division

After the acquisition of the GC-MS experiment, we obtain a raw file of several billion data points. Each of the data points is represented by three values: *MZ*, *RT* and *Intensity*. We can incorporate a preprocessing step to filter out the noise and zero intensity data points as they do not bear any significant information.

For noise filtering, we incorporate intensity threshold filtering. To estimate a filtering value, we use a method based on the histogram construction of the intensity values. We produce a histogram of the logarithms of the intensity values and find the maximum of the histogram. This value is set as the threshold to

filter out low-intensity and noisy data points. Based on the composition of the sample and the resolution of the log-intensity histogram, around 10–30% of the data points will be filtered out. Thus significantly reducing the number of data to process. The noise thresholding is visualized in Fig. 1.

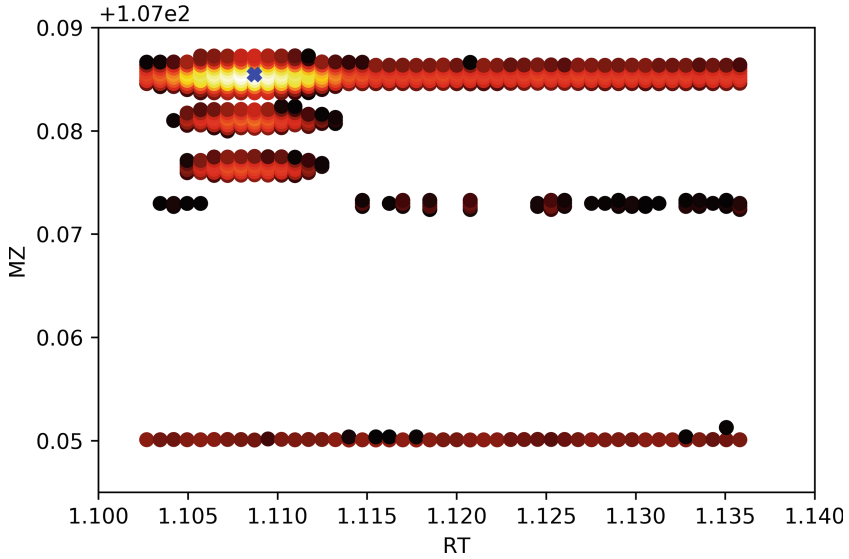
To utilize the clustering of the data points, the whole data space needs to be divided into smaller regions of an adequate number of data points. Thus, it can be processed with good memory and time demands. For our method, we intend to use a floating 2D window with data overlaps. The overlaps of the windows must be broad enough so they can cover a whole expected single peak region. The span of the overlap must be higher than the expected peak span in both the MZ and RT axis, thus ensuring that the peak regions in the overlaps of the window would completely fit one of the windows.



\* **Fig. 1.** Noise threshold identification for concentration of 500 ng/ml.

### 2.3 ROI Identification

To process and apply clustering algorithms, we do a preprocessing of the data. We segment the data into the regions of interest. Each region will contain only points within a region of the local peak values. First, the local maxima are identified based on the euclidean distance. Each point is labeled based on the nearest local maxima in the neighborhood. In the next iteration, the maximas will move towards the closest local maxima in their neighborhood, enlarging and joining their areas. After there is no change in the regional distribution, the regions of interest are extracted based on the labels of the points, thus belonging to the local maxima. Each of these regions is an independent area of measure, and there will be no peak area overlap between the areas. These regions do not have to have orthogonal dimensions. An example of one of the regions of interest with a peak value marked is shown in Fig. 2.



**Fig. 2.** Test region with annotated (cross) peak value displayed in transformed space.

## 2.4 Space Transformation

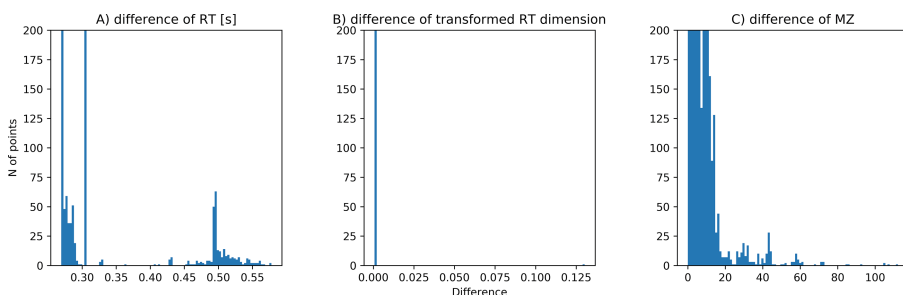
To utilize the known clustering algorithms, there is a need to transform the dataspace accordingly. First of all, we have a 2D plane with values of different dimensions. The acquisition gives the continuity of the data points by the scans. Each scan contains one RT value and a set of measured MZ values of the ion fragments. The distribution of the first difference of consecutive points in the MZ dimension is shown in Fig. 3C. The difference of consecutive scans is shown in Fig. 3A. To apply clustering kernels that are defined by the distance value, we must ensure that the distance value in both axes accurately represents the consecutive of the datapoints. The dispersion of the differences of consecutive points should be the same in both dimensions.

To ensure the consecutive RT dimension, we can substitute the RT values with a number of the corresponding scan in order of acquisition. Thus the difference of this axis will be equal to one. This substitution allows to scale this axis accordingly to the MZ axis and provide the same distance value for consecutive points.

The consecutive data points in the MZ axis is not strictly ensured. The consecutive points are only the ones belonging to the same measured ion fragment. Sorting the MZ values in one scan by the value will provide a set of many consecutive points and the few of them, where the difference is higher because they belong to different ions. The consecutive points should be in a higher number as the density of the points in the areas belonging to the ions should be higher. The measurements done in the region with no ion are noisy and should occur sparsely.

To estimate the distance of the consecutive points in the MZ dimension, we compute the first differentiation of the ordered MZ values in several scans. We then acquire the median of that values and use it as a scaling factor for the other axis.

After this transformation, we obtain a data space of MZ ax and accordingly scaled RT ax, as seen in Fig. 3B. The distance between consecutive points in both dimensions should be approximately the same in this dataspace. This transformation allows us to use clustering methods based on the distance computation without the need to define the distance individually for each of the dataset dimensions.



**Fig. 3.** The differences between consecutive points.

## 2.5 Clustering

Clustering is the task of dividing the data points into groups so that the data in the same group are more similar than the data in other groups. In other words, it aims to segregate data into groups of similar traits. These groups we call clusters. There are many clustering methods with a diverse approach to finding the similarity. The decision to use the suitable algorithm depends on the data structure and the desired clusters characteristics. One segment of the data can be seen in Fig. 2. We decided that a blank space clearly separated the clusters. Inside regions, the density of points is much higher. Thus, we utilize density-based algorithms rather than one based on a simple distance measure like K-means. We decide to use the DBScan, Optics, and BIRCH. Another type of algorithm we want to utilize is based on the distribution of the data points, and we choose the Gaussian Mixture Model (GMM).

Each of the clustering methods was set to the initial state with eps or threshold to 0.01, and was tuned towards the better identification of the ground truths peak values.

**BIRCH** (balanced iterative reducing and clustering using hierarchies) is an unsupervised algorithm used for hierarchical clustering. It performs exceptionally well over large datasets. There is no need to set the number of clusters in advance, only a distance threshold between the sample and subclusters. The example of predicted groups for this method can be seen in Fig. 4A [13].



**DBSCAN** (Density-Based Spatial Clustering of Applications with Noise) is an unsupervised clustering algorithm. It does not have shape constraints about the clusters. It is based on the dense areas separated by the area of low density. Dense areas we call core points and all the points within the distance  $eps$  we will call core points too. Thus it forms separated clusters. There is no need to set the number of clusters in advance, and we only set the  $eps$  value and the minimal number of points to form a cluster. The example of predicted clusters for this method can be seen in Fig. 4B [5, 11].

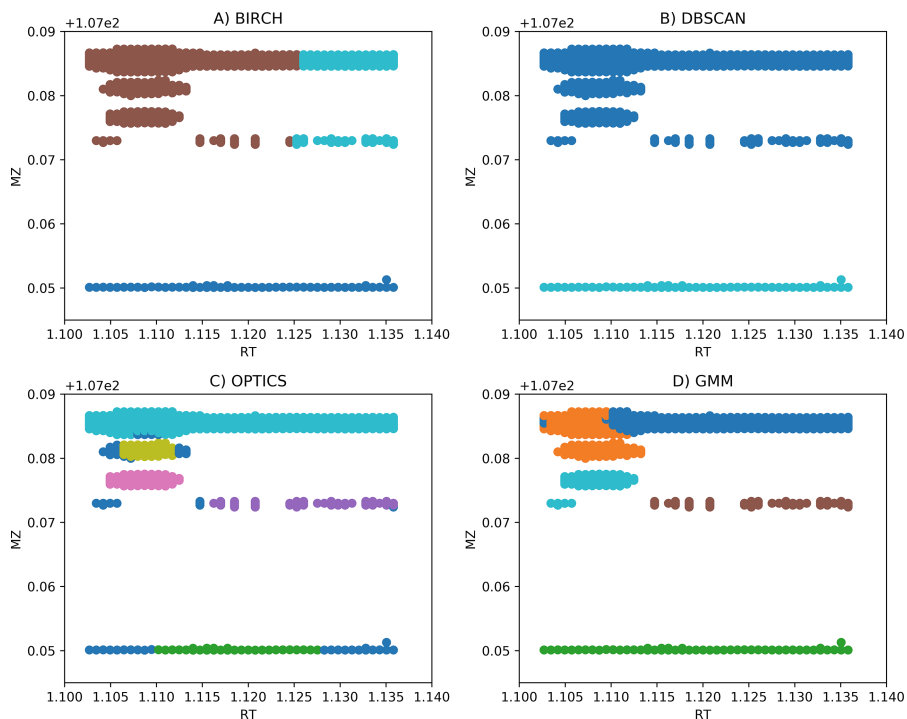
**OPTICS** (Ordering Points To Identify the Clustering Structure) can be viewed as a generalization of the DBSCAN method. It also depends on identifying the density in regions, but it keeps a cluster hierarchy for a variable neighborhood called the reachability graph. Cutting this graph produces the core set points and noise values. There is a need to set the maximum  $eps$  value and the minimum number of points to be considered as a cluster. The example of predicted clusters for this method can be seen in Fig. 4C [1].

**Gaussian Mixture Model** is a probabilistic method assuming that all the points came from a mixture of a finite number of processes with a gaussian distribution. It can be viewed as a generalization of the k-means with information about the covariance structure of the data. The data points are fitted to the set of the Gaussians, so the probability of the points coming from that Gaussians are maximized. For our purposes, we will be using a Variational Bayesian Gaussian Mixture. The example of predicted clusters for this method can be seen in Fig. 4D [3, 10].

### 3 Results

The space was divided as described in the method section for the selected dataset. We compute the regions of interest; thus, each area can be processed independently. Dimensions of each subspace were transformed; therefore, the distance of the consecutive points would have the same size in both dimensions. This allows using the clustering methods with defined one-dimensional distance measures. For each subspace, all of the clustering methods described above were executed. Each data point then obtains the identifier of the cluster it belongs to. The results from the subspaces were finally merged into one dataset. Each cluster containing less than 20 datapoints was relabeled as noisy one, as there is no data support to form a valid peak area.

Each algorithm provides a different number of clusters, as we can see in Table 2. Non-clustered points are points we consider noisy ones. The clustering also gives us the estimation of the noise in the measurement. There are significant differences between the used clustering methods. If we use the GMM or BIRCH algorithm, the noise points portion tends to be small compared to the OPTICS algorithm, which identifies about 17% of the points as the noisy one. Another



**Fig. 4.** Clustering algorithms performing on test region. Clusters were distinguished by color. (Color figure online)

**Table 2.** Number of clusters and their features for a dataset of concentration 500 ng/ml.

Method	N of clusters	Median cluster size	Non-clustered (noisy) points
BIRCH	164533	62	0.045%
DBSCAN	66783	213	2.507%
OPTICS	150890	43	17.304%
GMM	53150	179	0.045%

difference we see in the sizes of the obtained clusters. Birch and Optics tend to provide smaller clusters compared to DBScan and GMM method.

**Table 3.** Number of clusters with annotated peak.

Concentration	Method	Clusters with peak	Portion of identified peaks
1000 ng/ml (1057 peaks)	BIRCH	901	85.24%
	DBSCAN	886	83.82%
	OPTICS	813	76.92%
	GMM	873	82.59%
	Gridmass	889	84.11%
500 ng/ml (1057 peaks)	BIRCH	930	87.98%
	DBSCAN	912	86.28%
	OPTICS	845	79.94%
	GMM	902	85.33%
	Gridmass	866	81.93%
100 ng/ml (1046 peaks)	BIRCH	902	86.23%
	DBSCAN	885	84.61%
	OPTICS	835	79.83%
	GMM	876	83.75%
	Gridmass	699	66.83%

From the annotation of the dataset, we identify a set of ground-truth ions in the raw data. The *RT* and *MZ* values of these ions were marked as peak values. The annotated peaks were taken as the ground truth of the known features. Every peak should be clustered independently to its own area. From each of the identified clusters, we obtain the maximum intensity point. This point should be considered as a peak value. We can compare specified peak values with the ground truth. From the whole dataset, none of the algorithms was able to identify all of them, as we can see in the Table 3. The 100% identification was not expected as some of the annotated values were identified by hand with much guardedness. For each of the ions, there is a set of 20 peak values ordered with the relative abundance. For comparison, we provided the results obtained by the Gridmass method [12] with the maximum *MZ* tolerance set to 0.001. As we can see, the clustering methods were able to identify almost the same portions of the features as the Gridmass method. In the lower concentrated sample, it even outperformed. The portion of correctly identified features was not much decreasing with the dilution of the sample. Although the number of features is much higher from clustering methods than the GridMass, it considers more data properties.

## 4 Conclusion

We presented the proof-of-concept to identify the peaks and their features in raw GC-MS measurement. The integral step of this method to apply clustering methods is the space division and transformation.

We presented a method of raw space segmentation into the independent regions of interest. This method is based on a seeding algorithm and region merg-

ing. Each region contains local maxima and its belonging points. This ensures that each point is clustered within its local maxima; thus, the regions can be processed independently. It ensures that the data to the corresponding peak will be processed together and inseparably.

After the space division into processable regions, we transform the space to provide approximately the same absolute distance size between consecutive points in each dimension. This allows defining the distance measures for the clustering algorithm as a one-dimensional parameter.

After the dataspace transformation, we presented that the clustering algorithms can identify the peaks and their regions in raw data. This method can process the mass spectrometry measurement in 2D without centroiding or signal processing methods. These detectors consider the neighborhood of the peak and can label the whole peak area.

**Acknowledgment.** We make the test dataset and proof-of-concept available at [10.5281/zenodo.6337968](https://doi.org/10.5281/zenodo.6337968).

Authors thanks to Research Infrastructure RECETOX RI (No LM2018121) financed by the Ministry of Education, Youth and Sports, and Operational Programme Research, Development and Innovation - project CETOCOEN EXCELLENCE (No CZ.02.1.01/0.0/0.0/17.043/0009632) for supportive background.

## References

1. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: OPTICS. ACM SIGMOD Record **28**(2), 49–60 (1999). <https://doi.org/10.1145/304181.304187>. <https://dl.acm.org/doi/abs/10.1145/304181.304187>
2. Castillo, S., Gopalacharyulu, P., Yetukuri, L., Orešič, M.: Algorithms and tools for the preprocessing of LC-MS metabolomics data. Chemometr. Intell. Lab. Syst. **108**(1), 23–32 (2011). <https://doi.org/10.1016/J.CHEMOLAB.2011.03.010>
3. Constantinopoulos, C., Titsias, M.K., Likas, A.: Bayesian feature and model selection for Gaussian mixture models. IEEE Trans. Pattern Anal. Mach. Intell. **28**(6), 1013–1018 (2006). <https://doi.org/10.1109/TPAMI.2006.111>
4. Dixon, S.J., Brereton, R.G., Soini, H.A., Novotny, M.V., Penn, D.J.: An automated method for peak detection and matching in large gas chromatography-mass spectrometry data sets. J. Chemometr. **20**(8–10), 325–340 (2006). <https://doi.org/10.1002/CEM.1005>
5. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. Technical report (1996). [www.aai.org](http://www.aai.org)
6. Katajamaa, M., Orešič, M.: Data processing for mass spectrometry-based metabolomics. J. Chromatogr. A **1158**(1–2), 318–328 (2007). <https://doi.org/10.1016/J.CHROMA.2007.04.021>
7. McDonnell, L.A., van Remoortere, A., de Velde, N., van Zeijl, R.J., Deelder, A.M.: Imaging mass spectrometry data reduction: automated feature identification and extraction. J. Am. Soc. Mass Spectrom. **21**(12), 1969–1978 (2010). <https://doi.org/10.1016/J.JASMS.2010.08.008>
8. Melymuk, L., Diamond, M.L., Riddell, N., Wan, Y., Vojta, Š., Chittim, B.: Challenges in the analysis of novel flame retardants in indoor dust: results of the

- INTERFLAB 2 interlaboratory evaluation. *Environ. Sci. Technol.* **52**(16), 9295–9303 (2018)
9. Morris, J.S., Coombes, K.R., Koomen, J., Baggerly, K.A., Kobayashi, R.: Feature extraction and quantification for mass spectrometry in biomedical applications using the mean spectrum. *Bioinformatics* **21**(9), 1764–1775 (2005). <https://doi.org/10.1093/BIOINFORMATICS/BTI254>. <https://academic.oup.com/bioinformatics/article/21/9/1764/408956>
  10. Roberts, S.J., Husmeier, D., Rezek, I., Penny, W.: Bayesian approaches to Gaussian mixture modeling. *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(11), 1133–1142 (1998). <https://doi.org/10.1109/34.730550>
  11. Schubert, E., Sander, J., Ester, M., Kriegel, H.P., Xu, X.: DBSCAN revisited, revisited. *ACM Trans. Database Syst. (TODS)* **42**(3) (2017). <https://doi.org/10.1145/3068335>. <https://dl.acm.org/doi/abs/10.1145/3068335>
  12. Treviño, V., et al.: GridMass: a fast two-dimensional feature detection method for LC/MS. *J. Mass Spectrom.* **50**(1), 165–174 (2015). <https://doi.org/10.1002/jms.3512>. <http://doi.wiley.com/10.1002/jms.3512>
  13. Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: a new data clustering algorithm and its applications. *Data Min. Knowl. Discov.* **1**(2), 141–182 (1997). <https://doi.org/10.1023/A:1009783824328>. <https://link.springer.com/article/10.1023/A:1009783824328>



# Spolmap: An Enriched Visualization of CRISPR Diversity

Christophe Guyeux<sup>1</sup>(✉), Guislaine Refrégier<sup>2</sup>, and Christophe Sola<sup>2</sup>

<sup>1</sup> Femto-ST Institute, UMR 6174 CNRS, Université de Bourgogne Franche-Comté, Besançon, France

[christophe.guyeux@univ-fcomte.fr](mailto:christophe.guyeux@univ-fcomte.fr)

<sup>2</sup> Université Paris-Saclay, Paris, France

**Abstract.** In the study of the evolution of various bacteria, the content of the CRISPR locus has proven to be quite useful. This locus has been made famous because it allows for simple and inexpensive genome editing. And bacteriologists are used to studying this locus, through tools such as spoligotyping, in order to experimentally be able to determine the lineage or even the sub-lineage of a given strain, and to deduce an optimal antibiotic cocktail. The problem is that the study of the content of this locus is very often delicate and difficult. Therefore, we propose in this paper a new way of representing them, which makes sense biologically speaking, and which allows a simplified and enriched study of the CRISPR content. After explaining how to extract this locus from Whole Genome Sequencing data, we propose an embedding of this locus in a high dimensional space, followed by a reduction to dimension 2, which makes sense of the content. This method is applied to the case of the *Mycobacterium tuberculosis* complex, and a discussion is proposed to list the advantages of this approach.

## 1 Introduction

Tuberculosis remains one of the most deadly diseases in the world today, and its incidence has even increased in recent years following the COVID epidemic. This disease is caused by a bacterium called *Mycobacterium tuberculosis*, which was described more than 100 years ago. But since the discovery of the first antibiotics, little progress has been made in the fight against this bacterium, and the development of resistant to multi-resistant strains is certainly a problem. Therefore, any additional knowledge on this bacterium and its evolution is welcome.

To a lesser extent, this can also be established for other diseases such as salmonella or legionella. And the various bacteria involved in these diseases have the particularity to be studied through the content of the CRISPR locus. In some of them, such as the bacteria of the *Mycobacterium tuberculosis* complex (MTC), this locus is no longer active and now only faces deletions. In this case, the difference between the current content and the ancestral content [10, 11] is a specific characteristic of a strain, which allows it to be classified, for example, in a particular lineage. This barcode allowing the analysis of strains based on their

content in CRISPR is called spoligotype in *M. tuberculosis*. In other bacteria, this locus remains active, but it also contains sub-patterns which can be studied to gain knowledge. But in any case, the deciphering and analysis of this content is still a delicate task, and there is currently no tool to help study these complex motifs.

The objective of this paper is to propose a new way to represent these CRISPR motifs, illustrating it in the case of MTC. The idea is mainly to plunge them intelligently into an N-dimensional space, and then to do a quality dimension reduction, to obtain a planar view that makes sense biologically speaking, and that is easier to study. The various steps required to achieve this result are fully detailed, from downloading the genome directly from the sequencing, to extracting the lineage information and the CRISPR locus content. The latter is obtained here by a De Bruijn graph approach, after extraction of the reads of interest, and requires a manual step. Finally, the embedding is also fully detailed.

The remainder of this article is as follows. In the next section, basic recalls regarding the CRISPR locus and the spoligotyping technics is recalled. Section 3 is devoted to the proposed approach, which is fully detailed. It is experimented in Sect. 4 in the case of the *Mycobacterium tuberculosis* complex. This result section is followed by a discussion that extends this work. This article ends by a conclusion section, in which the contribution is summarized and intended future work is outlined.

## 2 Basic Recalls

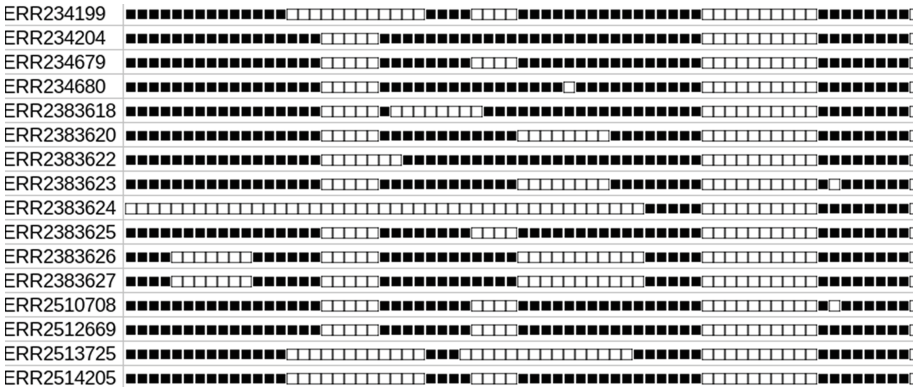
The CRISPR locus of *Mycobacterium tuberculosis* complex (MTC), the agent of tuberculosis (TB), was first described in 1993 as the “Direct Repeat” locus [9, 20]. It consists of 36 nucleotide-repeats interspersed with single spacers averaging 37 nt (range: 25–45 nt). The repeats were quickly referred to as direct repeats and abbreviated as such (DR), and the sequences of a single spacer + a DR were called direct-variant repeats (DVR). The first two isolates sequenced (*M. tuberculosis* H37Rv and *M. bovis* BCG) yielded 43 different spacer sequences. The detection of their presence/absence led to the development of the innovative method of “spoligotyping” [15]. This method has become very popular because of its ease of implementation and its digital format. It has indeed allowed us to decipher the structure of the global MTC population [4]. More recently, whole genome sequencing (WGS) studies have indeed confirmed that for the 6 major human lineages (L1 to L6) and many sub-lineages, the spoligotypic signature allows an approximate taxonomic assignment [16]. However, some generic signatures remain either meaningless, imprecise, or convergent, which largely justifies the use of SNPs as preferred taxonomic markers, whether at the global or national level [6], or for L4 [22], L1 [19], or L2 [12, 21].

As in other species with functional CRISPRs, this locus is accompanied by a set of CRISPR-associated genes (cas). Their number and nature make the MTC CRISPR type fall into the Type III-A group in the CRISPR-Cas taxonomy [17]. The CRISPR-Cas locus has recently been shown to be active in the H37Rv

system [26]. Yet some or all of the region is deleted in several MTC sublines [8]. Another important question is whether deletion of some of the cas genes in the CRISPR-Cas locus can promote genomic instability in some epidemic strains of MTC [23].

The genomic diversity of the CRISPR locus was studied in detail as early as 2000 in a study by J. van Emden et al. showing that spacer duplications, spacer variations and IS6110 insertion sites could be found in the different phylogenetic lineages of TCM [25]. However, it involved a very small sample ( $n = 34$ ) and did not include any investigation of cas genes [3, 7]. Understanding the evolutionary dynamics of this locus now requires exploration of the CRISPR-Cas region on an extensive data set.

The classical *in vitro* approach of spoligotyping lists the presence or absence of a well-known list of spacers in a sample. This robust method has been widely applied *in vitro* [15]. However, this approach did not explore many features, such as whether the order of spacers is different in one strain or the other. It also did not reveal whether there was duplication of any part of the locus. Finally, it did not provide information on the presence of insertions such as IS6110, nor on the existence of single nucleotide polymorphisms (SNPs) in its direct repeats or spacers. This masks potential functionally important changes in the loci, and makes it impossible to conduct in-depth evolutionary studies. New *in silico* approaches (SpolPred, SpoTyping) have been developed to produce spoligotypes from genome reads [5]. Although these methods reveal the presence/absence of spacers in a similar manner, they have the same limitations as *in vitro* spoligotyping techniques. Last, but not least, their exploitation is frequently difficult due to the existence of numerous patterns that are difficult to relate together.



**Fig. 1.** Example of spoligotypes of Lineage 5, defined by their accession numbers.



### 3 The Proposed Approach

Firstly, we have to download the genomes of interest, in the form of a Sequence Read Archive (SRA), for example with the `fastq-dump` command from the NCBI SRA Toolkit [1].

The first key step is then to extract the spoligotype from this SRA. Various tools exist in the literature [5], but none of them are suitable for our approach. Some of them require assembled genomes; however, the CRISPR locus is rich in repeated sequences (DR and IS6110), and its very difficult assembly often leads to gross errors. Others are compatible with SRA-type inputs, but the quality of the spoligotypes produced has proven insufficient for our needs. The problem is that these tools are not specific to MTBC: some just tell if a CRISPR locus is present, while others try to find the content of the locus without a priori knowledge of the spacer sequences to find. This is why we have chosen to follow the new approach proposed in [13].

We first build a blast database from the reads contained in the SRA file, then we blast the sequences of interest (spacers, DR, and CAS genes). To increase the diversity of the retained sequences and to guard against the discarding of reads containing mutations, we transform these reads that match into  $k$ -mers, where  $k$  is three-fourths the size of the reads. We then construct a De Bruijn graph from these  $k$ -mers, in which the nodes are these sequences, and there is an edge from a node  $i$  to a node  $j$  if and only if a suffix of  $i$  is a prefix of  $j$ .

We then traverse each of the connected components of this graph  $G$ . A first node is drawn at random, and we traverse the related component from vertex to vertex, as long as it is possible. The vertices thus traversed are removed from the  $G$  graph, and this traversal produces by concatenation of the sequences a part of the CRISPR locus. We then identify the elements of this part using the list of sequences of interest (spacers, DRs, CAS and IS6110 genes), and we thus obtain a first contig with the details of its content. This process is repeated until the vertices of  $G$  are exhausted (the process necessarily has an end). The contigs are then sorted by size, and the final assembly is done by hand.

Note that, with few exceptions, there is always at least one IS6110 in the CRISPR locus. Given the size of this insertion sequence, compared to  $k$ , as well as its large number of copies in the genome, contig construction by iterating on  $G$  necessarily stops when an IS6110 is encountered. Similarly, we have recently shown the existence of duplicated spacers (singly or in tandem), and these duplications are also a cause of stopping contig reconstruction [20]. These elements explain why a human final step is required.

Once the CRISPR of the strain has been reconstructed and the spoligotype deduced, we still need to determine the lineage of the genome. This is done by taking the list of SNPs per lineage from Coll [6], then extracting from the h37Rv reference a 40 base pair sequence around the SNP position, and blasting the result onto the database defined above. A majority vote is then needed to assign a lineage to this strain.

Let us now assume that the set of our spoligotypes of interest contains a total of  $N$  different gaps, e.g. (15, 26); (30, 34); (51, 60) for the first spoligotype

in Fig. 1. The next step is to transform each spoligotype into a point in an  $N$ -dimensional space, as follows. The gaps are sorted according to the lexicographic order, and an integer from 1 to  $N$  is then assigned to each gap positioned according to this order. The vector corresponding to the considered spoligotype is then constituted as follows: we place a 1 at each associated gap position, and a 0 everywhere else. In this way, we obtain a binary vector of size  $N$ , where each distinct spoligotype has a different position in space. In this  $N$ -dimensional space, points close in Manhattan distance correspond to similar spoligotypes. It remains then to make a reduction to dimension 2, using the t-SNE algorithm [24].

The implementation has been realized in Python 3.10, and an interface provided in Tk is available upon reasonable request.

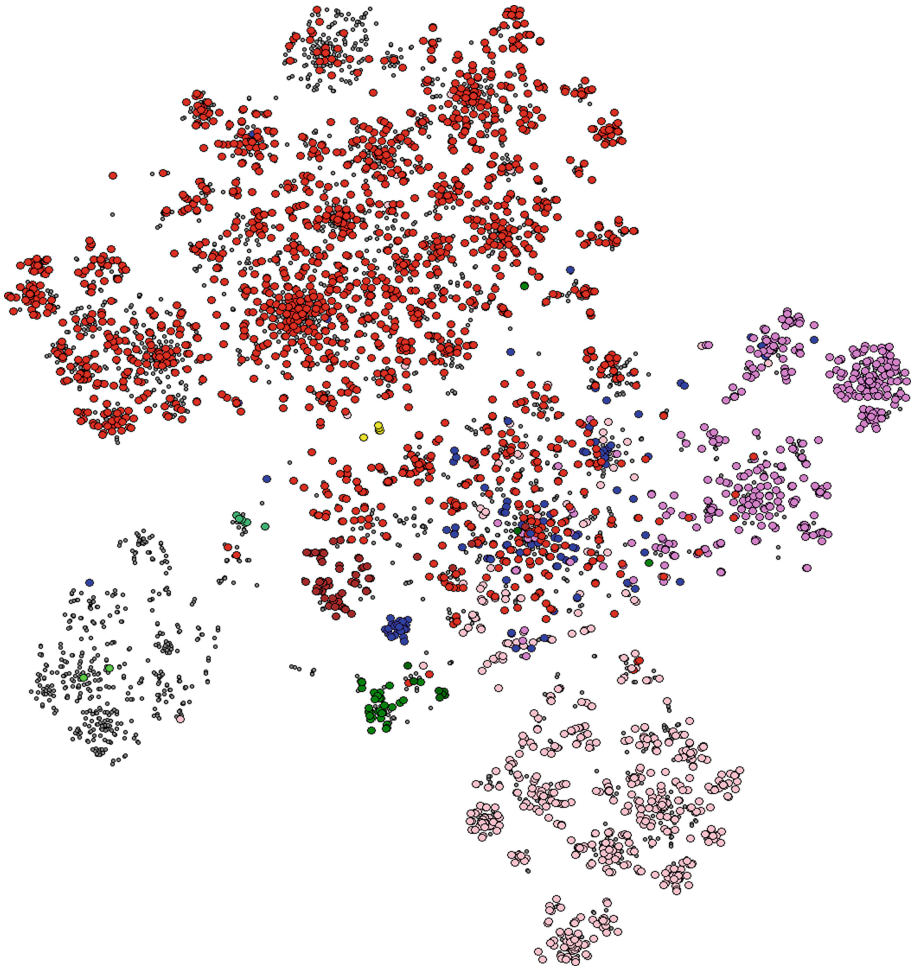
## 4 Obtained Results

An example of what Spolmap can lead to is shown in Fig. 2 in the MTC case. In this figure, each point corresponds to one strain (in WGS genome form), while the color of these points is made according to the strain lineage. In this picture, we can find:

- the lineage 1, indo-oceanian, in pink at the bottom of the picture;
- the lineage 2, Beijing, in two blue clusters: a spread out cluster at the center of the figure, corresponding to ancient strains, and a concentrated one at its bottom left, for the modern ones;
- the Middle-East lineage 3 in violet, on the right part of the cloud;
- the Americano-European lineage 4 in red, occupying the upper half of the figure;
- the two African lineages 5 and 6, respectively in brown and dark green, a little bit off-center;
- the Ethiopian lineage 7 in yellow, alone at the center of the cloud;
- the animal strains in green, a few circles in the bottom left part of the cloud.

Many conclusions can be drawn from this point cloud obtained from the spoligotypes. First, there are as many clusters as there are lineages, with sub-clusters associated with sub-lineages. Some lineages are very well separated and present a really pure cluster, such as lineages 5, 6 and 7. We also find, in the upper right part, lineages 2 to 4, and in the lower left part, lineages 1, 5, 6 and animal, and we know that these two subgroups are phylogenetically separated. The clusters of lineages 1 to 4 extend to the center of the cloud, arguing for a common origin of the tuberculosis complex, whose ancestor is probably *M. canettii*.

In some sublineages, the corresponding subcluster is only partially colored, suggesting a poor definition of said sublineage (an overly restrictive lineage SNP). This is evident in the circular cluster at the top of lineage 4, for example. We also see a whole big gray cluster with a few green dots in it, which would tend to show our very poor knowledge of animal TB.



**Fig. 2.** A 2D visualization of the MTC's spoligotypes (Color figure online)

Another lesson is that, in general, the complex is well described by the existing lineages: apart from the animal lineage, there are no large new grey clusters to investigate. However, there are several small grey clusters, the size of the lineage 8 cluster, which are isolated, such as the small ten points between the animal lineage and lineage 6 (in dark green). These small clusters probably reflect small exotic lineages, which should be further investigated to have a full understanding of the TB mycobacterial complex.

Finally, it is undeniable that the SNP-based lineage data and the spoligotype hole data are strongly correlated, arguing for a co-occurrence of these two evolutionary mechanisms at the same time.

## 5 Discussion

The spoligotype has long been considered useful for many lineage identifications, with for example the absence of spacers 18–22 on the one hand, and 51–60 on the other, as a definition of Lineage 5, cf. Fig. 1. However, its interpretation is often quite delicate, if one is content to focus on a linear representation. We have shown that a representation of the latter in high dimension followed by a reduction to the 2-dimension reveals something quite coherent, and a vision both summarized and useful.

We also saw that this approach made it possible to find new lineages or sub-lineages, to highlight definitional problems in the latter, as well as poorly explored areas in the diversity of the considered species. Such an approach is not limited to the bacteria that cause tuberculosis. It can potentially be applied to any bacterial species with a CRISPR locus that is no longer functional (and therefore, for which the number of spacers is finite), if exist. It can also be used in bacteria whose locus is active, but for which subgroups of spacers appear, and allow a use for characterization, such as in salmonella or legionella (or, in some plant pathogenic bacteria).

Note that we have only used an elementary definition of distance between two spoligotypes, and other choices are possible. Similarly, t-SNE is not the only recent tool for dimension reduction, and techniques such as the so-called Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP, [18]) could lead to other representations, equally useful and complementary.

Finally, dimension reduction techniques are often coupled with outlier detection methods [2, 14], and the latter seem promising either to rule out a strain that does not belong to the complex under consideration, or to highlight new lineages that were previously unknown (recall that lineages 8 and 9 have been discovered in the last three years: there are probably new things to discover).

## 6 Conclusion

Based on spoligotyping in *M.tuberculosis*, we have proposed a new way of representing the CRISPR locus, which both makes biological sense, and makes the study easier and more thorough. This approach has been fully detailed, from genome upload to locus extraction, through plotting in high-dimensional space and to the final dimension reduction step. This approach allows to detect outliers, to show the diversity of the studied strains and their respective relationships. It also allows to detect new lineages or sub-lineages, and to highlight possible inconsistencies.

For our next works, we wish to make this tool accessible through a neat interface, and propose versions for tuberculosis, salmonella and legionella. We then wish to integrate all available genomes (more than 100000 genomes in the case of *M. tuberculosis*), and then to search for unknown lineages. Finally, we wish to integrate this representation in a larger and complete tool, including for example the determination of lineages and MIRU-VNTRs in *M.tuberculosis*.



## References

1. SRA toolkit development team. <https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=software>. Accessed 16 Mar 2022
2. Ranga Suri, N.N.R., Murty M, N., Athithan, G.: Outlier detection. In: Outlier Detection: Techniques and Applications. ISRL, vol. 155, pp. 13–27. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-05127-3\\_2](https://doi.org/10.1007/978-3-030-05127-3_2)
3. Bland, C., et al.: Crispr recognition tool (crt): a tool for automatic detection of clustered regularly interspaced palindromic repeats. *BMC Bioinform.* **8**(1), 1–8 (2007)
4. Brudey, K., et al.: Mycobacterium tuberculosis complex genetic diversity: mining the fourth international spoligotyping database (spolddb4) for classification, population genetics and epidemiology. *BMC Microbiol.* **6**(1), 1–17 (2006)
5. Coll, F.: Spolpred: rapid and accurate prediction of mycobacterium tuberculosis spoligotypes from short genomic sequences. *Bioinformatics* **28**(22), 2991–2993 (2012)
6. Coll, F., et al.: A robust SNP barcode for typing mycobacterium tuberculosis complex strains. *Nat. Commun.* **5**(1), 1–5 (2014)
7. Faksri, K., Xia, E., Tan, J.H., Teo, Y.-Y., Ong, R.T.-H.: In silico region of difference (RD) analysis of mycobacterium tuberculosis complex from sequence reads using RD-analyzer. *BMC Genom.* **17**(1), 1–10 (2016)
8. Freidlin, P.J., et al.: Structure and variation of CRISPR and CRISPR-flanking regions in deleted-direct repeat region mycobacterium tuberculosis complex strains. *BMC Genom.* **18**(1), 1–14 (2017)
9. Groenen, P.M.A., Bunschoten, A.E., van Soolingen, D., van Erftbden, J.D.A.: Nature of DNA polymorphism in the direct repeat cluster of mycobacterium tuberculosis; application for strain differentiation by a novel typing method. *Mol. Microbiol.* **10**(5), 1057–1065 (1993)
10. Guyeux, C., Al-Nuaimi, B., AlKindy, B., Couchot, J.-F., Salomon, M.: On the reconstruction of the ancestral bacterial genomes in genus mycobacterium and Brucella. *BMC Syst. Biol., IWBBIO 2017 Special Issue* **12**(5), 100 (2018)
11. Guyeux, C., Salomon, M., Al-Nuaimi, B., AlKindy, B., Couchot, J.-F.: Ancestral reconstruction and investigations of genomic recombination on some pentapetalae chloroplasts. *J. Integrative Bioinform.* \*, 20180057 (2019)
12. Guyeux, C., Senelle, G., Refrégier, G., Bretelle-Establet, F., Cambau, E., Sola, C.: Connection between two historical tuberculosis outbreak sites in Japan, Honshu, by a new ancestral mycobacterium tuberculosis l2 sublineage. *Epidemiol. Infect.* **150**, e56 (2022)
13. Guyeux, C., Sola, C., Noûs, C., Refrégier, G.: Crisprbuilder-tb: “crispr-builder for tuberculosis”. Exhaustive reconstruction of the CRISPR locus in mycobacterium tuberculosis complex using SRA. *PLOS Computational Biology* **17**(3), 1–21 (2021)
14. Hodge, V., Austin, J.: A survey of outlier detection methodologies. *Artif. Intell. Rev.* **22**(2), 85–126 (2004)
15. Kamerbeek, J., et al.: Simultaneous detection and strain differentiation of mycobacterium tuberculosis for diagnosis and epidemiology. *J. Clin. Microbiol.* **35**(4), 907–914 (1997)
16. Kato-Maeda, M., et al.: Strain classification of mycobacterium tuberculosis: congruence between large sequence polymorphisms and spoligotypes. *Int. J. Tuberculosis Lung Disease* **15**(1), 131–133 (2011)

17. Makarova, K.S., Wolf, Y.I., Koonin, E.V.: Classification and nomenclature of CRISPR-CAS systems: where from here? *CRISPR J.* **1**(5), 325–336 (2018)
18. McInnes, L., Healy, J., Melville, J.: Umap: uniform manifold approximation and projection for dimension reduction. arXiv preprint [arXiv:1802.03426](https://arxiv.org/abs/1802.03426) (2018)
19. Palittapongarnpim, P., et al.: Evidence for host-bacterial co-evolution via genome sequence analysis of 480 THAI mycobacterium tuberculosis lineage 1 isolates. *Sci. Rep.* **8**(1), 1–14 (2018)
20. Refrégier, G., Sola, C., Guyeux, C.: Unexpected diversity of crispr unveils some evolutionary patterns of repeated sequences in mycobacterium tuberculosis. *BMC Genomics* **21**(1), 1–12 (2020)
21. Shitikov, E., et al.: Evolutionary pathway analysis and unified classification of east Asian lineage of mycobacterium tuberculosis. *Sci. Rep.* **7**(1), 1–10 (2017)
22. Stucki, D., et al.: Mycobacterium tuberculosis lineage 4 comprises globally distributed and geographically restricted sublineages. *Nat. Genet.* **48**(12), 1535–1543 (2016)
23. Tsolaki, A.G., et al.: Functional and evolutionary genomics of mycobacterium tuberculosis: insights from genomic deletions in 100 strains. *Proc. Natl. Acad. Sci.* **101**(14), 4865–4870 (2004)
24. Van der Maaten, L., Hinton, G.: Visualizing data using T-SNE. *J. Mach. Learn. Res.* **9**(11), 2579–2605 (2008)
25. Van Embden, J.D.A., Van Gorkom, T., Kremer, K., Jansen, R., Van der Zeijst, B.A.M., Schouls, L.M.: Genetic variation and evolutionary origin of the direct repeat locus of mycobacterium tuberculosis complex bacteria. *J. Bacteriol.* **182**(9), 2393–2401 (2000)
26. Wei, W., et al.: Mycobacterium tuberculosis type III-A CRISPR/Cas system CRRNA and its maturation have atypical features. *FASEB J.* **33**(1), 1496–1509 (2019)



# How to Compare Various Clustering Outcomes? Metrics to Investigate Breast Cancer Patient Subpopulations Based on Proteomic Profiles

Joanna Tobiasz<sup>1,2</sup>  and Joanna Polanska<sup>1</sup> 

<sup>1</sup> Department of Data Science and Engineering, Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland

{joanna.tobiasz, joanna.polanska}@polsl.pl

<sup>2</sup> Department of Graphics, Computer Vision and Digital Systems, Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland

**Abstract.** Breast cancer is a highly diverse disease. With the state-of-the-art methods of molecular studies, novel subgroups of breast cancer can be revealed. The proper identification of subtypes is crucial for treatment choice. Hence, further investigation of breast cancer subtypes is promising in terms of therapy tailoring. We applied various machine learning approaches to the set of protein level measurements to detect subpopulations of breast cancer patients. Those methods involved various dimensionality reduction techniques combined with clustering. The outcomes of those approaches depended on the algorithms involved and on their parameters. Hence, we proposed the methodology to compare the results of clustering algorithms when the proper number of groups is unknown. The used metrics based on the effect size measurements and allowed for the selection of the best machine learning approach. The values of the proposed pooled  $d$  measure varied from 1.6847 for the worst method to 2.0568 for the best one. The highest value was obtained for the custom DiviK approach. Potentially, the metrics can also serve for the proteomic characterization of differences between subtypes and the identification of novel biomarkers.

**Keywords:** Breast cancer · Machine learning · Proteomics · Clustering · Dimensionality reduction

## 1 Introduction

Breast cancer is a diverse disease with highly heterogeneous molecular characterization. Its subtypes vary in prognosis and therapy response. Proper diagnosis and subtype identification are crucial for treatment choice and planning.

In the early 2000s, Sørlie et al. [1] proposed a division of breast cancers into five intrinsic molecular subtypes: Luminal A, Luminal B, HER2-enriched, Basal, and Normal-like. This study led to the development of the PAM50 classifier [2], which allowed labeling a tumor with its intrinsic molecular subtype based on the gene expression microarray measurements. However, with the arrival of new technologies

for molecular profiling, it became possible to further investigate, extend, and modify well-established breast cancer subtype categorization.

Machine learning provides a variety of methods for clustering and feature extraction or selection. Those techniques can be successfully applied for large genomic or proteomic datasets to investigate the heterogenic and diverse structure of breast cancer. However, results of subtypes identification often distinctly differ between algorithms in terms of both patient assignment to clusters and the final number of clusters detected. Moreover, the clustering outcome strongly depends on the parameters used. Thus, a method to compare and select different grouping approaches and parameters is needed. However, this task seems to be challenging as the method should deal with an unbalanced number of cases among subpopulations, an unknown target number of subtypes, a huge number of features in comparison with observations, and various dissimilarity degrees between resulting clusters. Some of the difficulties result also from the biological background and disease characterization: for instance, basal breast cancers are expected to be far more isolated from other tumors, while luminal family members should tend to group together and then further split into smaller subgroups.

In this study, we aim to test various approaches for clustering evaluation as well as to propose a metrics that would handle the challenges mentioned above.

## 2 Materials

Data used in this study are the result of the Reverse Phase Protein Arrays (RPPA) experiment. This dataset was created as a part of The Cancer Genome Atlas Breast Invasive Carcinoma (TCGA-BRCA) project [3]. All results were downloaded from the Genomic Data Commons (GDC) Data Portal in the normalized form. Samples used for the RPPA measurements were collected from primary tumors of females suffering from breast cancer. TCGA provided molecular subtype labels obtained with the PAM50 classifier based on the gene expression microarrays [4]. We excluded the samples with missing PAM50 etiquette. Due to the insufficient number of normal-like cases, this group was not considered. We also excluded proteins which levels were missing for some patients due to the requirements of algorithms used in the further analysis. The remaining records were corrected for the batch effect with the ComBat tool [5]. Finally, the dataset consisted of expression levels for 166 proteins and 407 patients. The summary of patients included in the study regarding their PAM50 label is presented in Table 1.

**Table 1.** The numbers and percentages of patients included in the study concerning breast cancer subtype label given by the PAM50 classifier.

PAM50 subtype	No. patients	Percentage of patients [%]
Basal	86	21.13
HER2-enriched	50	12.28
Luminal A	173	42.51

(continued)



**Table 1.** (continued)

PAM50 subtype	No. patients	Percentage of patients [%]
Luminal B	98	24.08
<b>Total</b>	<b>407</b>	<b>100</b>

### 3 Methods

#### 3.1 Subtype Detection

To investigate the dataset composition and identify subpopulations of breast cancer patients, we tested various combinations of clustering algorithms and feature extraction or selection methods. We used the HDBSCAN [6], graph-based Louvain community detection [7], and custom Divisive intelligent k-means (DiviK) [8] algorithms for grouping. Those methods were applied either to the levels of all available proteins or to the reduced feature space. Features were extracted with Principal Components Analysis (PCA) to select top components explaining 90% of the variance in the data and with Uniform Manifold Approximation and Projection (UMAP) [9] performed on the PCA-reduced dataset. For the feature selection, we used the Gaussian Mixture Model (GMM) [10] decomposition of log<sub>2</sub>-scaled variances of protein levels. All tested combinations were presented in Table 2.

**Table 2.** Combinations of clustering algorithms and data dimensionality reduction methods used in the study. Abbreviations for each combination are written in italics. DiviK is marked with (\*) to indicate that the GMM-based filtration is built in each iteration of the algorithm.

Clustering	Feature engineering					
	No reduction		PCA		UMAP	
	Complete	GMM filtered	Complete	GMM filtered	Complete	GMM filtered
HDBSCAN	×	×	×	×	$H_{UMAP-C}$ ✓	$H_{UMAP-F}$ ✓
Louvain	$L_C$ ✓	$L_F$ ✓	$L_{PCA-C}$ ✓	$L_{PCA-F}$ ✓	×	×
DiviK*	×	✓	×	×	×	×

In the HDBSCAN algorithm, there was a need to assign classes to the cases which were left unclassified. We tested several methods for this prediction, based on:

1.  $H_{UMAP-C1}$ : Proximity in 2-dimensional UMAP
2.  $H_{UMAP-C2}$ : Proximity in the dataset with all protein levels (complete)
3.  $H_{UMAP-C3}$ : Proximity in the set of top principal components explaining 90% of the variance.

### 3.2 Comparison of Clustering Approaches

To evaluate clustering results and investigate proteomic profiles of identified subpopulations, we compared levels of each protein between the clusters with a one-way ANOVA procedure followed by the Tukey-Kramer post hoc tests. ANOVA results served for calculations of  $\eta^2$  effect size for each protein. The higher the  $\eta^2$  value, the better the cluster separation. The  $\eta^2$  metrics considers all clusters together, so its values do not provide insight into whether all clusters are well-separated, or just some of them are highly isolated.

Moreover, we calculated the values of modification of Cohen's  $d$  effect size to compare each obtained cluster versus all remaining ones considered jointly [11]. This measure was calculated based on the following equation:

$$d = \frac{\bar{x}_{subtype} - \bar{x}_{remaining}}{\sqrt{MS_{within}}} \quad (1)$$

Hence, for each protein, we obtained as many  $d$  values, as many subtypes were detected with a particular approach. As a result, for each method, we achieved a list of protein  $\eta^2$  values, and several lists of  $d$  values corresponding to subtypes.

To integrate  $\eta^2$  per method, we computed mean, median, and 3<sup>rd</sup> quartile of protein  $\eta^2$  values. To obtain a pooled value of  $d$  metrics per method, we proposed to assign the 3<sup>rd</sup> quartile of protein  $d$  absolute values to each subtype. Then, we projected the 3<sup>rd</sup> quartiles as a point in the  $k$ -dimensional space, where  $k$  was the number of subtypes detected. Finally, we calculated the pooled  $d$  value as a distance between the created point and the beginning of the coordinate system.

Moreover, we assessed the similarity between detected subtypes and PAM50 labels with the Dice coefficient. To further investigate the differences in outcomes of various method combinations, we referred the corresponding clusters to each other for the approaches with the lowest and the highest values of the pooled  $d$  metrics. We compared the values of  $d$  per protein for each subtype.

### 3.3 Biological Investigation

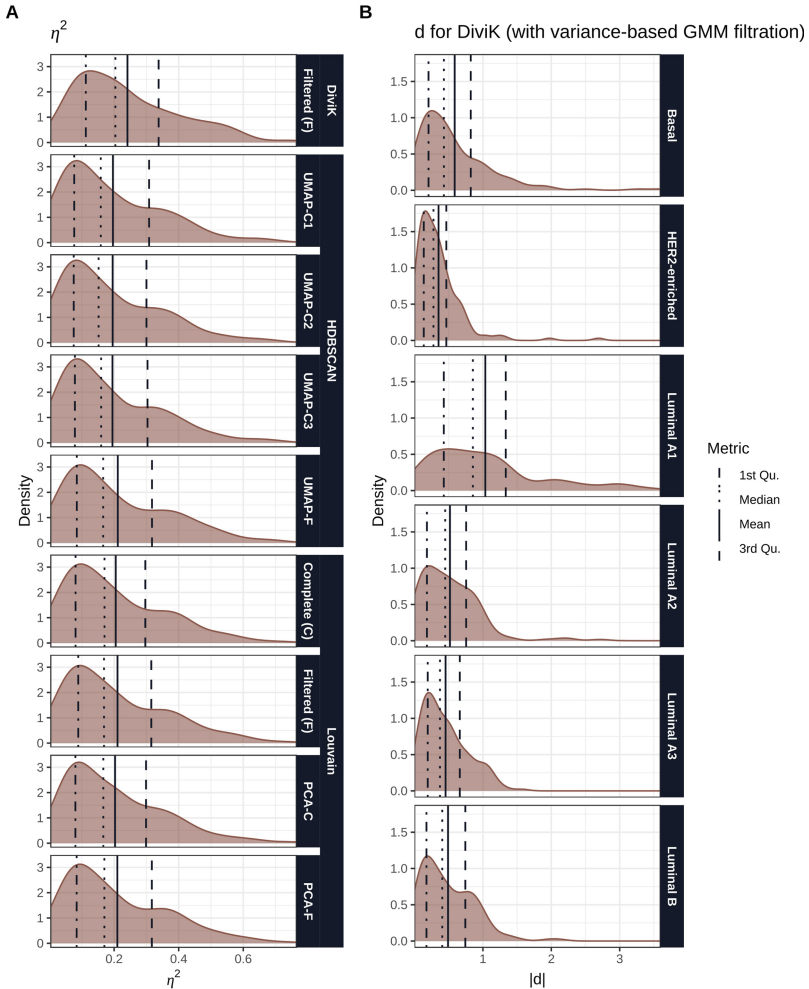
To biologically characterize each resulting cluster and evaluate the differences between the worst and the best approaches according to the pooled  $d$  metrics, we identified the proteins with significantly increased or decreased levels in each subtype compared to all remaining ones. Hence, we selected proteins with at least large or very large effect, so those with absolute values of  $d$  equal at least 0.8 or 1.2, respectively [11, 12]. We matched those proteins to the Kyoto Encyclopedia of Genes and Genomes (KEGG) database pathways in which they are involved [13] (accessed April 13, 2022).

## 4 Results

All HDBSCAN approaches without GMM filtration provided five clusters corresponding to Basal, HER2-enriched, Luminal A, and Luminal B subtypes. Luminal A cases

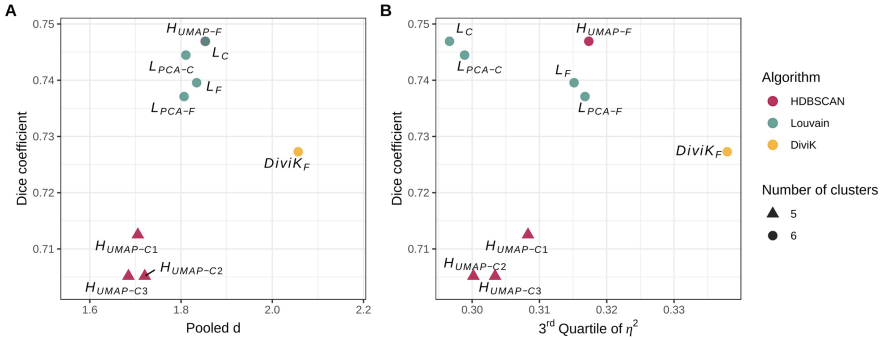
were divided into two subgroups. All the remaining combinations of methods (HDBSCAN with GMM, Louvain, and DiviK algorithms) gave six clusters. The clusters in all combinations corresponded to Basal, HER2-enriched, Luminal B, and three Luminal A subpopulations.

The distributions of  $\eta^2$  values per method are presented in Fig. 1A. The exemplary distributions of absolute  $d$  values for the DiviK method with built-in variance-based GMM filtration per subtype are presented in Fig. 1B.



**Fig. 1.** The distributions of metrics values with quartiles, median, and mean values marked with vertical lines. Panel A density plots showing distributions of  $\eta^2$  values per method. Panel B density plots showing distributions of absolute  $d$  values per subtype for the DiviK method with variance-based GMM filtration.

Obtained values of  $\eta^2$  quartiles and mean, pooled  $d$ , and Dice coefficient are presented in Table 3. Dice coefficient results are compared with pooled  $d$  and the 3<sup>rd</sup> quartile of  $\eta^2$  in Fig. 2.



**Fig. 2.** Values of pooled  $d$  (Panel A) and 3<sup>rd</sup> quartile of  $\eta^2$  (Panel B) compared with Dice coefficient for tested clustering approaches.

**Table 3.** Metrics values obtained with various combinations of feature dimensionality reduction methods and clustering algorithms.

Method	No. clusters	$\eta^2$				Pooled $d$	Dice
		Q1	Median	Mean	Q3		
$H_{UMAP-C1}$	5	0.0764	0.1587	0.1963	0.3083	1.7053	0.7125
$H_{UMAP-C2}$	5	0.0749	0.1519	0.1954	0.3002	1.7204	0.7052
$H_{UMAP-C3}$	5	0.0785	0.1598	0.1949	0.3034	1.6847	0.7052
$H_{UMAP-F}$	6	0.0844	0.1661	0.2113	0.3173	1.8529	0.7469
$L_C$	6	0.0806	0.1702	0.2050	0.2966	1.8534	0.7469
$L_{PCA-C}$	6	0.0800	0.1665	0.2030	0.2989	1.8105	0.7445
$L_F$	6	0.0889	0.1687	0.2105	0.3151	1.8342	0.7396
$L_{PCA-F}$	6	0.0839	0.1698	0.2100	0.3168	1.8066	0.7371
DiviK	6	0.1123	0.2040	0.2413	0.3379	2.0568	0.7273

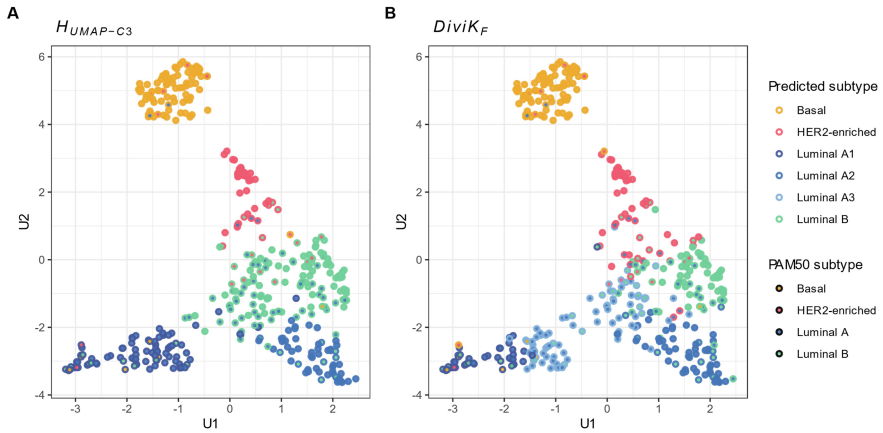
The results of the worst ( $H_{UMAP-C3}$ ) and the best (DiviK) approaches according to the pooled  $d$  values are also marked and compared to original PAM50 labels at the UMAP visualization in Fig. 3.

The primary difference between those two methods is that the DiviK algorithm provides an additional Luminal A3 cluster, containing cases included mainly in  $H_{UMAP-C3}$  Luminal B and Luminal A1 subtypes.

Those two contrasting approaches are further compared in Fig. 4. The protein values of  $d$  are referred to each other for corresponding Luminal subtypes: A1 versus A1, A2

versus A2, B versus B (respectively: Panels A, B, and C). Moreover, we compared the  $H_{UMAP-C3}$  Luminal B subtype with an additional Luminal A3 subtype given by DiviK (Panel D).

Total numbers of proteins with significantly higher or lower level for a certain subtype (with at least large or very large effects) are presented in Table 4 per subtype for the worst and the best approach. This table also contains the numbers of corresponding KEGG pathways.



**Fig. 3.** UMAP visualization with results of two clustering approaches referred to the original PAM50 subtype labels. Panel A corresponds to the worst approach according to the pooled  $d$  values (HDBSCAN algorithm with the proximity in the set of top principal components explaining 90% of the variance for prediction, preceded by UMAP dimension reduction -  $H_{UMAP-C3}$ ). Panel B corresponds to the best approach according to the pooled  $d$  values (DiviK algorithm with variance-based GMM filtration).

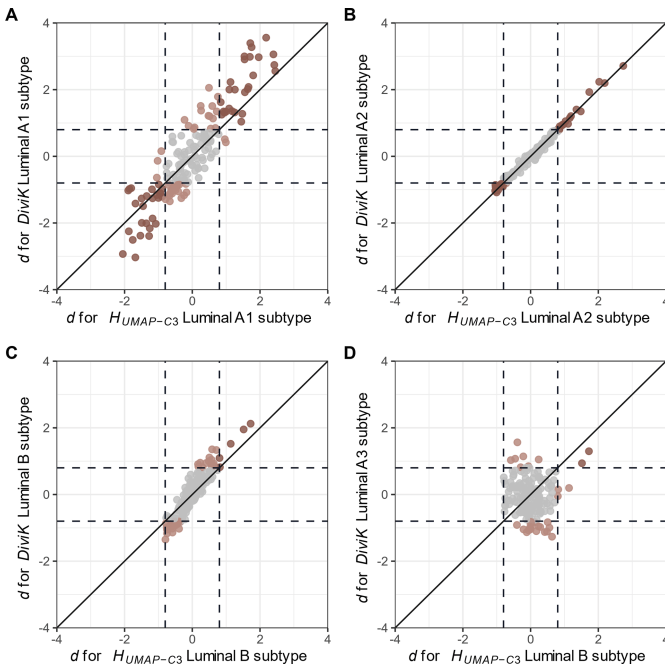
**Table 4.** Total numbers of proteins with at least large or very large effect size and corresponding KEGG pathways for the approaches with the lowest (HDBSCAN algorithm with the proximity in the set of top principal components explaining 90% of the variance for prediction, preceded by UMAP dimension reduction -  $H_{UMAP-C3}$ ) and the highest (DiviK algorithm with variance-based GMM filtration) pooled  $d$  values.

Subtype	At least large $ld$				At least very large $ld$			
	No. proteins		No. KEGG pathways		No. proteins		No. KEGG pathways	
	$H_{UMAP-C3}$	DiviK	$H_{UMAP-C3}$	DiviK	$H_{UMAP-C3}$	DiviK	$H_{UMAP-C3}$	DiviK
Basal	41	44	60	61	16	19	31	42

(continued)

**Table 4.** (continued)

Subtype	At least large ldl				At least very large ldl			
	No. proteins		No. KEGG pathways		No. proteins		No. KEGG pathways	
	$H_{UMAP-C3}$	DiviK	$H_{UMAP-C3}$	DiviK	$H_{UMAP-C3}$	DiviK	$H_{UMAP-C3}$	DiviK
HER2-enriched	12	9	47	31	5	4	27	23
Luminal A1	59	89	83	86	34	54	76	80
Luminal A2	37	38	65	64	6	7	4	4
Luminal A3	–	28	–	36	–	3	–	4
Luminal B	5	39	2	79	2	6	0	10



**Fig. 4.** Protein  $d$  values for the best (DiviK algorithm with variance-based GMM filtration) and the worst (HDBSCAN algorithm with the proximity in the set of top principal components explaining 90% of the variance for prediction, preceded by UMAP dimension reduction -  $H_{UMAP-C3}$ ) approach according to the pooled  $d$  metrics. Comparison of  $d$  values for the corresponding: Luminal A1 subtypes (Panel A), Luminal A2 subtypes (Panel B), Luminal B subtypes (Panel C), and DiviK Luminal A3 versus  $H_{UMAP-C3}$  Luminal B subtypes (Panel D). Dashed lines mark the threshold values for the large effect size, equal to  $-0.8$  and  $0.8$  [11]. Values for proteins with small or medium effect according to both approaches are marked in grey.

## 5 Discussion

Obtained results suggest the dataset should be divided into five or six clusters, with one cluster corresponding to each of the Basal, HER2-enriched, and Luminal B subtypes, and two or three subgroups for Luminal A cases.

Based on the  $\eta^2$  and  $d$  distributions we concluded that the 3<sup>rd</sup> quartile is an appropriate representation of metrics values for all proteins. It sufficiently reflects the impact of proteins which expression levels significantly vary between clusters. Still, it remains resistant to outliers.

The DiviK method obtained maximal values of all metrics based on  $\eta^2$  and  $d$ . However, in terms of Dice similarity coefficients, all methods that gave six clusters performed better. However, the aim was not to maximize the similarity to the original PAM50 labels but to obtain as distant clusters as possible. All effect size metrics were higher when six clusters were obtained instead of five. GMM filtration improved the values of the 3<sup>rd</sup> quartile of  $\eta^2$  for both HDBSCAN and Louvain algorithms and pooled  $d$  for HDBSCAN. This can be especially noticed for the 3<sup>rd</sup> quartile of  $\eta^2$  in Fig. 2B, in which results of the Louvain approach with and without filtration are more separated. Hence, it is beneficial to compare the pooled  $d$  metrics with other criteria, including the Dice similarity index.

The methods with the highest (DiviK algorithm) and the lowest ( $H_{UMAP-C3}$ ) values of the pooled  $d$  metrics differ mainly regarding Luminal cases handling.  $H_{UMAP-C3}$  gave only two Luminal A subgroups and one bigger Luminal B subtype. DiviK, on the other hand, distinguished one more Luminal A subgroup that consists of patients clustered as Luminal A1 or B by the  $H_{UMAP-C3}$  approach. Moreover, the HER2-enriched subtype is more numerous for the DiviK algorithm, as it also contains a part of patients grouped as Luminal B with the  $H_{UMAP-C3}$  approach.

Division obtained with the DiviK algorithm greatly increased the number of proteins with an effect at least large (with decreased or increased levels in a subtype) for Luminal A1 and B subtypes. In the case of the Luminal A1 cluster, the number of proteins with at least a very large effect is also distinctly higher. Consequently, the number of associated KEGG pathways increased. Luminal A2 clusters do not vary much between the methods. However, the number of proteins and KEGG signaling pathways identified for the HER2-enriched subtype is smaller for the DiviK algorithm than for the  $H_{UMAP-C3}$  approach.

## 6 Conclusions

We performed breast cancer subtype identification with various combinations of machine learning methods for clustering and data dimensionality reduction. The outcomes were evaluated with several metrics, including the Dice coefficient and  $\eta^2$  effect size. We also proposed a custom effect size-based measure that represents the differences between each cluster and all remaining ones. The results of all metrics were consistent in terms of the best machine learning approach for breast cancer subpopulation detection. However, we believe it is beneficial to consider at least two different criteria for the comparison of various clustering algorithms and their parameters. Moreover, the metrics we used

can serve for the characterization of proteomic profiles of breast cancer groups and the identification of novel biomarkers.

The approach which outperformed all the others was the custom Divisive intelligent k-means (DiviK) algorithm with the feature filtration based on the decomposition of the Gaussian Mixture Model of the log<sub>2</sub>-scaled protein level variance. For the other clustering methods, the GMM-based filtration also improved all or some metrics, depending on the algorithm.

We detected subgroups of the Luminal A breast cancer subtype: three with best performing approaches and two with the worst ones. We also identified the proteins with significantly increased or decreased levels in particular subgroups and related them to the Kyoto Encyclopedia of Genes and Genomes (KEGG) pathways. The selection of the additional third Luminal A subgroup increased the number of proteins with elevated or decreased levels characteristic for Luminal clusters as well as the number of the associated KEGG pathways, especially for the Luminal B subtype.

**Acknowledgment.** This study is supported by European Social Fund grant no. POWR.03.02.00-00-I029 [JT] and Silesian University of Technology grant no. 02/070/BK\_22/0033 for Support and Development of Research Potential [JP]. The results published here are based upon data generated by the TCGA Research Network: <https://www.cancer.gov/tcga>.

## References

1. Sørli, T., et al.: Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proc. Natl. Acad. Sci.* **98**(19), 10869–10874 (2001)
2. Parker, J.S., et al.: Supervised risk predictor of breast cancer based on intrinsic subtypes. *J. Clin. Oncol.* **27**(8), 1160 (2009)
3. Berger, A.C., et al.: A comprehensive pan-cancer molecular study of gynecologic and breast cancers. *Cancer Cell* **33**(4), 690–705 (2018)
4. Koboldt, D.C.F.R., et al.: Comprehensive molecular portraits of human breast tumours. *Nature* **490**(7418), 61–70 (2012)
5. Leek, J.T., et al.: sva: Surrogate Variable Analysis. R package version 3.38.0. (2020)
6. Campello, R.J.G.B., Moulavi, D., Sander, J.: Density-based clustering based on hierarchical density estimates. In: Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G. (eds.) PAKDD 2013. LNCS (LNAI), vol. 7819, pp. 160–172. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-37456-2\\_14](https://doi.org/10.1007/978-3-642-37456-2_14)
7. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *J. Stat. Mech: Theory Exp.* **2008**(10), P10008 (2008)
8. Mrukwa, G., Polanska, J.: DiviK: divisive intelligent K-means for hands-free unsupervised clustering in biological big data. arXiv preprint [arXiv:2009.10706](https://arxiv.org/abs/2009.10706) (2020)
9. McInnes, L., Healy, J., Melville, J.: Umap: uniform manifold approximation and projection for dimension reduction. arXiv preprint [arXiv:1802.03426](https://arxiv.org/abs/1802.03426) (2018)
10. Marczyk, M., Jaksik, R., Polanski, A., Polanska, J.: Gamred—Adaptive filtering of high-throughput biological data. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **17**(1), 149–157 (2018)
11. Cohen, J.: *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Earlbaum Associates, New York (1988)
12. Sawilowsky, S.S.: New effect size rules of thumb. *J. Mod. Appl. Stat. Methods* **8**(2), 26 (2009)
13. Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y., Morishima, K.: KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res.* **45**(D1), D353–D361 (2017)





# Sperm-cell Detection Using YOLOv5 Architecture

Michal Dobrovolny<sup>1(✉)</sup>, Jakub Benes<sup>1</sup>, Ondrej Krejcar<sup>1</sup>, and Ali Selamat<sup>1,2,3</sup>

<sup>1</sup> Faculty of Informatics and Management, Center for Basic and Applied Research, University of Hradec Kralove, Hradec Kralove, Czech Republic  
{michal.dobrovolny,ondrej.krejcar}@uhk.cz

<sup>2</sup> Malaysia Japan International Institute of Technology (MJIIT),  
Universiti Teknologi Malaysia Kuala Lumpur, Jalan Sultan Yahya Petra,  
54100 Kuala Lumpur, Malaysia  
aselamat@utm.my

<sup>3</sup> Faculty of Engineering, School of Computing,  
Universiti Teknologi Malaysia (UTM), Skudai 81310, Malaysia

**Abstract.** Infertility has become a severe health issue in recent years. Sperm morphology, sperm motility, and sperm density are the most critical factors in male infertility. As a result, sperm motility, density, and morphology are examined in semen analysis carried out by laboratory professionals. However, applying a subjective analysis based on laboratory observation is easy to make a mistake. To reduce the effect of specialists in semen analysis, a computer-aided sperm count estimation approach is proposed in this work. The quantity of active sperm in the semen is determined using object detection methods focusing on sperm motility. The proposed strategy was tested using data from the Visem dataset provided by Association for Computing Machinery. We created a small sample custom dataset to prove that our network will be able to detect sperms in images. The best not-super tuned result is mAP 72.15.

**Keywords:** Sperm-cell detection · Small-object detection · Yolo · Computer-aided sperm analysis

## 1 Introduction

One out of every ten couples suffers from infertility [2]. It can have a detrimental impact on a couple's quality of life and lead to social and psychological issues [19]. Male factor is responsible for over half of all infertility cases [16]. Semen analysis is used to identify male infertility or subfertility and establish treatment options [6]. The shape and size of sperm components are inspected, and the percentages of normal and aberrant sperms are determined in sperm morphology assessment, one of several procedures in semen analysis.

Sperm analysis can be more profound and lead toward DNA analysis [22]. Due to the uncertain efficiency of normal sperm parameters in detecting male factor

infertility problems and boosting the success rates of assisted reproductive procedures, additional, comprehensive sperm parameters that could affect male fertility and reproduction have been investigated. Thus, using previously described methods such as single-cell gel electrophoresis (COMET) assay, sperm chromatin structure assay (SCSA), acridine orange test (AOT), terminal deoxynucleotidyl transferase-mediated deoxyuridine (TdT) triphosphate (dUTP) nick end labelling (TUNEL) assay, and sperm chromatin dispersion (SCD), the effects of various However, examining sperm DNA may be difficult due to the unique structure of sperm DNA, which differs from that of somatic cells [14]. Furthermore, during spermatogenesis, sperm DNA undergoes numerous alterations and is compressed by being tightly packaged with various types and numbers of protamines in different species. Despite these challenges, these approaches provide valuable information regarding the causes and consequences of DNA damage in sperm and the consequences of these damages on reproduction.

The vast majority of earlier sperm cell detectors achieved good accuracy since the density was minimal, according to a survey report [20]. (only 10–20 sperm cells presents in the video). The accuracy reduces dramatically as the density rises. For example, as described in [3,11], Hamilton Thorne, a commercial computer-based automated system, produces measuring inaccuracies in densely populated sperm suspended due to multiple clashing sperms. Our previous results show possible applications of object detection architectures [17], also as using deep convolutional networks to upscale medical images [4].

In order to address the lack of beef production, the Indonesian government built and mandated artificial insemination centres, such as The Lembang Institute for Artificial Insemination, to provide high quality frozen bull semen as the primary substance for artificial insemination. As a result, artificial insemination is the most extensively used reproductive technology for increasing beef production in the country [8]. Currently, sperm assessment is done manually at The Lembang Institute for Artificial Insemination.

The head, midpiece, and tail are the three primary sections of a spermatozoon, with the head being separated into acrosome and nucleus [18,21]. Anomalies can occur in any of these areas, although the abnormalities of the skull are the most common [26]. The initial stage in automatically detecting head anomalies is to segment the head from the background and into its basic pieces, notably the acrosome and midpiece. The contour information of the sperm head has been proven to be crucial for improving sperm head description, and classification [5]. As a result, precision is crucial while removing the sperm head contour.

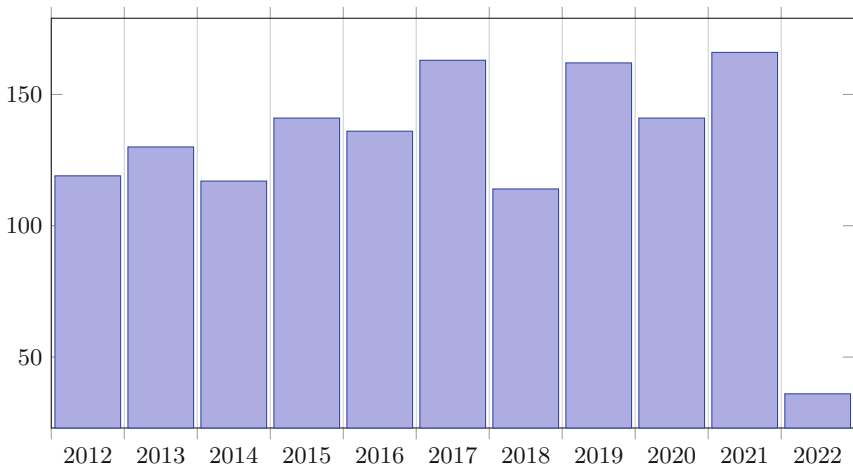
Subjectivity, low accuracy, inter variability, and intra-variability are all significant limitations of manual sperm motility measurement [1,10]. Computer-assisted sperm analysis (CASA) has been frequently adopted to circumvent these limitations. However, there are several limitations to employing CASA, especially when evaluating sperm motility in fresh bull sperm, where sperm motility is relatively fast, and partial occlusions are common. Our goal in this project is not to completely replace the current CASA system. We want to improve a few key components of the CASA system. The accuracy and speed of multi-sperm

tracking are also constrained. The second important issue is the difficulty in accurately classifying motility.

The low accuracy and speed of multi-sperm tracking are one of the major roadblocks. Several researchers have attempted to solve this problem [20]. Sorensen et al. [23], for example, utilized a Particle Filter and a Kalman Filter with a Hungarian algorithm for labeling, which is comparable to the method used by Jati et al. [13]. The authors of Imani et al. [12] used frame difference background subtraction and a non-linear diffusion filter to select the threshold value. The samples in these trials exhibited low sperm densities, with only a few sperm visible in one field of view, and blockage or passing sperm were uncommon.

Hidayatullaha et al. created a new method called deep sperm. This method reports better results than YOLOv3 and YOLOv4. Particularly on YOLOv3 by 2% of validation accuracy and on YOLOv4 0,25% [9]. This work does not include YOLOv5 architecture since there is no official paper describing exact parameters.

## 1.1 Topic Overview



**Fig. 1.** Yearly count of articles published on the Web of Science.

Regarding the Web of Science database, the topic of sperm detection is slightly increasing. We used a query *sperm(All Fields) AND detection(AllFields)*. Since 2012 there has been an increase of 39% of published articles; see the Fig. 1. Any criteria did not restrict the search. Their results also include conference papers. Indexes included in the search were: SCI-EXPANDED (2995), CPCI-S (272), ESCI (109), BKCI-S (36), SSCI (28), IC (2) and CPCI-SSH (1).

Keyword analysis in Fig. 2 shows the connection between sperm detection and other topics; in total, 100 keywords are connected into 6 clusters by 2974



the dataset, we opted only to include one video per participant. The video files in the dataset are over 35 terabytes, with each movie lasting between two and seven minutes.

The videos have a  $640 \times 480$  pixels resolution and a frame rate of 50 frames per second. The dataset includes six CSV files (five for data and one for video to participant ID mapping), a description file, and a video folder. Each video file is labelled with an ID, the date of video capture, and a brief optional explanation. The code of the person who assessed the video using the WHO standard is then included at the end of the filename. VISEM also includes five CSV files for each of the other data sets, a CSV file holding the IDs associated with each video and a text file containing definitions of some of the CSV columns.

Our use-case study needed more exact data about spermatozoa. Using the object detection method requires object position data not provided in the dataset. We decided to mark data on our own. We were using an annotation tool that will export boxes as coordinates used later for training.

We extracted images from videos in jpg format. The example shown in Fig. 3 We reduced the number of images to 382. The distribution of photographs was made similar for each subject. Our dataset is not publicly available yet and can be provided upon request on email address.

Annotation tool creates text annotation files with format.

$$\langle object - class \rangle \langle x\_center \rangle \langle y\_center \rangle \langle width \rangle \langle height \rangle \quad (1)$$

where  $\langle object - class \rangle$  is the object identity, integer number ranging from 0 to  $(classes - 1)$  and  $\langle x\_center \rangle \langle y\_center \rangle \langle width \rangle \langle height \rangle$  is the bounding box specification, float number relative to width and height of image ranging from 0.0 to 1.0.

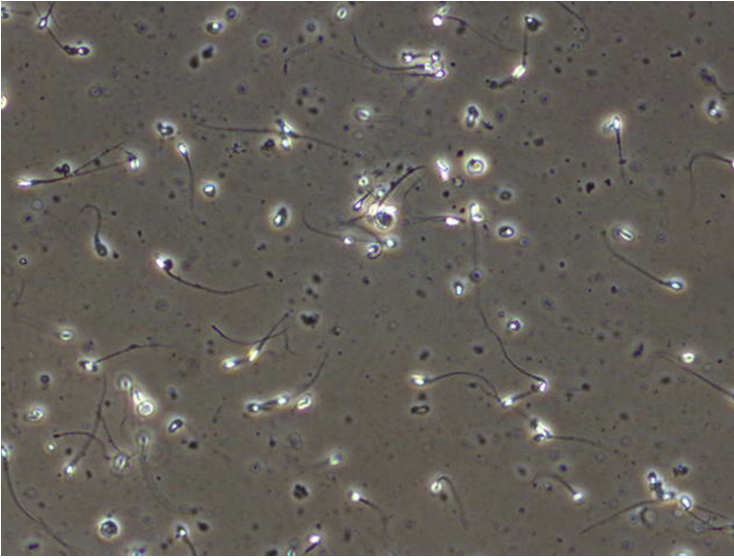
Since our focus is in this study to determine the possibility to use YOLOv5 architecture, we did not classify sperm cells with any defects. There can be a lot of biological defects. There are defects in heads, midpieces and tails. These defect cells we completely ignore. Sperm morphology detection can be built on top of our current results.

Images also include many artefacts that make this detection hard for deep networks. For example, there can be a blurry image, lousy lighting or wrong contrast.

We split marked images into two datasets. One is for training with a size of 368 images. Second, for validation, that contains 14 images to evaluate the training process.

The dataset we created for training had 3500 labels. Width and height graph is shown in Fig. 4.

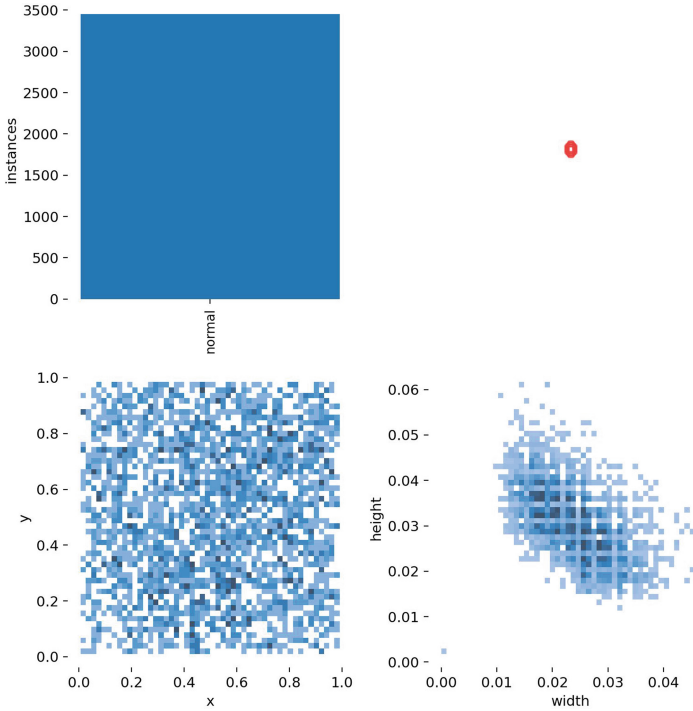
Regarding the current size of marked images, we will extend the size of both datasets. This approach will lead to better accuracy, lower overfitting, and better performance.



**Fig. 3.** Example image extracted from dataset video [7].

## 2.2 Neural Network Architecture

Yolov5 was chosen as our initial learner for three reasons. To begin, Yolov5 combined the cross-stage partial network (CSPNet) [24] into Darknet, resulting in the creation of CSPDarknet as the network’s backbone [27]. CSPNet solves the problem of recurrent gradient information in large-scale backbones by including gradient changes into the feature map, reducing model parameters and FLOPS (floating-point operations per second), ensuring inference speed and accuracy while simultaneously reducing model size. In detecting a sperm cell, speed and accuracy are critical, and the size of the model impacts its inference efficiency on resource-limited edge devices. Second, to improve information flow, the Yolov5 used a path aggregation network (PANet) [25] as its neck. PANet uses a new feature pyramid network (FPN) topology with an improved bottom-up approach to improving low-level feature propagation. Simultaneously, adaptive feature pooling, which connects the feature grid to all feature levels, ensures that meaningful information from each feature level reaches the next subnetwork. In addition, PANet improves precise localization signals in lower layers, significantly improving the object’s location accuracy. Finally, Yolov5’s head, the Yolo layer, generates three various sizes of feature maps to provide multi-scale prediction, allowing the model to handle tiny, medium, and large objects (Table 1).



**Fig. 4.** Explained labels and their sizes as width and height of boxes. Plotted with seaborn package.

**Table 1.** Described models in numbers of size.

Model	Nano	Small	Medium	Large	Xtra
Input size	640 × 480				
Number of layers	270	270	369	468	567
Number of parameters	1,765,270	7,022,326	20,871,318	46,138,294	86,217,814
Memory size	0.93 GB	1.73 GB	3.2 GB	4.97 GB	7.34 GB

During training all hyper parameters were set to same values. Learning rate: 0.01; momentum: 0.937; weight decay: 0.0005; batch size: 8. Pretrained weights were loaded from COCO [15] dataset training.

### 2.3 Hardware

In general, performance requirements for deep learning are very high. On our machine, we have two cards with 7 934 CUDA cores. This card is one of NVIDIA's best-performing cards. We chose NVIDIA cards solely because of the framework support. The graphics clock rate on one of our 1080TI cards



is 11 176 MB, with a clock rate of 1607 MHz. Another is a 2080TI, which has 11019 MB of graphics memory and a maximum clock rate of 1545. The processor used is an i7-8700 with a 3.20 GHz clock speed. Described in Table 2.

In version 3.8.2, we used Python as a programming language. Our Python programming environment was the cli-based script. PyTorch is our main machine learning framework.

Our environment is built on top of IntelliJ remote development and IntelliJ Idea. We ran the development backend on the server, and the coding was done directly on the machine with remote access.

## 2.4 Results

The best model achieved 72.15 mAP on the validation dataset, comparable to YOLOv4. Table 3 present the comparison of the results. All networks use an input size of image  $640 \times 480$  pixels images.

**Table 2.** Hardware specification of training machine.

Processor	Intel (R) Core™i7-8700 3.20 GHz (6xCORE)
RAM	16 GB $\times$ 4 (2666 MHz) CL13
GPU	GeForce GTX 1080TI (11176 MB) 1607 MHz
GPU	GeForce RTX 2080TI (11019 MB) 1545 MHz

**Table 3.** Results achieved on validation dataset.

Model	Nano	Small	Medium	Large	Xtra
Precision	64.7	61.6	71.7	88.6	64.6
Recall	61.4	64.9	57.8	52.6	71.9
mAP	69.6	64.6	66.4	72.1	68.6

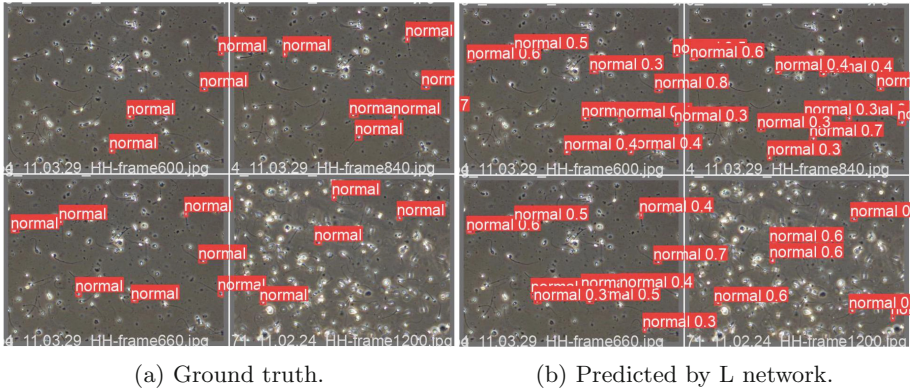
In a more detailed quantitative investigation (see Table 3). The best performing model is large. This network achieves a precision of 88.6%, recall of 52.6, and mAP is 72.1. Other networks are achieving lower results. The second best network is nano, with an mAP of 69.6 and precision of 64.7%.

If we compare models by precision only, we get them in order large, medium, nano, small, xtra. So we can determine that in our case, xtra is overfilling.

We can determine that the nano network is too resilient to learn these small objects such as sperm cells. Of course, this also applies to a small network.

**Artifact Handling.** Artefacts are also a significant source of inaccuracy when it comes to detection. For example, small markings were seen in one of the test samples. They possessed a grayscale similar to that of sperm cells, but they were smaller or defected (Fig. 5).





**Fig. 5.** Comparison for validation dataset labels of (a) Ground truth of labels, (b) predicted labels by network L. Generated during training - best epoch.

**Overfitting Handling.** The detectors may overfit if the training dataset’s samples contain too slight variation. We included samples in the dataset that we believe have enough variety to prevent overfitting in the model. The sperm cells seem relatively little when detected with a magnification of 100x. Therefore having annotated samples is generally limited. To investigate the impact on accuracy and decide which model has the best generalization ability, we add a single dataset split with low variation in the training data.

### 3 Conclusion

This study tested a deep neural network architecture, with its hyper-parameters and configurations detailed in the material and methods section. Detection of a sperm cell is the main target of the study. It was unaffected by partial occlusion, artefacts, many moving objects, the small size of the objects, low contrast, low video resolution, fuzzy objects, and a variety of lighting conditions.

To summarise, the proposed method performs well in precision, speed, and resource use. On the validation dataset, the mAP was 72.15. However, the tested method uses a significant amount of memory. Also, one of the networks ended up overfitted. Therefore, we will investigate the training dataset profoundly and try to propose a solution for this xtra network size.

The used dataset is an excellent opportunity to provide data for object detection in terms of sperm detection [7]. In future, we will also focus on using another dataset to make this work better compared with other methods.

Used results can provide great information for future applications; Applications as automatic determination of infertility.

**Acknowledgement.** The work and the contribution were supported by the SPEV project “Smart Solutions in Ubiquitous Computing Environments”, University of Hradec Kralove, Faculty of Informatics and Management, Czech Republic (under ID: UHK-FIM-SPEV-2022-2102).

**Conflict of Interest.** The authors declare that they have no conflicts of interest regarding the publication of this article.

## References

1. Auger, J., et al.: Intra- and inter-individual variability in human sperm concentration, motility and vitality assessment during a workshop involving ten laboratories. *15*(11), 2360–2368 (2000). <https://doi.org/10.1093/humrep/15.11.2360>
2. Boivin, J., Bunting, L., Collins, J.A., Nygren, K.G.: International estimates of infertility prevalence and treatment-seeking: potential need and demand for infertility medical care, *22*(6), 1506–1512 (2007). <https://doi.org/10.1093/humrep/dem046>
3. Broekhuijse, M.L.W.J., Šoštarić, E., Feitsma, H., Gadella, B.M.: Additional value of computer assisted semen analysis (CASA) compared to conventional motility assessments in pig artificial insemination, *76*(8), 1473–1486.e1 (2011). <https://www.sciencedirect.com/science/article/pii/S0093691X11003001>
4. Dobrovolny, M., Mls, K., Krejcar, O., Mambou, S., Selamat, A.: Medical image data upscaling with generative adversarial networks. In: Rojas, I., Valenzuela, O., Rojas, F., Herrera, L.J., Ortuño, F. (eds.) *IWBBIO 2020*. LNCS, vol. 12108, pp. 739–749. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-45385-5\\_66](https://doi.org/10.1007/978-3-030-45385-5_66)
5. García-Olalla, O., Alegre, E., Fernández-Robles, L., Malm, P., Bengtsson, E.: Acrosome integrity assessment of boar spermatozoa images using an early fusion of texture and contour descriptors, *120*(1), 49–64 (2015). <https://www.sciencedirect.com/science/article/pii/S0169260715000590>
6. Gumuscu, A., Tenekeci, M.E.: Estimation of active sperm count in spermogram using motion detection methods, *34*(3), 1274–1280 (2019). <https://www.webofscience.com/wos/woscc/full-record/WOS:000469481500012>. Faculty Engineering Architecture, Gazi University, Ankara
7. Haugen, T.B., et al.: VISEM: a multimodal video dataset of human spermatozoa. In: *Proceedings of the 10th ACM Multimedia Systems Conference*, pp. 261–266. ACM (2019). <https://dl.acm.org/doi/10.1145/3304109.3325814>
8. Hidayatullah, P., Mengko, T.L.E.R., Munir, R., Barlian, A.: Bull sperm tracking and machine learning-based motility classification, *9*, 61159–61170 (2021). *IEEE Access*
9. Hidayatullah, P., et al.: DeepSperm: a robust and real-time bull sperm-cell detection in densely populated semen videos, *209*, 106302 (2021). <https://www.sciencedirect.com/science/article/pii/S016926072100376X>
10. Hoogewijs, M.K., De Vlieghe, S.P., Govaere, J.L., De Schauwer, C., De Kruif, A., Van Soom, A.: Influence of counting chamber type on CASA outcomes of equine semen analysis, *44*(5), 542–549 (2012). <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.2042-3306.2011.00523.x>, eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.2042-3306.2011.00523.x>
11. Iguer-ouada, M., Verstegen, J.P.: Evaluation of the “hamilton thorn computer-based automated system” for dog semen analysis, *55*(3), 733–749 (2001). <https://www.sciencedirect.com/science/article/pii/S0093691X0100440X>



12. Imani, Y., Teyfour, N., Ahmadzadeh, M.R., Golabbakhsh, M.: A new method for multiple sperm cells tracking, **4**(1), 35 (2014). <https://www.jmssjournal.net/article.asp?issn=2228-7477;year=2014;volume=4;issue=1;page=35;epage=42;aulast=Imani;type=0>, Medknow Publications and Media Pvt. Ltd
13. Jati, G., Gunawan, A.A.S., Lestari, S.W., Jatmiko, W., Hilman, M.H.: Multi-sperm tracking using Hungarian Kalman filter on low frame rate video. In: 2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS), pp. 530–535 (2016)
14. Kucuk, N.: Sperm DNA and detection of DNA fragmentations in sperm, **44**(1), 1–5 (2018). <https://www.webofscience.com/wos/woscc/full-record/WOS:000422986400002>, Aves, Sisli
15. Lin, T.Y., et al.: Microsoft COCO: common objects in context <http://arxiv.org/abs/1405.0312> (2014)
16. Maduro, M.R., Lamb, D.J.: Understanding new genetics of male infertility, **168**(5), 2197–2205 (2002). <https://www.sciencedirect.com/science/article/pii/S0022534705643558>
17. Mambou, S., Krejcar, O., Selamat, A., Dobrovolny, M., Maresova, P., Kuca, K.: Novel thermal image classification based on techniques derived from mathematical morphology: case of breast cancer. In: Rojas, I., Valenzuela, O., Rojas, F., Herrera, L.J., Ortuño, F. (eds.) IWBBIO 2020. LNCS, vol. 12108, pp. 683–694. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-45385-5\\_61](https://doi.org/10.1007/978-3-030-45385-5_61)
18. Martin, R.: Detection of genetic-damage in human sperm, **7**, 47–52 (1993). <https://www.webofscience.com/wos/woscc/full-record/WOS:A1993LU30500006>, Pergamon-Elsevier Science Ltd., Oxford
19. Schmidt, L.: Psychosocial consequences of infertility and treatment. In: Carrell, D.T., Peterson, C.M. (eds.) Reproductive Endocrinology and Infertility: Integrating Modern Clinical and Laboratory Practice, pp. 93–100. Springer, New York (2010). [https://doi.org/10.1007/978-1-4419-1436-1\\_7](https://doi.org/10.1007/978-1-4419-1436-1_7)
20. Hidayatullah, P., Mengko, T.L.E.R., Munir, R.: A survey on multisperm tracking for sperm motility measurement, **7**(5), 144–151 (2017). School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Jl. Ganesha no. 10 Bandung, Jawa Barat, Indonesia. <http://www.ijmlc.org/vol7/637-C46.pdf>
21. Silva, P.F.N., Gadella, B.M.: Detection of damage in mammalian sperm cells, **65**(5), 958–978 (2006). <https://www.webofscience.com/wos/woscc/full-record/WOS:000236041300006>, Elsevier Science Inc., New York
22. Suttipasit, P.: Forensic spermatozoa detection, **40**(4), 304–311 (2019). <https://www.webofscience.com/wos/woscc/full-record/WOS:000499474800001>, Lippincott Williams & Wilkins, Philadelphia
23. Sørensen, L., Østergaard, J., Johansen, P., Bruijne, M.d.: Multi-object tracking of human spermatozoa. In: Medical Imaging 2008: Image Processing, vol. 6914, pp. 784–795 (2008). SPIE, <https://www.spiedigitallibrary.org/conference-proceedings-of-spie/6914/69142C/Multi-object-tracking-of-human-spermatozoa/10.1117/12.771135.full>
24. Wang, C.Y., Liao, H.Y.M., Yeh, I.H., Wu, Y.H., Chen, P.Y., Hsieh, J.W.: CSPNet: a new backbone that can enhance learning capability of CNN (2020). <http://arxiv.org/abs/1911.11929>
25. Wang, K., Liew, J.H., Zou, Y., Zhou, D., Feng, J.: PANet: few-shot image semantic segmentation with prototype alignment (2019). <http://arxiv.org/abs/1908.06391>

26. World Health Organization: Regional Office for the Eastern Mediterranean: List of basic sources in English for a medical faculty library (2013). <https://apps.who.int/iris/handle/10665/119927>, section: vi, 133 p.; 30 cm
27. Xu, R., Lin, H., Lu, K., Cao, L., Liu, Y.: A forest fire detection system based on ensemble learning, **12**, 217 (2021)

# **Machine Learning in Bioinformatics**



# Comparative Analysis of Supervised Cell Type Detection in Single-Cell RNA-seq Data

Akram Vasighizaker , Sheena Hora, Yash Trivedi, and Luis Rueda 

School of Computer Science, University of Windsor, Windsor, ON, Canada  
{vasighi, horas, trived22, lrueda}@uwindsor.ca

**Abstract.** Recent studies on Single-cell RNA sequencing (scRNA-seq) technology have been widely applied in biological research and drug discovery. Before in-depth investigations of the functionality of single cells for pathological goals, identification of cell types is an essential step. Recently, several unsupervised learning methods have been developed to identify cell types. However, annotating clusters with the correct cell types require considerable efforts using marker genes. Due to the lack of enough annotated datasets, supervised techniques have not been commonly used in scRNA-seq studies. On the other hand, classification methods use feature selection algorithms to improve the prediction accuracy by finding the most informative features among many in high-dimensional datasets. Hence, to automating the process of annotation of clusters of cell types, we can take advantage of classification models. This article evaluated the performance of three state-of-the-art supervised classification methods, namely support vector machine,  $k$ -nearest neighbor, and random forest combined with three feature selection methods, namely Chi-squared, information gain, and ANOVA F-value. The results of applying nine combinations of these methods on three standard scRNA-seq datasets show that support vector machine combined with information gain outperforms other combinations of techniques. Moreover, we investigated reference gene sets and found 11 out of 20 highly variable genes in two different Pancreas gene sets to validate our findings. This article sheds some light on the potential use of identifying marker genes to improve the automatic identification of cell types.

**Keywords:** Cell type identification · scRNA-seq data analysis · Marker gene identification · Feature selection · Classification

## 1 Introduction

Tumor heterogeneity is a common phenomenon in studying different types of cancer. In this regard, novel techniques such as single-cell RNA sequencing (scRNA sequencing) can be used to detect unknown tumors and consequently drug discovery, better treatment, diagnosis, and prognosis. Thus, one of the first fundamental steps to perform an in-depth analysis of single-cell sequencing data

consists of identifying cell types. Hidden diversity and characteristics of a particular cell type can be found via differentially expressed genes or marker genes.

Supervised or unsupervised learning approaches can effectively be used to identify various cell types depending on the dataset, annotated or unannotated, respectively. Typically, in single-cell RNA-seq downstream analysis, clustering techniques are used to reveal well-separated clusters of cells and annotate them manually with different cell types using canonical markers and reference databases. Different clustering methods try multiple parameters to achieve higher performance. Setting up the clustering parameters, such as the number of clusters, is a challenging point [14]. For example, several clustering methods are compared in [4]. Among them, SC3 [7], CIDR [8], Ascend [12], SAFE-clustering [16], and TSCAN [6] all possess built-in methods for estimating the optimal number of clusters. However, Ascend and CIDR underestimated the number of clusters, whereas SC3 and TSCAN tend to overestimate. Moreover, manually annotating the obtained clusters using differential expression analysis is time-consuming and non-reproducible in clustering methods.

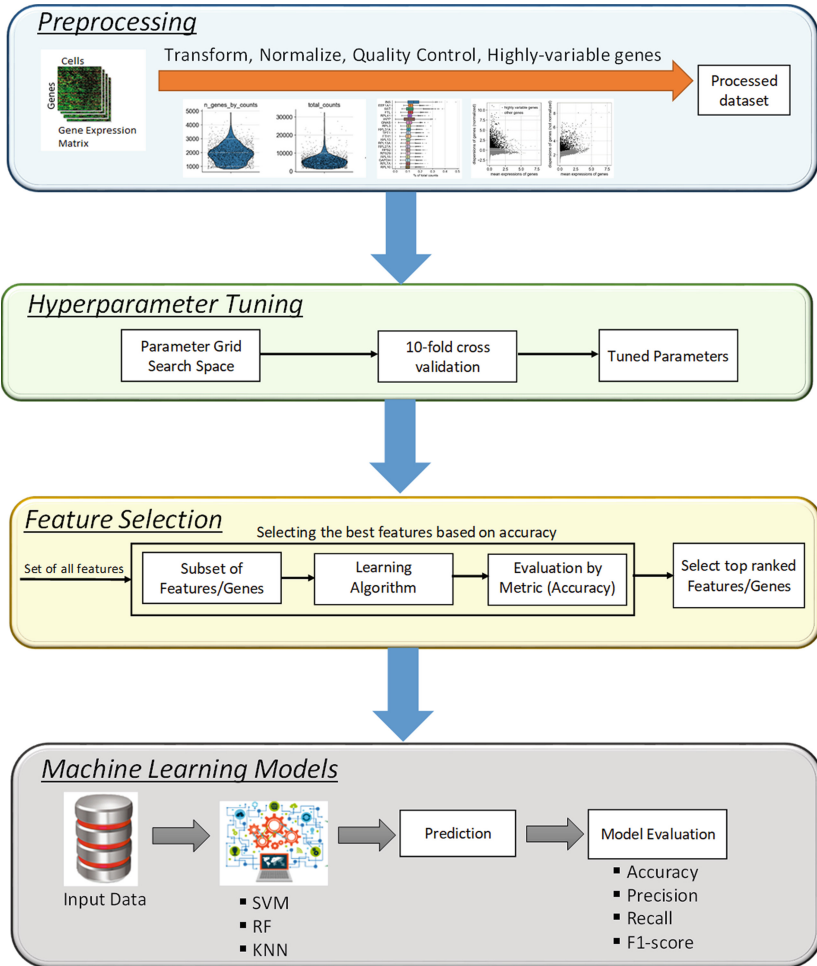
On the other hand, classification techniques have increasingly developed to identify cell types automatically instead of manually annotating clusters of cells. In addition to this, different feature selection techniques can be used to avoid the “curse of dimensionality” and select a reduced number of the significant marker genes. A comparative study in [1] discussed 22 supervised techniques, including random forest classifier (RF),  $k$ -nearest neighbor ( $k$ -NN), support vector machine (SVM). One of the challenges covered in this study is feature selection. Three different cell-specific purpose feature selection techniques have been used, including random gene selection, highly variable genes (HVG) selection, and selecting genes based on the number of dropouts (zero expression). They benchmarked their experiments based on the number of features. The findings show that the performance of the classifiers highly depends on the number of cells and genes, selected marker genes, and dataset complexity. In this study, we used general-purpose techniques, instead of cell-specific ones, to compare three state-of-the-art feature selection techniques combined with three popular classifiers to complement the feature selection step. We also biologically validate cell type marker genes identified by the best feature selection method.

## 2 Materials and Methods

### 2.1 Framework

Three general-purpose feature selection methods, namely ANOVA F-value, Chi-squared, and information gain (IG), along with three state-of-the-art classification methods, including SVM,  $k$ -NN, and RF, are used in our experiments to identify cell types automatically. A comparative study on scRNA-seq data is done in this work. To this end, we followed the pipeline depicted in Fig. 1. First, we performed pre-processing steps, including filtering, normalization, and scaling. Then, to find the best parameters for the classification methods, hyperparameter tuning and optimization were done on pre-processed data. The most

informative features were extracted in the feature selection step. Three classifiers combined with three feature selection algorithms were evaluated to find the best model. Finally, cell types are predicted by the method with higher accuracy.



**Fig. 1.** Pipeline overview of the experiments.

It is worth mentioning that although there are other state-of-the-art classification methods including deep learning ones, feeding this group of methods requires rich labeled datasets which is the main limitation of scRNA-seq datasets. We used the Scikit-learn in Python version 3.7 to perform the feature selection and classification methods [10].



## 2.2 Dataset

Public, annotated scRNA-seq data sets with the accession number of GSM2230757 and GSM2230758 under series GSE84133 [3], and PBMC 10X V2 were extracted from NCBI’s Gene Expression Omnibus [5] and used in this article to evaluate the classification performance. These datasets include transcripts of pancreatic and peripheral blood cells from human donors. Pancreatic cells are divided into eight groups of previously characterized cell types: alpha, beta, acinar, delta, quiescent, activated pancreatic stellate cells, endothelial, and ductal cells. The existence of these cell types is validated with immuno-histochemistry stains [3] so that it can be a good resource for the discovery of cell types. Also, the PBMC dataset includes nine different cell types. The details of datasets are listed in Table 1.

**Table 1.** Details of the datasets studied in this work.

Dataset	Tissue	Accession #	Cell Types #	Cells #	Genes #
Baron-human1 (Data1)	Human-Pancreas	GSM2230757	8	1,937	20,125
Baron-human2 (Data2)	Human-Pancreas	GSM2230758	8	1,724	20,125
PBMC (Data3)	Preperhal Blood	10X V2	9	23,154	22,280

## 2.3 Data Pre-processing

Raw read count matrices generated using next-generation sequencing technologies contain low-quality sequencing information based on the expression levels. Pre-processing step (Fig. 1) is to ensure removing any weakly expressed genes or low-quality cells, including damaged, dead, or degraded during sequencing, and are represented by a low number of expressed genes in the read count matrices. To perform pre-processing, we followed the standard pre-processing pipeline in scRNA-seq data analysis [9]. According to this pipeline, cells with less than 200 expressed genes, and genes expressed in less than three cells are filtered out. In Data1, for example, we first filtered out 5,387 low-expression genes that were detected in less than three cells and kept 14,739 genes. Further analysis of the data distribution showed low-quality cells and led to removing seven cells. After per-gene quantification, we selected a subset of highly variable genes to use in downstream analyses. To this end, we chose a common strategy routinely used [2] and defined the set of highly variable genes given a normalized dispersion higher than 0.5 after normalization and obtained 2,546 genes at the end. We used Scanpy [15], a specifically designed package to work with scRNA-seq datasets, for pre-processing steps.

## 2.4 Feature Selection

In scRNA-seq data analysis, feature selection or gene selection can be an essential component due to the curse of dimensionality. The primary motivation behind

feature selection or gene selection in cell type identification is that cell types are often distinguished by only a few essential genes known as biomarkers. This study investigated three general-purpose feature selection approaches, including Analysis of Variance (ANOVA) F-value, Chi-squared, and information gain (IG) to select a sorted list of genes. The best number of genes for the training model is chosen by calculating the model's performance for top  $k$  genes where  $k = 100, 200, 300, \text{ and } 400$ . We evaluated the accuracy of the methods by varying the number of marker genes based on different computational approaches.

**Analysis of Variance (Anova) F-value.** ANOVA F-value assumes that there is a linear relationship between variables and target, and also the variables are normally distributed. It uses F-tests to statistically measure the ratio of two variances, i.e. how far the data points are dispersed from the mean. The results show the statistical significance of the test. F-value is a very important part of ANOVA and is calculated by the Eq. 1.

$$F = \frac{\sigma_1^2}{\sigma_2^2} \quad (1)$$

where  $F$  is the F-value,  $\sigma_1$  is the larger sample variance and  $\sigma_2$  is the smaller sample variance.

**Chi-squared.** Pearson's Chi-squared test or just Chisquared test is a statistical test applied to the categorical features to test the relationships among them. It is suited for non-negative variables and mostly boolean, frequencies, or counts. It uses frequency distribution of the features to determine the correlation or association among them. The test calculates chi-squared statistics i.e. the expected frequencies of the observations and then determines whether the observed frequencies match the expected frequencies. The Eq. 2 shows how this method calculates the correlation among features.

$$\chi^2 = \sum \frac{(\text{ObservedFrequency} - \text{ExpectedFrequency})^2}{\text{Expected}} \quad (2)$$

where  $\chi^2$  is Chi-squared.

**Information Gain.** Information Gain is defined in terms of uncertainty. The lesser the information gain, the higher the uncertainty. If  $IG(X) > IG(Y)$ , it means feature  $X$  will be better and preferred where  $IG(X)$  represents the information gain from feature  $X$ . The relevance of feature is estimated by considering the information gain for each feature and choosing the one with maximum value. It is defined as the difference between prior uncertainty and uncertainty after considering feature  $X$  as shown in Eq. 3 [11].

$$IG(X) = \sum_i U(P(C_i)) - E\left[\sum_i U(P(C_i|X))\right] \quad (3)$$

where  $U$  represents uncertainty function,  $P(C_i)$  represents probability of class  $C_i$  before considering feature  $X$  and  $P(C_i|X)$  represents posterior probability of class  $C_i$  after considering feature  $X$ .

## 2.5 Evaluation Metrics

We applied the most commonly used evaluation metrics, namely accuracy, precision, recall, and F-score to systematically estimate and compare the performance of different methods. To this end, we used 10-fold cross-validation to test and train the model.

## 3 Results and Discussion

### 3.1 Parameter Optimization

To select the best parameters of the classifiers ( $K$ -NN, RF, and SVM), we used a Bayesian model-based optimization approach with Gaussian as an adaptive hyperparameter search. It is a fast approach compared to grid search and random search. We employed Bayesian search to tune hyperparameters, which rather than scanning the hyperparameter space mindlessly (as in the grid or random search), this strategy emphasizes the use of knowledge obtained in one step to discover the next set of hyperparameters that would improve model performance. This method, in an iterative manner, continues until the optimal result is obtained. Since it prioritizes hyperparameters that appear more promising from previous steps, the Bayesian technique is able to find the best hyperparameters in less time (fewer iterations) than grid search and random search.

**Table 2.** The best parameters for each method obtained using Bayesian Optimization for the datasets.

Method	Best parameters found			
<b>Data1</b>				
K-NN	$k = 5$			
RF	$n\_estimators = 359$	$max\_depth = 41$	$criteria = 'gini'$	$max\_features = 'sqrt'$
SVM	$C = 0.5$	$gamma = 0.2$	$kernel = 'linear'$	
<b>Data2</b>				
K-NN	$k = 4$			
RF	$n\_estimators = 100$	$max\_depth = 1$	$criteria = 'gini'$	$max\_features = 'sqrt'$
SVM	$C = 0.5$	$gamma = 0.2$	$kernel = 'linear'$	
<b>Data3</b>				
K-NN	$k = 6$			
RF	$n\_estimators = 495$	$max\_depth = 54$	$criteria = 'gini'$	$max\_features = 'sqrt'$
SVM	$C = 0.1$	$gamma = 0.2$	$kernel = 'poly'$	$degree = 2$

The best parameter based on the optimization results for each classification method for Data1, Data2 and Data3 are presented in Table 2.

For selecting the best value of  $k$  for the  $k$ -NN classifier, the following values of the  $k = (4,5,6)$  in the search space are inspected. The quality of the result is determined by  $k$  with the highest average accuracy of the three feature selection methods.

For RF, the following values for the search space are investigated:  $n\_estimators = (100, 500)$ ,  $max\_features = (sqrt, log2)$ ,  $max\_depth = (1, 60)$  and  $criterion = (gini, entropy)$ . The  $n\_estimators$  parameter are the number of trees to be considered. The parameter  $max\_features$  are the maximum number of features to be considered for individual tree.  $max\_depth$  parameter is the maximum depth of the tree where maximum depth is defined as the longest path from root node to the leaf node and the parameter  $criterion$  is the function which is used to evaluate the quality of split.

RF, by default, uses built-in feature selection methods, including ‘Ginni’ and ‘entropy’. To ensure that each method uses its approach for classification, we allowed RF to use this ability during the training process with a list of selected features using the feature selection methods.

For SVM, the following values for the search space are inspected:  $C = (0.1, 0.5, 1)$ ,  $gamma = (0.1, 0.2, 0.3)$ ,  $degree = (1, 8)$  and  $kernel = (rbf, poly, linear)$ . The regularization parameter, aka the cost of misclassification,  $C$ , is a degree of importance that is given to the misclassifications error. SVM seeks a trade-off to maximize the margin among the classes and minimize the number of misclassifications. The larger the value of  $C$ , the larger is the miss-classification cost. Kernels are functions used to solve non-linear problems by making a curvative hyperplane to separate classes. The parameter  $gamma$  decides the curvature in the decision boundary in non-linear kernels, where a large value of  $gamma$  means more curvature, i.e., softer and tends to overfit the data.

### 3.2 Classification Results

To investigate the effect of the selected features (genes) as a form of prior knowledge, we evaluated the performance of the classifiers based on the different number of selected features using three different approaches. We examined  $k$  features where  $k = 100, 200, 300$ , and 400 to determine the best number of features to optimize the performance of the classifier. The best value of  $k$  with the highest accuracy of a combination of each feature selection and classification method for Data1 is shown in Table 3.

For Data1, the  $k$ -NN classification method results reveal a high accuracy of 96.11% with 400 features when using the Information Gain feature selection method. The RF classification method for Data1 indicates high accuracy with 400 features for all three feature selection methods. A combination of this classifier with IG gives the best accuracy of 97.05%. Observing the results of the SVM classification method for Data1, all three feature selection methods reveal high accuracy with 400 features. Again, SVM combined with IG gives the highest

accuracy of 98.08%. Among all the combinations, SVM combined with IG shows highest performance with 98.08% accuracy for Data1.

For Data2 among all the combination,  $k$ -NN classification method achieves high accuracy (94.66%) with 200 features selected from IG feature selection method, RF and IG combination achieves high accuracy (96.06%) with 400 features, and lastly, SVM achieves high accuracy with 400 features (98.09%) selected from IG feature selection method. For Data2, SVM coupled with IG provides the best performance, with 98.09% accuracy.

For Data3, SVM achieves highest performance (84.91% accuracy) with 200 features selected from Anova feature selection method. In general, SVM outperformed the other two classification methods for all three datasets.

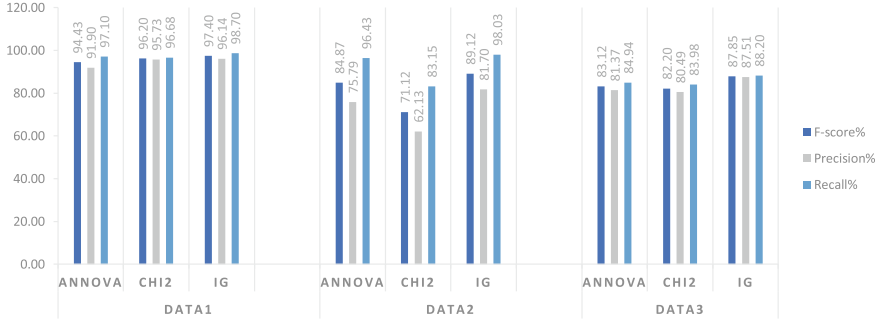
To generalize our experiments, we used two datasets with the same number of genes and the different number of cells (i.e., Data1 and Data2), and another dataset with the higher number of cells and genes (i.e., Data3) comparatively. Other metrics are presented in Figs. 2, 3, and 4.

Among all combinations of classification and feature selection methods, SVM combined with IG significantly outperformed other approaches. High accuracy of 98.08% for Data1 means that the features that have been selected are highly correlated and significantly help fulfill our primary objective.

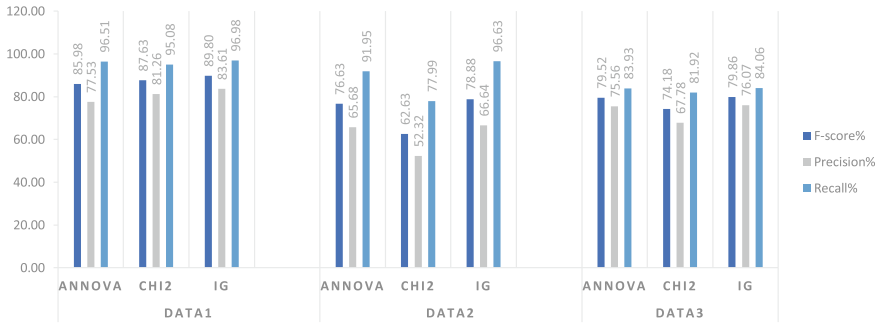
**Table 3.** Classification accuracy obtained by three classification methods combined with feature selection methods through selected features for Data1.

Method	No. of Features	Accuracy %
<b><math>k</math>-NN</b>		
ANOVA F-value	400	95.65
Chi-squared	400	93.99
Information Gain	400	96.11
<b>Random Forest (RF)</b>		
ANOVA F-value	400	96.74
Chi-squared	400	96.84
Information Gain	400	97.05
<b>SVM</b>		
ANOVA F-value	400	97.72
Chi-squared	400	96.79
Information Gain	400	98.08

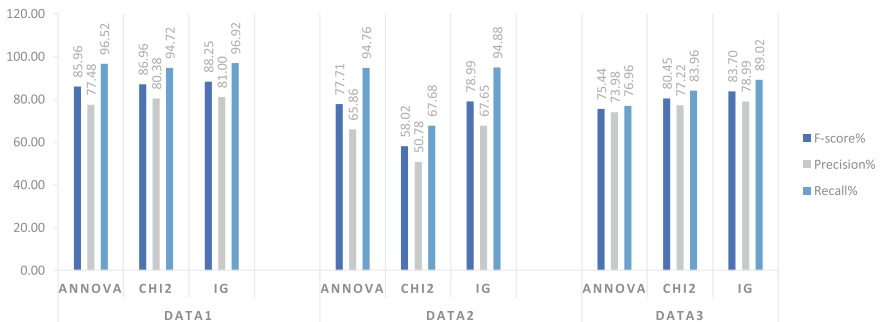
Our results highlight the power of the SVM classifier combined with the IG as the best approach. Also, it shows that the performance of classifiers highly depends on the selected marker genes using different techniques.



**Fig. 2.** Average performance of the SVM classifier combined with three feature selection methods.



**Fig. 3.** Average performance of the *k*-NN classifier combined with three feature selection methods.



**Fig. 4.** Average performance of the RF classifier combined with three feature selection methods.

### 3.3 Biological Validation

We evaluated the performance of our method for detecting cell types using the high-ranked features or differentially expressed genes through investigating the current literature and reference databases. By investigating GSEA [13] on the result of Data1, we found 9 out of 20 overlapped genes between Pancreas gene sets, “Muraro Pancreas Endothelial Cell”, and top genes found by our method. The list of 9 overlapped genes, along with the description of their functionality, is depicted in Table 4. Moreover, we conducted a biological validation on the other datasets, Baron Human2 (Data2) and PBMC (Data3). The results are depicted on Tables 5 and 6. Overall, our results show the power of our method to identify the cell types using a list of marker genes in scRNA-seq datasets.

**Table 4.** Muraro Pancreas Endothelial Cell gene set.

Gene symbol	Description
IFITM3	interferon induced transmembrane protein 3 [Source:HGNC Symbol;Acc:HGNC:5414]
IGFBP4	insulin like growth factor binding protein 4 [Source:HGNC Symbol;Acc:HGNC:5473]
IFITM2	interferon induced transmembrane protein 2 [Source:HGNC Symbol;Acc:HGNC:5413]
COL4A1	collagen type IV alpha 1 chain [Source:HGNC Symbol;Acc:HGNC:2202]
SPARC	secreted protein acidic and cysteine rich [Source:HGNC Symbol;Acc:HGNC:11219]
IGFBP7	insulin like growth factor binding protein 7 [Source:HGNC Symbol;Acc:HGNC:5476]
VIM	vimentin [Source:HGNC Symbol;Acc:HGNC:12692]
TM4SF1	transmembrane 4L six family member 1 [Source:HGNC Symbol;Acc:HGNC:11853]
HLA-B	“major histocompatibility complex, class I, B [Source:HGNC Symbol;Acc:HGNC:4932]”

**Table 5.** Muraro Pancreas Ductal Cell gene set.

Gene symbol	Description
CDC42EP1	CDC42 effector protein 1 [Source:HGNC Symbol;Acc:HGNC:17014]
PMEPA1	“prostate transmembrane protein, androgen induced 1 [Source:HGNC Symbol;Acc:HGNC:14107]”
TACSTD2	tumor associated calcium signal transducer 2 [Source:HGNC Symbol;Acc:HGNC:11530]
KRT7	keratin 7 [Source:HGNC Symbol;Acc:HGNC:6445]
SDC4	syndecan 4 [Source:HGNC Symbol;Acc:HGNC:10661]
KRT19	keratin 19 [Source:HGNC Symbol;Acc:HGNC:6436]
FLNA	filamin A [Source:HGNC Symbol;Acc:HGNC:3754]
IFITM3	interferon induced transmembrane protein 3 [Source:HGNC Symbol;Acc:HGNC:5414]
SERPING1	serpin family G member 1 [Source:HGNC Symbol;Acc:HGNC:1228]
COL18A1	collagen type XVIII alpha 1 chain [Source:HGNC Symbol;Acc:HGNC:2195]

**Table 6.** Travaglini Lung Ereg Dendritic Cell gene set.

Gene symbol	Description
HLA-DPB1	“major histocompatibility complex, class II, DP beta 1 [Source:HGNC Symbol;Acc:HGNC:4940]”
TYROBP	transmembrane immune signaling adaptor TYROBP [Source:HGNC Symbol;Acc:HGNC:12449]
HLA-DPA1	“major histocompatibility complex, class II, DP alpha 1 [Source:HGNC Symbol;Acc:HGNC:4938]”
AIF1	allograft inflammatory factor 1 [Source:HGNC Symbol;Acc:HGNC:352]
LST1	leukocyte specific transcript 1 [Source:HGNC Symbol;Acc:HGNC:14189]
FCER1G	Fc fragment of IgE receptor Ig [Source:HGNC Symbol;Acc:HGNC:3611]
HLA-DQB1	“major histocompatibility complex, class II, DQ beta 1 [Source:HGNC Symbol;Acc:HGNC:4944]”
CST3	cystatin C [Source:HGNC Symbol;Acc:HGNC:2475]
FCN1	ficolin 1 [Source:HGNC Symbol;Acc:HGNC:3623]
VCAN	versican [Source:HGNC Symbol;Acc:HGNC:2464]
HLA-DRB1	“major histocompatibility complex, class II, DR beta 1 [Source:HGNC Symbol;Acc:HGNC:4948]”
GPX1	glutathione peroxidase 1 [Source:HGNC Symbol;Acc:HGNC:4553]
GZMB	granzyme B [Source:HGNC Symbol;Acc:HGNC:4709]



## 4 Conclusion and Future Work

This work focuses on the supervised identification of cell types using feature selection methods combined with classification techniques on an annotated dataset. Investigating similarities among features using three state-of-the-art feature selection methods to reduce the dimension of the feature space helps enhance the classification task and overcome its inherent computational complexity. Finding similarities can result from linear or non-linear relationships among the features, data distribution, or data entropy. Biologically speaking, the similarity is defined by structural, functional, or evolutionary relationships among the genes that lead to finding the most accurate class for a new test sample. In our experiments, we have demonstrated that genes in our dataset that have similar expression patterns were grouped in highly-scored classes. Identifying biomarker genes that are differentially expressed among different cell types is done in the feature selection step. This work highlights the power of using only a sub-group of highly effective genes to find cell types. Thus, we can take advantage of disregarding a considerable number of uninformative genes for identifying the corresponding cell types. Moreover, there are some potential future avenues to find cell types automatically using scRNA-seq data. For example, conducting a comprehensive experiment using a more significant number of samples obtained from different tissues shows potential in enhancing the results on a larger scale.

## References

1. Abdelaal, T., et al.: A comparison of automatic cell identification methods for single-cell rna sequencing data. *Genome Biol.* **20**(1), 1–19 (2019)
2. Amezquita, R.A., et al.: Orchestrating single-cell analysis with bioconductor. *Nature Methods* **17**(2), 137–145 (2020)
3. Baron, M., et al.: A single-cell transcriptomic map of the human and mouse pancreas reveals inter- and intra-cell population structure. *Cell Syst.* **3**(4), 346–360 (2016)
4. Duò, A., Robinson, M.D., Soneson, C.: A systematic performance evaluation of clustering methods for single-cell rna-seq data. *F1000Research*, 7 (2018)
5. Edgar, R., Domrachev, M., Lash, A.E.: Gene expression omnibus: Ncbi gene expression and hybridization array data repository. *Nucleic Acids Res.* **30**(1), 207–210 (2002)
6. Ji, Z., Ji, H.: Tscan: Pseudo-time reconstruction and evaluation in single-cell rna-seq analysis. *Nucleic Acids Res.* **44**(13), e117–e117 (2016)
7. Kiselev, V.Y., et al.: SC3: consensus clustering of single-cell RNA-SEQ data. *Nature Methods* **14**(5), 483–486 (2017)
8. Lin, P., Troup, M., Ho, J.W.: CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data. *Genome Biol.* **18**(1), 1–11 (2017)
9. Luecken, M.D., Theis, F.J.: Current best practices in single-cell RNA-SEQ analysis: a tutorial. *Molecular Syst. Biol.* **15**(6), e8746 (2019)
10. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)

11. Raza, M.S., Qamar, U.: Understanding and Using Rough Set Based Feature Selection: Concepts, Techniques and Applications. Springer, Singapore (2019). <https://doi.org/10.1007/978-981-32-9166-9>
12. Senabouth, A., et al.: ascend: R package for analysis of single-cell RNA-seq data. *GigaScience* **8**(8), giz087 (2019)
13. Subramanian, A., Kuehn, H., Gould, J., Tamayo, P., Mesirov, J.P.: GSEA-P: a desktop application for gene set enrichment analysis. *Bioinformatics* **23**(23), 3251–3253 (2007)
14. Vasighizaker, A., Danda, S., Rueda, L.: Discovering cell types using manifold learning and enhanced visualization of single-cell RNA-seq data. *Sci. Rep.* **12**(1), 1–16 (2022). <https://doi.org/10.1038/s41598-021-03613-0>
15. Wolf, F.A., Wolf, F.A., Angerer, P., Theis, F.J.: SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol.* **19**(1), 1–5 (2018)
16. Yang, Y., Huh, R., Culpepper, H.W., Lin, Y., Love, M.I., Li, Y.: Safe-clustering: single-cell aggregated (from ensemble) clustering for single-cell RNA-seq data. *Bioinformatics* **35**(8), 1269–1277 (2019)



# PathWeigh – Quantifying the Behavior of Biochemical Pathway Cascades

Dani Livne and Sol Efroni<sup>(✉)</sup>

The Mina & Everard Goodman Faculty of Life Sciences, Bar-Ilan University, Ramat-Gan, Israel  
sol.efroni@biu.ac.il

**Abstract.** Biochemical pathways analysis is an effective tool for understanding changes in gene expression data and associating such changes with cellular phenotypes. Pathway research aims to identify associated proteins within a cell using pathways and at building new pathways from a group of molecules of interest. Using pathway-based methods we gain insight into different functions of relevant molecules and find direct and indirect relations between them. We present PathWeigh, a Python-based tool for pathway analysis and graph presentation. The tool is open-sourced, extendable and runtime efficient.

PathWeigh is available at <https://github.com/zurkin1/Pathweigh> and is released under MIT license. A sample Python notebook is provided with examples of running the tool.

**Keywords:** Pathway analysis · RNA sequencing · Machine learning

## 1 Introduction

A typical biological signaling pathway is defined as a network of molecular or chemical interactions. Each interaction contains one or more input genes, proteins or other complex molecules, promoters and inhibitors, and one or more output molecules.

Pathway analysis exposes networks of regulation between elements that lead to cell phenotypes. Pathway network research is often able to provide robust solutions, in cases where differential expression at the gene level cannot provide such insights. It is a key tool in a large area of research called ‘system biology’ where researchers search for macro properties of a given biological network rather than local micro behavior [8]. System biology tools include dynamical modeling using differential equations or statistical modeling using sample data such as RNA-sequencing reads [6, 9].

One of the earliest works to make use of network pathways is [11]. The authors showed how gene expression profiles define biological modules that are either activated or deactivated in various tumor conditions. TAPPA [4] used molecular connectivity concepts to calculate scores that are based on topological structures. Later CLIPPER by

---

**Supplementary Information** The online version contains supplementary material available at [https://doi.org/10.1007/978-3-031-07802-6\\_29](https://doi.org/10.1007/978-3-031-07802-6_29).

© Springer Nature Switzerland AG 2022

I. Rojas et al. (Eds.): IWBBIO 2022, LNBI 13347, pp. 346–352, 2022.

[https://doi.org/10.1007/978-3-031-07802-6\\_29](https://doi.org/10.1007/978-3-031-07802-6_29)

[9] introduced a graphical library called Graphite for path analysis. Hipathia [5] used a propagation algorithm to estimate the amount of signal that arrives to the effector protein from the receptor protein.

The Pathologist [3] was the first pathway analysis algorithm to leverage a statistical model of gene expression data to assess the strength of activations in a given group of pathway interactions and to inspect the co-behavior of interacting molecules. A distribution is fitted to data to better estimate the ‘center of mass’ of a gene distribution and thus better estimate its level of activation in each interaction. It is one of the first topological based methods, where the network structure of the pathway is considered as well and not only gene sets. In a recent paper [1] demonstrated the advantages of using this approach over other methods. Our work builds upon this work, improving the original algorithm and its performance, supporting more sequencing platforms, and providing accessible and modifiable code. PathWeigh networks database supports 581 different paths from sources such as KEGG and BioCarta and provides the following novel tools for pathway topology analysis:

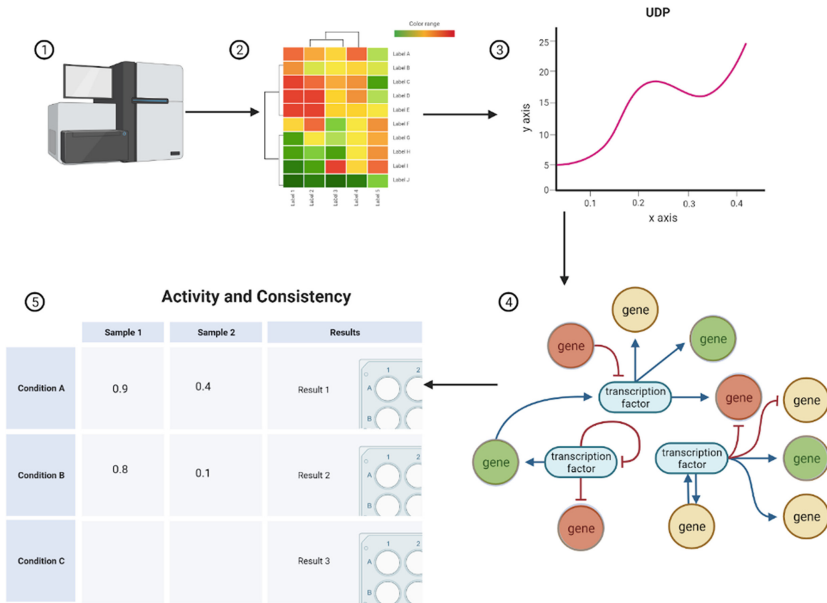
- Using a distributed framework for optimization algorithms allows path activity calculation in seconds.
- Assessing different discrete and continuous distributions for fitting gene expression data.
- Support both microarray and RNA sequencing data from different platforms.
- Optionally support single cell data using extension to our distribution fit framework.
- Provide multiple runtime options: as a desktop application, Python library or a web service.
- API for assessing pathway effectiveness by comparing results of two datasets.

## 2 PathWeigh Algorithm

Our core algorithm can learn the behavior of different pathways given enough data. Input is RNA sequencing of samples from same or different individuals in different treatment or time points. We start by fitting a probability model for each gene in the data. This is done row-wise for each gene, requires enough sample points and might vary depending on the sequencing platform. Using this probability function we move from expression values to probabilities of being in Up or Down state (UDP). This probability is a uniform way of comparing different genes across different interactions. By looking at probability instead of gene expression we can aggregate UDP values to access overall activation.

Next focusing on a specific sample, we assess the activity level of a given pathway. This is done by first calculating the activity level of all the interactions in the pathway, and then aggregate at the pathway level. To determine the activity at the pathway level we average the activities of all interactions.

PathWeigh activities are calculated using pathway data from sources such as KEGG, BioCarta and PID. The pathway file format is a simple text file describing its interactions and can easily be extended to support more pathways. Following is the high-level pipeline of the PathWeigh algorithm (Fig. 1).



**Fig. 1.** PathWeigh pipeline: 1) RNA sequencing. 2) Data after alignment quality control and normalization. 3) Fitting a known distribution per gene to calculate UDP. 4) Calculate interaction activity and consistency. 5) Aggregate for all samples and pathways.

### 3 UDP Fitting

The learning stage includes fitting a known distribution to the RNA sequencing data. Research [2, 10] shows that different RNA sequencing platforms adhere to different distributions of expression values. RNA-Seq data tends to fit a negative binomial distribution, and gene expression microarray data is best described using a mixture model of one or two Gaussian distributions. PathWeigh provides a framework for assessing the fit of other common distributions to support other platforms. The following example shows various tests for fitting different distributions to microarray RNA data of lung cancer (GSE29013). The AIC is the average of 1000 fittings done for 1000 genes (Table 1).

**Table 1.** Fitting various distributions to GSE29013 data.

Distribution	Avg. AIC
Poisson	207
Negative binomial	240
Gaussian mixture	162
Normal	169
Generalized normal	160

We use a Bayesian maximum likelihood estimator with the expectation maximization simulation approach for model fitting. In various discrete and continuous distributions (e.g., Binomial, Poisson and Normal) a closed form formula for the best parameters exists and can be used. For other distributions (e.g., negative Binomial) no such formula is available, hence we developed an iterative numerical estimation algorithm to fit the data. Our optimization algorithm uses a known method for minimization of the likelihood called BFGS [12]. This method allows minimization of unconstrained nonlinear function using gradient descent and considering general curvature for gradient direction. Once a distribution is fitted to data, PathWeigh assigns to each gene in each sample its probability for up or down state.

The negative Binomial distribution is a discrete probability distribution modeling the number of successful IID Bernoulli events before a specific failure happens. Its probability mass function is:

$$f(K; r, p) = P(X = K) = \binom{K + r - 1}{r - 1} (1 - p)^k p^r \quad (1)$$

and its maximum likelihood estimator must be numerically calculated like Newton's method. The pseudo code for fitting this function is:

```
def log_likelihood(r, p):
    N = data.size
    result = np.sum(gammaln(X + r)) \
        - np.sum(np.log(factorial(X))) \
        - N * (gammaln(r)) \
        + N * r * np.log((1 - p)) \
        + np.sum(X * np.log(p))

def fit(data):
    #Initial values
    p0 = np.mean(data)
    v0 = np.var(data)
    r0 = (m ** 2) / (v - m) if v > m else 10
    optimres = bfgs_optim(log_likelihood, x0=(r0,p0),
        args=data, fprime=log_likelihood_deriv)
```

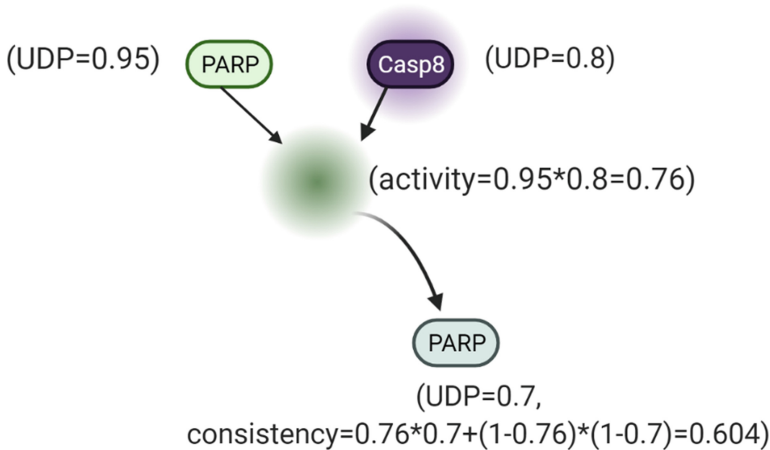
## 4 Activity and Consistency

For every sample and every pathway, we can now evaluate its activity level. This is done by first assessing all the interactions. By using the UDP level of its inputs, and their graphical dependencies, inhibitors or promoters, we can calculate the activity of the interaction. Once all interactions activities are calculated, we aggregate at the pathway level by taking the average.

We also calculate a consistency level for each interaction in the pathway. Consistency assesses how much the interaction activity is consistent with the interaction output molecule's UDP value, in other words, it is the likelihood of getting the output UDP result given the interaction activity. We define consistency as:

$$\text{Consistency}(\text{interaction}) = \text{Likelihood}(\text{output}) = P(\text{active interaction}) * P(\text{output is in "up" state}) + P(\text{inactive interaction}) * P(\text{output is in "down" state}) \quad (2)$$

Here again we aggregate at the pathway level using average. Following is an example of activity and consistency calculations (Fig. 2).



**Fig. 2.** Interaction activity and consistency.

## 5 Visualization

PathWeigh supports few output visualizations options using KEGG's KGML pathway map standard. KGML is a standard describing graphs and networks of biological molecular interactions and is commonly used by researchers. KGML diagrams are often imported to visualization tools such as Cytoscape.

PathWeigh can export a standard KGML file which depicts the graphical layout of a pathway, together with its activity and consistency levels. In the supplemental is an example of such a KGML file.

Another option is using the PathWeigh built in UI graphics to present an outline of the given pathway together with its corresponding activity and consistency values. Following is an example of PathWeigh output in a Google Collab notebook (Fig. 3).

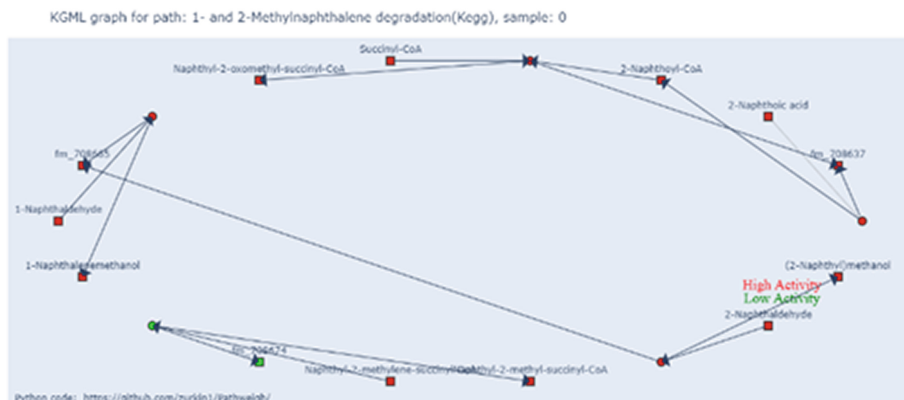


Fig. 3. PathWeigh's output.

## 6 Pathway Assessment

Kim [7] suggests five criteria for assessment of a pathway development tool.

- Reproducibility: how are gene level differences affect the overall activity level.
- Noise robustness: be able to handle noisy data.
- Separation of normal vs. abnormal tissues: either using a classification method or other statistical method.
- Classification of survival information: using clinical data and concordance index.
- Classification of cancer subtypes: using molecular signatures of the sequencing data.

We demonstrate PathWeigh usage for evaluating a novel pathway for the interaction of the BRCA1 gene recently used in our lab, thus being able to separate healthy vs. unhealthy patients.

- We collected the gene expression data from NCBI dataset GSE50948. It contains samples from 156 breast cancer patients and a reference dataset of 51 healthy donors.
- Calculate activity and consistency for a known BRCA1 pathway from BioCarta.
- These values are contrasted using a two-sample rank sum test across the two sample sets.
- The significance of the test would then tell if the pathway is able to efficiently differentiate the two classes of patients.

Here are the results we received when running these tests (Table 2):



**Table 2.** Comparing Results of A newly Developed Path for BRCA show a high P value

Pathway	P value	U stat
brca1 dependent ub ligase activity(BioCarta)	0.003	2977
New path for BRCA	<b>0.008</b>	2814
Role of brca1 brca2 and atr in cancer susceptibility(BioCarta)	2.4e-07	2109

## 7 Summary

PathWeigh is an efficient pathway activity estimator that works with different kinds of sequencing platforms. It uses few algorithmic optimizations to fit parameters to data. It supports parallel run of calculations and other methods for runtime efficiency. PathWeigh is written in Python and can easily be extended to support more sequencing platforms and normalization methods. An online Google Collab based notebook is provided for quickly accessing the tool.

## References

1. Ben-Hamo, R., Jacob Berger, A., Gavert, N., et al.: Predicting and affecting response to cancer therapy based on pathway-level biomarkers. *Nat. Commun.* **11**, 3296 (2020). <https://doi.org/10.1038/s41467-020-17090-y>
2. Conesa, A., Madrigal, P., Tarazona, S., Gomez-Cabrero, D., Cervera, A., McPherson, A., Mortazavi, A. (n.d.): A survey of best practices for RNA-seq data analysis. *Genome Biology*, **17**(1), 13. (2020)
3. Efroni, S., Schaefer, C.F., Buetow, K.H.: Identification of Key Processes Underlying Cancer Phenotypes Using Biologic Pathway Analysis. *PLOS ONE*, **2**(5) (2007). <https://journals.plos.org/plosone/article?id=https://doi.org/10.1371/journal.pone.0000425>
4. Gao, S., Wang, X.: TAPPA: topological analysis of pathway phenotype association. *Bioinformatics*, **23**(22), 3100–3102. (2007) [http://people.vcu.edu/~mreimers/sysbio/gao-topological\\_gene\\_sets\\_analysis.pdf](http://people.vcu.edu/~mreimers/sysbio/gao-topological_gene_sets_analysis.pdf)
5. Hidalgo: High throughput estimation of functional cell activities reveals disease mechanisms and predicts relevant clinical outcomes. *Oncotarget*, 5160–5178 (2017)
6. Karlebach, G., Shamir, R.: Modeling and analysis of gene regulatory network, *Nature* **9** (2008)
7. Kim: Comprehensive and critical evaluation of individualized pathway activity measurement tools on pan-cancer data. *Briefings in Bioinformatics* **21**(1) (2020)
8. Klipp, E., Herwig, R., Kowald, A., Wierling, C., Lehrach, H.: *Systems Biology in Practice*, Wiley-VCH (2005)
9. Martini, P., Sales, G., Massa, M.S., Chiogna, M., Romualdi, C.: Along signal paths: an empirical gene set approach exploiting pathway topology. *Nucleic Acids Res.*, **41**(1), 1–11 (2013). <https://ncbi.nlm.nih.gov/pmc/articles/pmc3592432>
10. Robinson, M.D.: A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biol.* **2010**(11), R25 (2010)
11. Segal: A module map showing conditional activity of expression modules in cancer. *Nature*, 1090–1098 (2004)
12. Wright, J.N.: *Numerical Optimization*. Springer (2006). <http://www.ece.northwestern.edu/~nocedal/book/num-opt.html2020>



# Translational Challenges of Biomedical Machine Learning Solutions in Clinical and Laboratory Settings

Carlos Vega<sup>(✉)</sup> , Miroslav Kratochvil , Venkata Satagopam ,  
and Reinhard Schneider 

Luxembourg Centre for Systems Biomedicine, Université du Luxembourg,  
Esch-sur-Alzette, Luxembourg

{carlos.vega,miroslav.kratochvil,venkata.satagopam,  
reinhard.schneider}@uni.lu

**Abstract.** The ever increasing use of artificial intelligence (AI) methods in biomedical sciences calls for closer inter-disciplinary collaborations that transfer the domain knowledge from life scientists to computer science researchers and vice-versa. We highlight two general areas where the use of AI-based solutions designed for clinical and laboratory settings has proven problematic. These are used to demonstrate common sources of translational challenges that often stem from the differences in data interpretation between the clinical and research view, and the unmatched expectations and requirements on the result quality metrics. We outline how explicit interpretable inference reporting might be used as a guide to overcome such translational challenges. We conclude with several recommendations for safer translation of machine learning solutions into real-world settings.

**Keywords:** Machine learning · Biomedicine

## 1 Introduction

Transfer of machine learning (ML) solutions into laboratory and clinical settings is complicated by many diverse sources of pitfalls. In laboratories, the methods need to be readily used to infer conclusions from data collected using state-of-the-art research methods with unforeseen properties. Likewise, the clinicians need to apply the methods to highly variable, often irreproducible and always-original data from patients. The deficiencies that risk the transferability of such solutions range from reproducibility issues (e.g. data/code/model availability), lack of expert auditing, external validation, to ethical and legal concerns (e.g. informed consent, bias, liability) [20, 24].

Dodging these problems, literature on biomedical ML solutions often focuses on simple evaluation metrics to assess the performance, while disregarding qualities such as adaptability, auditability, and resilience that are key to deploy solutions into real-world settings and prevent data cascades [27].

The growing popularity of ML challenges (and their impact in areas such as biomedical image analysis [16]) only worsens the issue, promoting the ranking of solutions based on goodness-of-fit metrics that do not reflect phenomenological fidelity, and thus their suitability for clinical or lab settings [28]. Even many peer-reviewed studies lack in data and code availability [14], let alone other aspects concerning external validity such as generalisation to different populations or robustness against concept drift and data shift [3, 10]. Moreover, almost all studies employing ML solutions have been retrospective, despite that only through prospective studies we can properly assess external validity and avoid potential biases arising from *ad-hoc* hyperparameter tuning [10].

These issues also extend to unsupervised methods aimed at reduction of problem complexity. For instance, single-cell gene expression studies often rely on dimensionality reduction to produce scrutinizable data visualisations. However, the results are used as well for qualitative and quantitative analysis on the basis that properties such as the structure of the data remain faithfully represented [11]. Although this practice lacks enough theoretical support, such visuals are employed to identify phenotypes and infer cell relationships [4]. Despite of the misuse, we identify a trend where the advanced visualisations are repurposed for quality-control tasks, offering a boost in verifiability and explainability of other ML solutions.

Below, we present two case studies illustrating the impact of aforementioned problems in biomedical solutions.

## 2 Case Studies

### 2.1 Chest X-ray Image Diagnosis of COVID-19

At the pandemic outset, researchers rushed to develop solutions from crowd-sourced repositories<sup>1</sup> to predict the COVID-19 severity and outcome from chest X-ray (CXR) images. Questionable methods and poor annotation of the datasets spawned a multitude of problems [6, 15, 26]. Commonly, the solutions were based on binary or multi-class classification methods, employing either conventional ML as well as deep learning, that considered a small subset of diseases. However, these solutions assume mutual exclusion of the classes while, in fact, many lung diseases may co-exist. For example, COVID-19 and Tuberculosis share abnormalities such as fibrosis and capacities, and produce a spectrum of pathologies that evolves over time, requiring a combination of tests (e.g. blood, sputum) for their diagnosis [21, 31, 34]. Attempts to diagnose lung diseases with just CXR images are thus unnecessarily partial and defy the multimodal nature of diagnosis. For all these reasons, regardless of the reported evaluation metric, binary and multinomial classification solutions are rarely suited for real clinical settings, mainly due to unrealistic assumptions about the nature of the predicted phenomena [26, 30].

---

<sup>1</sup> For example: <https://git.io/J0xva>.

Notably, high output modality issues are common in biomedical research, and attract possible improvements: Waegeman et al. [32] delved into the problem of multi-target (multivariate) prediction (MTP), providing a unifying view of MTP problems to help researchers on method choice. In a related work, Rauschenberger and Glaab [25] employ multivariate regressions for molecular data.

## 2.2 Disease Status Detection in Cytometry

Single-cell measurement techniques [1] have enabled gathering of tremendously detailed datasets of millions of cells as individual data-points, offering a direct way to diagnose many forms of cancer and autoimmune diseases. Multitude of approaches appeared to cluster the observations and predict the disease status, suffering from similar deficiencies as the CXR diagnostics despite the ongoing efforts [5, 8, 13]. In particular, ML studies are often limited to retrospective data from at most several dozen patients, model the outcome with discrete class assignment, and ignore a plethora of data variability (batch effects and patient variability) that jeopardises their transfer to a general medical setting [23]. After a decade of published research, the problematic re-applicability of the methods has resulted in little adoption even in laboratory practice [14, 19, 22].

Lately, researchers began to utilise advanced dimensionality reduction methods to produce interpretable data visualisations [2]. Despite the unavoidable bias caused by the data dimensionality reduction and their often erroneous use as base data for analyses (e.g. clustering), the visuals have proven surprisingly effective in communicating the complicated modality of the data to experts, who use them to infer cell population presence and relationships quite reliably [4, 11]. Conversely, this gave a novel use where the interpretable but likely misleading visualisation could be used for quick quality control of the ML algorithm output [12], where the visualisation may be intuitively used to detect problematic inter-sample variability and failures of clustering or population identification.

## 3 Discussion

These two use-cases show that despite the common pitfalls may be repeatedly observed across the translational medicine landscape, there are viable research directions that may alleviate the problems. These may eventually give a reliable methodology for integrating advanced ML solutions into the laboratory and clinical practice.

The most problematic feature of the ML studies that impedes adoption is the representation of the diagnostic as a multi-class assignment problem that carries almost no resemblance to the clinical reality and the modelled phenomena. In this sense, proper modelling of the ML output, e.g. as multi-modal feature systems or fuzzy assignments, may also alleviate the issues.

As a guideline, sufficient metadata must be available to model such a rich space of possible outcomes. Interpretation of the dataset guided by rich metadata also improves latent space representations [4], which is, in turn, a key property

for training better predictors. Metadata availability may further alleviate the common problem of practitioners who face situations where they need to “work with what they have”, often lacking parts of measurements and the “selective capability” of laboratories to discard poor quality samples, because of the data acquisition limits in clinical settings [27]. In particular, data acquisition issues may result in production of ascertained datasets containing potential collider bias, which undermine both external and internal validity of the solutions [7]. This calls for advocating data accountability and maximising the transparency of the data acquisition process [9]. In this vein, efforts are already being made to increase research reproducibility, data quality, model interpretability and minimise biases in model evaluation and optimisation, such as with the DOME recommendations [33].

As a relevant development, a new set of maintenance and monitoring practices (MLOps) is finding its place within the machine learning workflows, adding continuous testing of both the data and models by continual monitoring of their distribution properties and alerting on data shift events [17, 29]. Thus, the modelling does not end with the ‘deployment’ of the model, but the models are now continuously assessed and iterated for timely adaptation to the evolving reality.

We observed the shift of utilisation of the ML-based visualisation techniques, from serving as a likely biased base for analysis to a valid resource for quick validation of the results of other ML methods. Despite bringing no improvement for actual ML methodology, this enables safer utilisation of the existing ML approaches in the clinical settings [18], where trained personnel may quickly recognise a classification problem and act accordingly, thus increasing the tolerance of the deployment to classifier errors and, collaterally, improving the compliance with existing regulations.

## 4 Conclusion

We have reviewed two recent cases where the utilisation of ML solutions in real-world laboratory and clinical settings has proven problematic. After reviewing the causes of the translational difficulties, we highlight that many of the problems may be alleviated by improvements in dataset annotation with metadata, removing the bias towards simple binary or categorical decisions. Furthermore, novel applications of the visualisation techniques enhance the capabilities of ML solutions by utilising the visualisation as a quality control tool, rather than as an analysis cornerstone. In the long term, we expect that similar improvements may be developed for many other kinds of ML algorithms, potentially advancing the adoption of existing methods and driving the long-term shift towards the wide adoption of translational ML solutions.

## References








1. Adan, A., Alizada, G., Kiraz, Y., Baran, Y., Nalbant, A.: Flow cytometry: basic principles and applications. *Crit. Rev. Biotechnol.* **37**(2), 163–176 (2017)

2. Becht, E., et al.: Dimensionality reduction for visualizing single-cell data using umap. *Nature Biotechnol.* **37**(1), 38–44 (2019)
3. Cabitza, F., Campagner, A.: The need to separate the wheat from the chaff in medical informatics: Introducing a comprehensive checklist for the (self)-assessment of medical ai studies (2021). <https://www.sciencedirect.com/science/article/pii/S1386505621001362>. ISSN 1386–5056
4. Chari, T., Banerjee, J., Pachter, L.: The specious art of single-cell genomics. *bioRxiv* (2021)
5. Ding, J., Condon, A., Shah, S.P.: Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. *Nature Commun.* **9**(1), 1–13 (2018)
6. Cruz, B.G.S., Bossa, M.N., Sölter, J., Husch, A.D.: Public covid-19 x-ray datasets and their impact on model bias - a systematic review of a significant problem. *Med. Image Anal.* **74**, 102225 (2021). <https://doi.org/10.1016/j.media.2021.102225>. <https://www.sciencedirect.com/science/article/pii/S136184152100270X>. ISSN 1361–8415
7. Griffith, G.J., et al.: Collider bias undermines our understanding of covid-19 disease risk and severity. *Nature Commun.* **11**(1), 1–12 (2020)
8. Hu, Z., Tang, A., Singh, J., Bhattacharya, S., Butte, A.J.: A robust and interpretable end-to-end deep learning model for cytometry data. *Proc. Natl. Acad. Sci.* **117**(35), 21373–21380 (2020)
9. Hutchinson, B., et al.: Towards accountability for machine learning datasets: Practices from software engineering and infrastructure. In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pp. 560–575 (2021)
10. Kelly, C.J., Karthikesalingam, A., Suleyman, M., Corrado, G., King, D.: Key challenges for delivering clinical impact with artificial intelligence. *BMC Med.* **17**(1), 1–9 (2019). <https://doi.org/10.1186/s12916-019-1426-2>
11. Kobak, D., Berens, P.: The art of using t-sne for single-cell transcriptomics. *Nat. Commun.* **10**(1), 1–14 (2019)
12. Kratochvíl, M., Bednárek, D., Sieger, T., Fišer, K., Vondrášek, J.: ShinySom: graphical som-based analysis of single-cell cytometry data. *Bioinformatics* **36**(10), 3288–3289 (2020)
13. Li, H., Shaham, U., Stanton, K.P., Yao, Y., Montgomery, R.R., Kluger, Y.: Gating mass cytometry data by deep learning. *Bioinformatics* **33**(21), 3423–3430 (2017)
14. Littmann, M., et al.: Validity of machine learning in biology and medicine increased through collaborations across fields of expertise. *Nature Mach. Intell.* **2**(1), 18–24 (2020). <https://doi.org/10.1038/s42256-019-0139-8>
15. Maguolo, G., Nanni, L.: A critic evaluation of methods for covid-19 automatic detection from x-ray images. *arXiv preprint arXiv:2004.12823* (2020). <https://arxiv.org/abs/2004.12823v1>
16. Maier-Hein, L., et al.: Why rankings of biomedical image analysis competitions should be interpreted with care. *Nature Commun.* **9** (2018). <https://doi.org/10.1038/s41467-018-07619-7>. Art. no. 5217
17. Mäkinen, S., Skogström, H., Laaksonen, E., Mikkonen, T.: Who needs mlops: what data scientists seek to accomplish and how can mlops help? In: *2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN)*, pp. 109–112. IEEE (2021)
18. Marcinkevičs, R., Vogt, J.E.: Interpretability and explainability: A machine learning zoo mini-tour. *arXiv preprint arXiv:2012.01805* (2020)

19. McKinnon, K.M.: Flow cytometry: an overview. *Current protocols in immunology*, **120**(1), 5–1 (2018)
20. Morley, J., et al.: The ethics of AI in health care: a mapping review. *Soc. Sci. Med.* **260** (2020). <https://doi.org/10.1016/j.socscimed.2020.113172>Get. Art. no. 113172
21. Mousquer, G.T., Peres, A., Fiegenbaum, M.: Pathology of tb/covid-19 co-infection: the phantom menace. *Tuberculosis* **126** (2020). <https://doi.org/10.1016/j.tube.2020.102020>. Art. no. 102020
22. Nagendran, M., et al.: Artificial intelligence versus clinicians: systematic review of design, reporting standards, and claims of deep learning studies. *Bmj* **368** (2020)
23. Pedersen, C.B., Olsen, L.R.: Algorithmic clustering of single-cell cytometry data—how unsupervised are these analyses really? *Cytometry A* **97**(3), 219–221 (2020)
24. Price, W.N., Gerke, S., Cohen, I.G.: Potential liability for physicians using artificial intelligence. *Jama* **322**(18), 1765–1766 (2019). <https://doi.org/10.1001/jama.2019.15064>
25. Rauschenberger, A., Glaab, E.: Predicting correlated outcomes from molecular data. *Bioinformatics* (2021). <https://doi.org/10.1093/bioinformatics/btab576>
26. Roberts, M., et al.: Common pitfalls and recommendations for using machine learning to detect and prognosticate for covid-19 using chest radiographs and CT scans. *Nature Mach. Intell.* **3**(3), 199–217 (2021). <https://doi.org/10.1038/s42256-021-00307-0>
27. Sambasivan, N., Kapania, S., Highfill, H., Akrong, D., Paritosh, P., Aroyo, L.M.: “Everyone wants to do the model work, not the data work”: Data cascades in high-stakes AI. In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–15 (2021)
28. Sculley, D., Snoek, J., Wiltschko, A.B., Rahimi, A.: Winner’s curse? on pace, progress, and empirical rigor. In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30–3 May 3, 2018, Workshop Track Proceedings*. OpenReview.net (2018). <https://openreview.net/forum?id=rJWF0Fywf>
29. David Sculley, D., et al.: Hidden technical debt in machine learning systems. In: *Advances in Neural Information Processing Systems*, 28 (2015)
30. Vega, C.: From Hume to Wuhan: an epistemological journey on the problem of induction in COVID-19 machine learning models and its impact upon medical research. *IEEE Access* **9**, 97243–97250 (2021). <https://doi.org/10.1109/ACCESS.2021.3095222>
31. Visca, D., et al.: Tuberculosis and covid-19 interaction: a review of biological, clinical and public health effects. *Pulmonology* **27**(2), 151–165 (2021). <https://doi.org/10.1016/j.pulmoe.2020.12.012>. ISSN 2531–0437
32. Waegeman, W., Dembczyński, K., Hüllermeier, E.: Multi-target prediction: a unifying view on problems and methods. *Data Min. Knowl. Disc.* **33**(2), 293–324 (2018). <https://doi.org/10.1007/s10618-018-0595-5>
33. Walsh, I., et al.: Dome: recommendations for supervised machine learning validation in biology. *Nature Methods* **18**(10), 1122–1127 (2021)
34. Yousaf, Z., et al.: Cavitory pulmonary tuberculosis with covid-19 coinfection. *IDCases* **22** (2020). <https://doi.org/10.1016/j.idcr.2020.e00973>. Art. no. e00973



# Human Multi-omics Data Pre-processing for Predictive Purposes Using Machine Learning: A Case Study in Childhood Obesity

Álvaro Torres-Martos<sup>1</sup> , Augusto Anguita-Ruiz<sup>1,2,3,4</sup> ,  
Mireia Bustos-Aibar<sup>1</sup> , Sofia Cámara-Sánchez<sup>1</sup> , Rafael Alcalá<sup>5</sup> ,  
Concepción M. Aguilera<sup>1,2,3</sup> , and Jesús Alcalá-Fdez<sup>5</sup> 

<sup>1</sup> Department of Biochemistry and Molecular Biology II, School of Pharmacy, University of Granada, 18071 Granada, Spain

alvarotorresmartos@gmail.com, augustoanguitaruiz@gmail.com, mireia251019@gmail.com, s.camarasanchez@gmail.com, caguiler@ugr.es

<sup>2</sup> Institute of Nutrition and Food Technology “José Mataix” Center of Biomedical Research, Instituto de Investigación Biosanitaria IBS.GRANADA, Complejo Hospitalario Universitario de Granada, University of Granada, Avda. del Conocimiento s/n., 18016, 18012 Granada, Spain

<sup>3</sup> CIBEROBN (CIBER Physiopathology of Obesity and Nutrition), Instituto de Salud Carlos III, 28029 Madrid, Spain

<sup>4</sup> Barcelona Institute for Global Health (ISGlobal), Doctor Aiguader 88, 08003 Barcelona, Spain

<sup>5</sup> Department of Computer Science and Artificial Intelligence, Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Granada, 18071 Granada, Spain  
{alcala,jalcala}@decsai.ugr.es

**Abstract.** The Machine Learning applications in the medical field using omics data are countless and promising, highlighting the possibility of creating long-term predictive models for highly prevalent diseases. Nevertheless, to take advantage of the virtues of omics data and machine learning tools, we first need to perform adequate data pre-processing just as taking some considerations for the constructions of the models. The present paper is an example of how to face the main challenges encountered when constructing machine learning predictive models with multi-omics human data. Some topics covered in this work include a description of the main particularities of each omics data layer, the most appropriate pre-processing approaches for each source, and a collection of good practices and tips for applying machine learning to this kind

---

Supported organization in part by ERDF/Regional Government of Andalusia/Ministry of Economic Transformation, Industry, Knowledge and Universities (grant numbers P18-RT-2248 and B-CTS-536-UGR20) and by the ERDF/Health Institute Carlos III/Spanish Ministry of Science, Innovation and Universities (grant number PI20/00711, PI16/00871 and PI20/00563).

Á. Torres-Martos and A. Anguita-Ruiz—Equal contributors.

© Springer Nature Switzerland AG 2022

I. Rojas et al. (Eds.): IWBBIO 2022, LNBI 13347, pp. 359–374, 2022.

[https://doi.org/10.1007/978-3-031-07802-6\\_31](https://doi.org/10.1007/978-3-031-07802-6_31)



of data with predictive purposes. Using real data examples (blood samples), we illustrate how some of the key issues are addressed in this kind of research (technical noise, biological heterogeneity, class imbalance, high dimensionality, and presence of missing values, among others). Additionally, we set the basis for future work incorporating some proposals to improve models, arguing their need according to encountered insights.

**Keywords:** Multi-omics · Data pre-processing · Machine learning · eXplainable Artificial Intelligence

## 1 Introduction

The biomedical field has undergone a true big data revolution during the past decades. Starting with the appearance of the first microarray technologies, our analytic ability has grown exponentially and now we can perform almost any type of molecular analysis at a genome-wide scale, generally referred to as ‘omics’ analysis. From our ability to identify alterations in the DNA sequence by Genome Wide-Association Studies (GWAS), passing through the study of gene expression levels by RNAseq experiments, or the possibility to study chemical environment-inducible DNA modifications with Epigenome Wide-Association Studies (EWAS), omics technological advances have led to major breakthroughs in our fundamental understanding of cell biology. Likewise, they have promised a true revolution for the clinical treatment and management of many diseases. Particularly, one of the most promising clinical applications of omics technologies has involved the generation of predictive panels of biomarkers for personalized disease risk estimation and the consequent implementation of stratified clinical guidelines. For that purpose, omics technologies have further taken advantage of the recent advances in the machine learning (ML) field. ML is a research sub-branch of Artificial Intelligence that has recently experienced a notable boost due to its ability to automatically generate predictive and descriptive models from massive amounts of data. Within the context of predictive modelling, more and more sophisticated ML algorithms have become available; highlighting ensemble models or the recent revolution of deep learning [7].

Despite all mentioned benefits and potential applications emanating from omics and ML fields, the major obstacles in translating these promises into tangible predictive models in the daily clinic remain unsolved. Most of the encountered challenges are related to the implementation of adequate analytic pipelines; a situation that is worsened by the shortage of suitably trained professionals to perform such complex data analysis tasks. Omics data present a complex nature with huge differences across omics platforms, different needs for pre-processing steps, intense variability within and between human subjects, and the ubiquitous problem of high-dimensionality-and-low-sample size settings.

The selection of the most suitable pre-processing pipeline for each omics layer and the choice of the most appropriate ML model are critical steps that must take place considering the particularities of human datasets and depending on

the purpose of each predictive modelling. This problem increases if we keep in mind the need for creating interpretable models. Following this line, to face this need have emerged the recent eXplainable Artificial Intelligence (XAI) revolution that recommends the use of transparent models that are easy to understand by human users; and are especially relevant for medical applications [2].

Taking all this into account, in the present paper we will review some of the particularities that make predictive modelling with multi-omics data a challenging task and will propose adequate solutions that are currently employed in ML-omics research. In order to illustrate the process, we will present a case study consisting of the creation of a predictive ML model on a longitudinal design in children with obesity and metabolic dysfunction. In this dataset, a series of multi-omics data layers (GWAS and EWAS), as well as, biochemical and clinical variables will be available at a pre-pubertal stage. Likewise, the metabolic status (insulin-resistance or IR) reached by each child when entering puberty will be available. The main goal of the project will be constructing a robust ML predictive model that using multi-omics and biochemical pre-pubertal data is able to predict the IR status of each child [7].

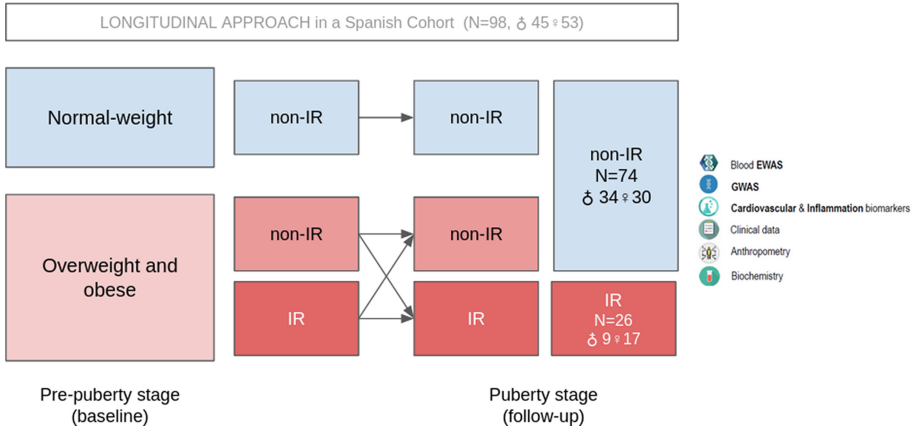
The paper follows into different sections. Section 2 describes the research problem and the datasets employed. Section 3 introduces the main faced challenges in omics ML predictive modelling commonly. Section 4 proposes specific data pre-processing guidelines and different analytical solutions for mentioned challenges. Section 5 gathers the basis and recommendations that must guide the selection of an ML algorithm and the experimental design. Section 6 presents the main results and insights from our case study. Finally, Sect. 7 is for exposing the concluding remarks.

## 2 Description of the Case Study Population and Data

The PUBMEP (“PUBberty and Metabolic risk in obese children. Epigenetic alterations and Pathophysiological and diagnostic implications”) project is a longitudinal research study in which children with and without obesity are followed from pre-puberty to puberty evaluating the prevalence of metabolic syndrome and the progression of the cardiometabolic risk factors related to it. In this population, a series of multi-omics analyses have been conducted with the aim of discovering new and promising blood molecular biomarkers of IR during the metabolically critical period of puberty (see Fig. 1) [1].

IR is one of the metabolic disturbances derived from obesity that present the earliest appearance in patients. If not properly addressed, IR finally results in the development of more severe diseases such as cardiovascular disease or Type II Diabetes. For this reason, IR has become a cornerstone in preventing obesity-associated morbimortality. In the PUBMEP study, a population composed of 90 Spanish children (47 females) were allocated into two experimental groups according to their IR status (IR or non-IR) after the onset of puberty (see Fig. 1). The number of children with the respective distribution of sex in each group can be found also in the Fig. 1. In this population, as mentioned in the introduction

section, pre-pubertal ( $T_0$ ) data (GWAS, EWAS, clinical, anthropometrical and biochemistry) were employed as predictors for the IR status at the pubertal stage ( $T_1$ ). For that purpose, a number of pre-processing steps and ML models were implemented as detailed below. A wide description of the PUBMEP project can be found elsewhere [1]. In the current paper, datasets were divided into GWAS, EWAS, and Biochemistry (which also incorporated data from anthropometry and clinical history).



**Fig. 1.** Summary of PUBMEP project.

Children from the PUBMEP project were recruited in **three different Spanish cities**: Santiago de Compostela, Zaragoza and Córdoba. As it will be detailed below, special attention was paid to this, considering the origin as a substantial source of confounding during analyses.

### 3 Main Challenges that are Usually Faced in Omics ML Predictive Modelling

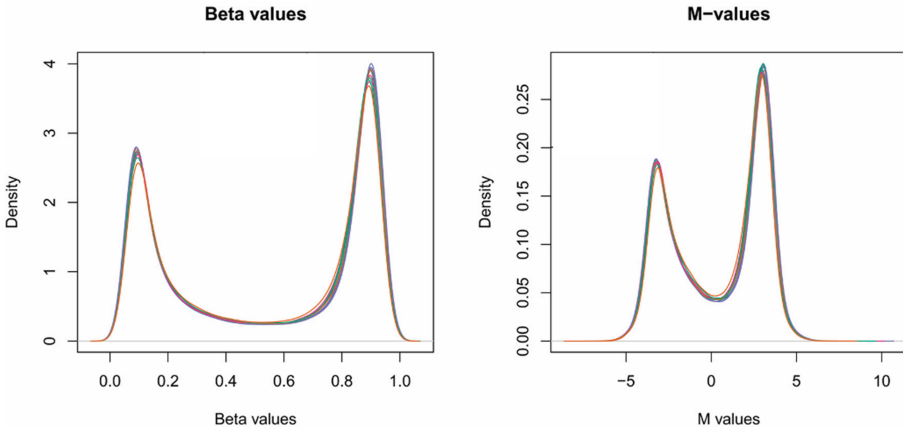
One of the main problems related to the work with human data is the difficulty of patient recruitment, the access to invasive biopsies and the high costs associated with omics technologies, which directly result in studies with a relatively low sample size. The issue gets even more problematic in the context of omics data, where we have millions of variables measured, massively increasing the rate of false-positive discoveries (Curse of dimensionality). In the context of ML predictive modelling, the low sample sizes available in omics studies along with the huge search space have direct effects on the performance of constructed models (increasing computational burden and inducing overfitting). For these reasons, it is indispensable to perform feature-selection steps, previous to model training. As we will see in the next sections, there are several ways of performing a feature

selection, and the choice of one or another method will depend mainly on the type of data and research problem to work with. Another common challenge of human research is the high presence of unbalanced design (in which one class is over-represented concerning the other). Often, this happens in a context in which the disease under study is not frequent and the recruitment of patients is a complicated task. As it happens with the low sample size and the high-dimensionality problem, the presence of unbalanced design directly affects ML predictive models inducing overfitting. Again, a lot of solutions have been proposed to face this problem (undersampling, oversampling,...) and the selection of the most appropriate method will strongly depend on the characteristics of the sample. In our case study, we will demonstrate how the undersampling solution is one of the most recommendable for the majority of the human context (avoiding introducing additional noise to the data). Another issue of importance when dealing with human data is the **strong variability that exists between subjects**. To deal with it, it is of vital importance the development of a good study/experimental design, minimizing sources of bias (randomizing subjects, balancing sex and ages across recruiting centres, controlling the batch effect, etc.). Moreover, it will be crucial to validate the findings in an external population to ensure that our model is robust. For that purpose, in those cases in which it is not possible to recruit additional patients, there exist several iterative validation solutions based on cross-validation methodologies which are indispensable[10].

Focusing now on the characteristics of omics data, there are also a range of particularities (inherent to each platform) that should be highlighted, and which involve the need of implementing different pre-processing procedures for each molecular layer. In all omics analyses, there is a background noise or unwanted source of variability which is inherent to technical laboratory analyses. This heterogeneity is therefore not related to the biological question under study and must be removed from the analysis. Background **noise** due to technical procedures usually differs not only across omics types but also across the different technological platforms normally employed for the analysis of each omic (intra- and inter-omics variability).

GWAS refers to any observational study of a genome-wide set of genetic variants or single nucleotide polymorphisms (SNPs) in different individuals to see if any variant is associated with a trait. GWAS are evaluated by the use of microarrays and therefore are subjects to the **usual problems** that typically affect these technologies; erroneous genotype call assignments due to poor quality of DNA samples, poor DNA hybridization to the array, poorly performing genotype probes, and sample mix-ups or contamination. Moreover, although currently available GWAS platforms map a great number of SNPs (500.000 SNPs), there are still many unmeasured variants with interest for disease prediction that could be imputed through appropriate procedures ([3]). In order to deal with these and other problems, several **quality control filters** are usually applied in GWAS research (e.g., assessing missingness of SNPs and individuals, evaluating sex discrepancies according to sex chromosomes, filtering by minor allele

frequencies, controlling Hardy-Weinberg equilibrium (HWE), heterozygosity or population stratification). Another particularity of genetics data is the existence of a certain linkage between SNPs, which means that some groups of SNPs are inherited in blocks (i.e., their minor alleles are inherited as a complete allelic phase). These SNPs are therefore redundant for predictive purposes and a previous pruning step must be always performed before passing GWAS data to a ML model.



**Fig. 2.** Comparison between  $\beta$  and M values. This image has been taken from [12]

$$\beta = \frac{M}{M + U}$$

$$M\text{value} = \log_2\left(\frac{M}{U}\right)$$

In EWASs, the DNA methylation (DNAm) status across the whole genome is interrogated at the CpG level. For each molecule of DNA in a single cell, DNAm is a binary entity, in that at any cytosine it is either present or absent. However, as DNAm studies profile either **bulk tissue** - comprising multiple cell types - or a population of purified cells, DNAm measurements for CpGs are always reported as continuous values representing the proportion of methylated CpGs for a DNA position. Regardless of the Illumina microarray version employed, for each CpG, there are two measurements: a methylated intensity (denoted by M) and an unmethylated intensity (denoted by U). These intensity values can be used to determine the proportion of methylation at each CpG locus. Methylation levels are commonly reported as either  **$\beta$  values** or **M-values** (see the formulations and the Fig. 2). M-values have more robust statistical properties and for that reason, they are preferred in ML tasks than beta values, which have a better

biological interpretation and are frequently used for visualizing data. A detailed comparison of M-values and  $\beta$  values is available elsewhere [12].

As it happens for GWAS, EWAS data is also subject to many sources of **unwanted variability**, some of them deriving from the microarray nature of analytic platforms and some others not; detection errors, the existence of cross-reactive probes, need for special treatment of sex-chromosome located probes, or the need for data normalization of fluorescence intensity raw signals (to address within and between-subject variability). Regarding normalization processes, although there is no single method that is universally considered best, the Functional normalization method is most appropriate for datasets with global methylation differences between different tissue types [6], and the Beta-Mixture Quantile (BMIQ) method is considered a golden standard to deal with datasets in which not big differences are expected in terms of DNAm between samples (e.g., when all samples derive from the same tissue [21]).

The fact of EWASs analyzing a mix of cells in a tissue is also an important issue as, in some cases, tissues present infiltration of other cell types or are so heterogeneous that might confound the findings. In the case of blood sample types, which is the most common one analyzed in EWASs, there is an important part of variability coming from the **white cells proportions** that each individual presents. Therefore, and especially when dealing with diseases with an inflammatory component (such is the case of obesity), it will be extremely important to correct findings by the proportion of white cells types that every subject has, as it might affect the DNAm findings and thereby confound the effects of DNAm on disease. This is usually solved in EWAS through the use of the houseman procedure which deduces the proportion of white cell types in each subject. Then, estimated proportions can be included as confounding variables in models.

Beyond mentioned technical sources of variability associated with each technology, there are also other particularities affecting data pre-processing that are of special importance when one wants to predict an outcome. In the case of GWAS data, it is a fact that certain diseases present a **strong polygenetic architecture**; being many the number genes involved in disease development and progression. Moreover, associated genes usually present small individual effects on the phenotype. So, it's the accumulation of many small-effect variants that constitutes a susceptibility profile. Regarding EWAS, it is a fact how **environmental confounders** may strongly affect the epigenetics patterns. For this reason, it is well-known that the findings from a study population are not easily extrapolated to another different population.

## 4 Data Pre-processing Guidelines and Analytical Solutions for Mentioned Challenges

In this section, we review step by step the methods that were used to approach some of the mentioned problems in the previous section for the different datasets (GWAS, EWAS, and Biochemistry).

#### 4.1 GWAS Data

Regarding genomics, data were generated using a Bead Chip named Infinium Global Screening-24-v3.0. Some quality control filters were implemented in PLINK software before statistical analysis. **The quality control filters, which were applied to treat the mentioned technical problems,** were: 1) Exclusion of SNPs with a missing data rate  $\geq 5\%$  and individuals with a missing data rate  $\geq 20\%$ . 2) Exclusion of SNPs with a allele frequency  $< 5\%$  or Hardy-Weinberg Disequilibrium p-value  $< 0.05$  [13] in control subjects. Additionally, **we removed ambiguous SNPs,** using the GenotypeHarmonizer software[4], which is an indispensable step if we want to impute missing variants. **Next, imputation of missing data for available SNPs in the array was conducted with the Beagle software.** Additionally, this software allows performing **genotype imputation** for unmeasured SNPs making use of a reference population. As a result, the number of available markers in the array could increase to millions. Although this second missing data imputation procedure was not incorporated in our current study, we consider it as an interesting option for future works. From the application of all introduced filters, 467,526 SNPs remained in the final dataset.

When talking about the genetic basis of a disease, there are several modes of inheritance we could assume (autosomal dominant, recessive, co-dominant, etc.), and this will have direct effects on the way we code data and construct predictive ML models. In the case of obesity, as we have mentioned in the previous section, we are in front of a complex trait with a strong polygenic and additive nature (the accumulation of many small-risk effects SNPs is what constitutes a high-risk profile). Considering this, GWAS data were encoded following the **additive model** in this work. For that reason, we propose to use a **dosage format** (.raw) to do the classification task: this format is easily obtained through PLINK software. The dosage format indicates the presence or absence of a risk/reference allele in a SNP encoded as 0, 1, or 2. This format transformation allows having numerical genetic variables making it suitable for the algorithms' learning process [13].

Regarding feature selection before ML construction, here we opted by selecting a subset of SNPs from the whole array according to previous evidence in the literature. Particularly, we collected highlighted SNPs in meta-analysis since these are the studies with the highest degree of evidence, ensuring a strong statistical power to detect the small effects that each SNP could exert on the phenotype. For that purpose, **we performed a literature search and selected 7 articles performing meta-analysis in huge populations of predominantly European descent** [11,17,23]. A limitation of this work is the age study because all articles are in adults not children [8,15,16]. From these articles, we selected a list of 146 SNPs strongly associated with Type 2 Diabetes [19]. Only 46 SNPs from the initial list were available in our GWAS. In these cases, it is recommendable therefore to perform a genotype imputation for missing SNPs increasing the number of genetic variants available in the dataset. As

we said, future lines of work will involve the imputation of missing SNPs before constructing ML models.

## 4.2 EWAS Data

EWAS data were generated through the use of the Infinium MethylationEPIC 850K from blood samples. To remove any source of technical variability, poor performing probes were filtered out according to different criteria: probes with a detection p-value above 0.01 in more than 10 cross-reactive probes aligning to multiple locations (number of probes = 25,570) and probes located on the Y chromosome (number of probes = 246). Regarding normalization, we applied BMIQ normalization, which affects only biased type II probes, though watermelon R package [21]. The selection of this normalization method was argued in the fact that all samples under study were obtained from the same tissue (blood). Lastly, we obtained the  $\beta$  and M values of 834,371 CpG sites [12].

Here, the feature-selection procedure consisted of the application of an **agnostic EWAS**, a type of feature selection in which have been extracted the differential methylated CpG sites associated with IR across the whole genome (hypothesis-free). This procedure was conducted in an independent population study with the same origin as our study population, being some samples coincident. The study population, which has facilitated the EWAS agnostic, is part of a study in 139 children (76 girls) including longitudinal and cross-sectional approaches and following the same experimental design. From this approach, we selected 267 CpG sites. More details about the selection of these CpG sites could be found on references [1]. The choice of performing an agnostic EWAS for the phenotype of interest (IR) instead of relying on literature findings as in the case of GWAS was motivated by the fact of epigenetics findings are strongly conditioned by the characteristics and environmental exposures of each population. On this matter, having an independent sample with the same characteristics as the current study cohort was a better option than selecting CpGs according to European population studies (among which children studies are scarce).

## 4.3 Biochemistry, Anthropometrical and Clinical Data

The last dataset referenced as the Biochemistry data set; is the combination of data of diverse origin as mentioned in previous sections. This dataset consisted of 49 input variables related to the pubertal IR problem. In this dataset, the main problem to address was the presence of missing values. The structure of missing data in our cohort was checked (Missing completely at random (MCAR), missing at random, missing not at random, and structurally missing). Then, we revised several imputation methods as mean/median imputation, knn imputation, bagged trees [10], Multiple Imputation by Chained Equations (MICE) [22] and missForest [20]. Finally, we chose the missForest method for several reasons: it is a non-parametric method that can impute continuous and categorical features, it does not need tuning parameters because of their robust performance,



and does not require assumptions about the distribution of the features. This method was used through missForest R package [20].

#### 4.4 Future Perspectives on Pre-processing

Future lines of work to improve current guidelines might include; 1) Performing imputation of missing SNPs in order to increase the number of genetic variants available in the GWAS, and consequently, the number of literature SNPs to be included into ML models, 2) Evaluating **the performance of other feature-selection methods** beyond our classical proposals, highlighting the method recursive feature elimination (RFE) or other multivariate methods such as LASSO (Least Absolute Shrinkage and Selection Operator), Ridge regression, or Elastic net, strongly used in biological sciences [10].

## 5 Basis and Recommendations that Must Guide the Selection of a ML Algorithm and the Experimental Design

### 5.1 Experimental Design

After completing individual pre-processing procedures, three different datasets (GWAS, EWAS and biochemistry) were obtained. Each dataset has 1 response variable with 2 classes differentiated (IR and non-IR) in 90 children. The number of input features by dataset was 46, 267 and 34 for the GWAS, EWAS and biochemistry data, respectively.

Although a promising approach would have involved the simultaneous modelling of several omics layers along with biochemistry data together, merging so much information in a single model would also increase the problem of high dimensionality. Moreover, the different nature of each dataset makes it indispensable to take a first look at the models constructed separately, so we can understand the amount of valuable information available in each source. In this work, as a preliminary approach, we propose therefore to generate independent ML predictive models for each data layer, while letting the multi-omics modelling as a pending task for future works. Our approach allowed us to extract the predictive information from the different biological layers and identify the most important variables for the IR problem without falling into the undesired overfitting [10].

One of the most important good practices in the ML field is to train the algorithms on a set of individuals that is different from the set aimed to evaluate the model performance. If it is not possible to access an independent population, then, it is established that the training and test sets must be selected iteratively from the same population through a process which is known as cross-validation (CV). There are several types of CV: Leave One Out (LOOCV), Montecarlo CV, Bootstrap, k-fold CV, and repeated k-fold CV. Generally, it is preferred to use by default k-fold CV because it presents the average estimations with the

least possible error. Choosing the right validation methodology according to the characteristics of the data is key for avoiding getting wrong conclusions from models [10].

Another important factor is that the learning process should be as much homogeneous as possible in every iteration. That is to say, the distribution of the variables as well as the class proportion should be the same in the training and test sets of each iteration from the CV process. In this paper, a **stratified repeated k fold cross validation** was used to evaluate the model performance. This approach has been pointed in the literature as one of the best CV procedures to reduce the variability from the average classification metrics in the case of low sample size designs. Although other CV methodologies such as LOOVC have also been commonly used in the context of low sample sizes, we still recommend the use of repeated k fold cross validation for similar studies where the sample size is low because this methodology has the lowest estimation error [10].

As it can be seen in Fig. 1, the datasets from the case study present a severe class imbalance which could lead to overfitting in terms of the majority and negative class. Considering this, oversampling and undersampling techniques were tested on the training sets to “balance” the learning procedure. The resampling method with the best performance in our case study was the nearmiss undersampling, which was implemented through the R package themis [9]. To confirm that learning has occurred equally in both classes it is necessary to evaluate the performance of the models by looking at different classification metrics [10].

## 5.2 Selection of ML Algorithms and Classification Metrics

Another point of debate when constructing a predictive model is the choice of the ML algorithm and the metrics to be used, which will be strongly conditioned by the objective to be pursued. For example, in the case of seeking a model with high predictive ability, neural networks, support vector machines or random forests can be valuable options. However, if a model is meant to be implemented in a hospital, the clinician must understand how the algorithm takes decisions. In these cases, we would opt for more interpretable models such as decision trees or other rule-based methods. The objective motivating the creation of the model will also affect the choice of metrics used. In our case study, following the XAI recommendations [2], the selection of ML algorithms was made pursuing a balance between accuracy and interpretability. Some of the most interpretable algorithms, as mentioned in the literature [5], are C4.5, Ripper, PART, and C5.0 which are implemented in R through the caret package with the names J48, JRip, PART, and C5.0Rules. In this case, we choose to use C4.5, which is a classical and popular algorithm among professionals with the default parameters (Confidence Threshold = 0.35, Minimum Instances Per Leaf = 2). Regarding metrics, since we are much more interested in adequately predicting the minority/positive class, we should not focus on accuracy and specificity but sensitivity [10].

**Table 1.** Classification metrics obtained by C4.5 models with/without undersampling in training sets.

Metrics	Datasets			Datasets (undersampling)		
	GWAS	EWAS	Biochemistry	GWAS	EWAS	Biochemistry
Accuracy	0.60 (0.10)	<b>0.65(0.13)</b>	<b>0.65(0.09)</b>	0.51 (0.10)	0.56 (0.13)	0.59 (0.12)
Sensitivity	0.32 (0.22)	0.40 (0.19)	0.35 (0.18)	0.51 (0.19)	0.50 (0.22)	<b>0.60(0.20)</b>
Specificity	0.71 (0.14)	0.75 (0.15)	<b>0.78(0.09)</b>	0.51 (0.12)	0.58 (0.17)	0.58 (0.16)
PPV	0.73 (0.08)	0.75 (0.08)	0.75 (0.07)	0.72 (0.09)	0.74 (0.12)	<b>0.78(0.09)</b>
NPV	0.31 (0.17)	<b>0.43(0.23)</b>	0.39 (0.18)	0.30 (0.10)	0.34 (0.13)	0.38 (0.12)
AUC	0.52 (0.1)	0.58 (0.13)	0.56 (0.11)	0.51 (0.11)	0.54 (0.13)	<b>0.59(0.13)</b>

## 6 Main Results and Insights from the Case Study

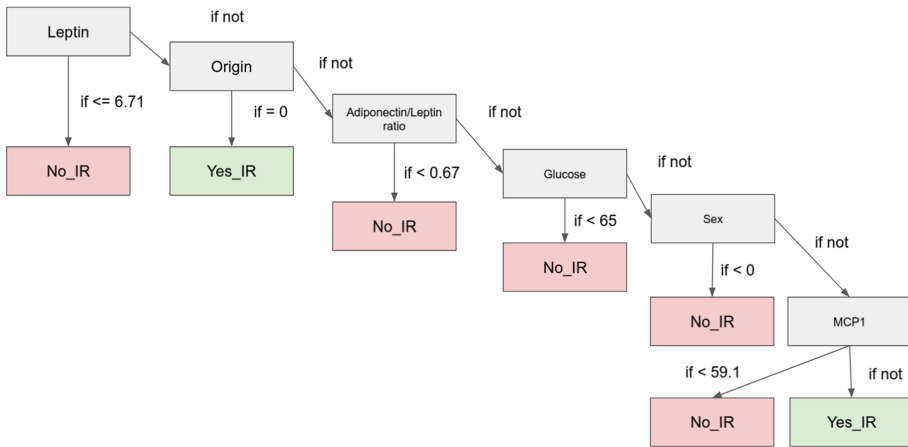
The results obtained after applying the C4.5 algorithm under a stratified repeated k fold cross-validation in each dataset separately (with/without nearmiss undersampling) can be found in the Table 1.

Regarding genetics, it is interesting how models have been overfitted to the negative/majority class (see the differences between sensitivity and specificity values). Looking at the Accuracy one might think that these models classify well but this is only true for the negative/majority class (please, take a look at specificity values). This is relevant because the objective of our case study was exactly the opposite, to be able to predict the positive/minority class correctly. Despite using undersampling methods to avoid overfitting, it can be observed that the classifiers constructed on GWAS data were not better than randomly assigning individuals to one or another class (Area Under Roc Curve or AUC  $\approx 0.5$ ). These results make us think that the GWAS data itself does not contain useful information patterns for predictive tasks. Among other reasons, this might be attributed to the complex genetic architecture of obesity traits and the additive effects of SNPs on disease risk (they are thousands of SNPs, with small risk-effects on the phenotype, which constitute a high susceptibility profile). Regarding epigenetics, the overfitting also occurred in terms of the negative/majority class. Despite using undersampling to try to improve sensitivity through balanced learning, the sensitivity maintained too low to be considered ( $\approx 0.5$ ). Lastly, seeing the Table 1 we can conclude that using undersampling successfully reduced the overfitting in the Biochemistry dataset. Particularly, the values of Sensitivity were almost double when the undersampling was applied (0.32 vs. 0.6). The biochemical dataset, therefore, provided the patterns with the most useful predictive information, achieving the best values in most of the metrics.

Obtained results for the biochemistry dataset are not surprising, since it included phenotypic, anthropometric, and clinical variables with a direct implication in the development of the disease (e.g., origin, or waist circumference), which indeed are currently used in the daily clinic to estimate the risk of metabolic syndrome in children with obesity. On the other hand, the poor performance of C4.5 in the omics datasets could be argued in several facts: the elapsed time between temporal points (comprising several years) and the biological heterogeneity of omics data. Particularly, in the case of genetics, it is crucial how we select input SNPs, and how we pass such information to the ML model. As we previously mentioned, obesity and other complex traits involve a complex poly-genetic architecture and it has been demonstrated that directly using individual SNPs is not the best tool for predictive purposes. Otherwise, the construction of risk scores (which could be also extended to EWAS and environmental data), has been pointed as a powerful tool to account for the complex structures of omics data and the best way to predict long-term outcomes. In risk scores, we can gather information for thousands of SNPs (or variables), so we reduce the problem of dimensionality at the time we also model the complex structures of omics. In future works, therefore, **others ways of encoding omics data as genetic, methylation or metabolic risk scores (polygenetic risk score) should be explored.** Likewise, performing an appropriate **feature-selection about omics data which presents imbalance class is an unsolved task** in the omics ML field. For that reason, some **multivariate methods could be tested checking their promising ability to deal with omics data to reduce their high dimensionality** [10].

Another point to be considered for future work, and evidenced as crucial according to our results, it is the **integration of multi-omics data with bio-chemistry/clinical data in a single model.** Despite it, such combination procedures are not always as straightforward as putting all data together into the same model. An example of these exciting and promising approaches for integrating the multi-omics data can be found in the Omics Data Integration Project (mixOmics R package) [14] proposing the use of multivariate methods such as Principal Component Analysis, Projection to Latent Structures, Canonical Correlation Analysis, and DIABLO to reduce dimensionality [18].

Finally, to demonstrate the high interpretability and the great utility for predictive purposes of ML models such as C4.5 in the medical domain, we plotted the important variables employed by the model and proposed the cuts off; see Fig. 3. Some variables which appear in Fig. 3 are widely described in the bibliography and of vital importance for the IR problem. For example, we can highlight the role of Leptin in the development of IR or the importance of confounding variables such as Origin. The fact of being able to see and understand the model working is key and one of the basis from the XAI easing the decision making and the new knowledge extraction [2].



**Fig. 3.** Generated model through biochemistry dataset.

## 7 Conclusion

This paper is an example of how to face the main challenges encountered when constructing ML predictive models with multi-omics human data. Some topics covered in this work include a description of the main particularities of each omics data layer, the most appropriate pre-processing approaches for each source, and collection of good practices and tips for applying ML to this kind of data with predictive purposes. Making use of a real data example, we illustrate some of the key issues to be addressed in this kind of research (technical noise, biological heterogeneity, class imbalance, high dimensionality, and presence of missing values, among others). This paper can be viewed as a sort of good practices and guidelines that could be extrapolated to other human diseases with a complex basis such as obesity. Although some topics have not been covered here given the nature of this work (conference paper), we also set the basis for future work incorporating some proposals to improve models, arguing their need according to encountered insights.

## References

1. Anguita-Ruiz, A.: Multi-omics integration and machine learning for the identification of molecular markers of insulin resistance in prepubertal and pubertal children with obesity (2021)
2. Barredo Arrieta, A., et al.: Explainable Artificial Intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **58**, 82–115 (2020). <https://doi.org/10.1016/J.INFUS.2019.12.012>
3. Browning, B.L., Tian, X., Zhou, Y., Browning, S.R.: Fast two-stage phasing of large-scale sequence data. *Am. J. Hum. Genetics* **108**(10), 1880–1890 (2021). <https://doi.org/10.1016/J.AJHG.2021.08.005>

4. Deelen, P., et al.: Genotype harmonizer: automatic strand alignment and format conversion for genotype data integration. *BMC. Res. Notes* **7**(1), 1–4 (2014). <https://doi.org/10.1186/1756-0500-7-901>
5. Fernández-Delgado, M., et al.: Do we need hundreds of classifiers to solve real world classification problems? *J. Mach. Learn. Res.* **15**, 3133–3181 (2014). <https://jmlr.org/papers/v15/delgado14a.html>
6. Fortin, J.P., et al.: Functional normalization of 450k methylation array data improves replication in large cancer studies. *Genome Biol.* **15**(12) (2014). <https://doi.org/10.1186/S13059-014-0503-2>
7. Goecks, J., et al.: How machine learning will transform biomedicine. *Cell* **181**(1), 92–101 (2020). <https://doi.org/10.1016/J.CELL.2020.03.022>
8. Goodarzi, M.O.: Genetics of obesity: what genetic association studies have taught us about the biology of obesity and its complications. *Lancet Diabetes Endocrinol.* **6**(3), 223–236 (2018). [https://doi.org/10.1016/S2213-8587\(17\)30200-0](https://doi.org/10.1016/S2213-8587(17)30200-0)
9. Hvitfeldt, E.: themis: Extra Recipes Steps for Dealing with Unbalanced Data (2020) <https://CRAN.R-project.org/package=themis>, r package version 0.1.0
10. James, G., Witten, D., Hastie, T., Tibshirani, R.: *An Introduction to Statistical Learning - with Applications in R* (2013). <https://doi.org/10.1007/978-1-4614-7138-7>
11. Mahajan, A., et al.: Refining the accuracy of validated target identification through coding variant fine-mapping in type 2 diabetes article. *Nat. Genet.* **50**(4), 559–571 (2018). <https://doi.org/10.1038/s41588-018-0084-1>
12. Maksimovic, J., Phipson, B., Oshlack, A.: A cross-package Bioconductor workflow for analysing methylation array data. *F1000Research* **5** (2016). <https://doi.org/10.12688/F1000RESEARCH.8839.3>
13. Purcell, S., et al.: PLINK: a tool set for whole-genome association and population-based linkage analyses. *Am. J. Hum. Genet.* **81**(3), 559 (2007). <https://doi.org/10.1086/519795>
14. Rohart, F., Gautier, B., Singh, A., Le, C.: mixomics: an r package for 'omics feature selection and multiple data integration. *PLoS Comput. Biol.* **13**(11), e1005752 (2017). <https://doi.org/10.1371/journal.pcbi.1005752>
15. Saxena, R., et al.: Genome-wide association analysis identifies loci for type 2 diabetes and triglyceride levels. *Science* **316**(5829), 1331–1336 (2007). <https://doi.org/10.1126/science.1142358>
16. Scott, L.J., et al.: A genome-wide association study of type 2 diabetes in finns detects multiple susceptibility variants. *Science* **316**(5829), 1341–1345 (2007). <https://doi.org/10.1126/science.1142382>
17. Scott, R.A., et al.: An expanded genome-wide association study of type 2 diabetes in Europeans. *Diabetes* **66**(11), 2888–2902 (2017). <https://doi.org/10.2337/db16-1253>
18. Singh, A., et al.: DIABLO: an integrative approach for identifying key molecular drivers from multi-omics assays. *Bioinformatics* **35**(17), 3055–3062 (2019). <https://doi.org/10.1093/BIOINFORMATICS/BTY1054>
19. Sladek, R., et al.: A genome-wide association study identifies novel risk loci for type 2 diabetes. *Nature* **445**(7130), 881–885 (2007). <https://doi.org/10.1038/nature05616>
20. Stekhoven, D.J., Bühlmann, P.: MissForest-non-parametric missing value imputation for mixed-type data. *Bioinformatics* **28**(1), 112–118 (2012). <https://doi.org/10.1093/BIOINFORMATICS/BTR597>

21. Teschendorff, A.E., et al.: A beta-mixture quantile normalization method for correcting probe design bias in Illumina Infinium 450 k DNA methylation data. *Bioinformatics* **29**(2), 189–196 (2013). <https://doi.org/10.1093/BIOINFORMATICS/BTS680>
22. Van Buuren, S.: Multiple imputation of discrete and continuous data by fully conditional specification. *Stat. Methods Med. Res.* **16**(3), 219–242 (2007). <https://doi.org/10.1177/0962280206074463>
23. Zhao, W., et al.: Identification of new susceptibility loci for type 2 diabetes and shared etiological pathways with coronary heart disease. *Nat. Genet.* **49**(10), 1450–1457 (2017). <https://doi.org/10.1038/ng.3943>



# Feature Density as an Uncertainty Estimator Method in the Binary Classification Mammography Images Task for a Supervised Deep Learning Model

Ricardo Javier Fuentes-Fino<sup>1,2</sup>, Saúl Calderón-Ramírez<sup>2</sup>,  
Enrique Domínguez<sup>1,3</sup>, Ezequiel López-Rubio<sup>1,3</sup>,  
Marco A. Hernandez-Vasquez<sup>2</sup>, and Miguel A. Molina-Cabello<sup>1,3</sup>(✉)

<sup>1</sup> Department of Computer Languages and Computer Science,  
University of Malaga, Málaga, Spain  
{enriqued,ezeqlr,miguelangel}@lcc.uma.es

<sup>2</sup> Instituto Tecnológico de Costa Rica, Cartago, Costa Rica  
RicardoFino@estudiantec.cr, {sacalderon,marco.hernandez}@itcr.ac.cr

<sup>3</sup> Instituto de Investigación Biomédica de Málaga - IBIMA, Málaga, Spain

**Abstract.** Labeled medical datasets may include a limited number of observations for each class, while unlabeled datasets may include observations from patients with pathologies other than those observed in the labeled dataset. This negatively influences the performance of the prediction algorithms. Including out-of-distribution data in the unlabeled dataset can lead to varying degrees of performance degradation, or even improvement, by using a distance to measure how out-of-distribution a piece of data is. This work aims to propose an approach that allows estimating the predictive uncertainty of supervised algorithms, improving the behaviour when atypical samples are presented to the distribution of the dataset. In particular, we have used this approach to mammograms X-ray images applied to binary classification tasks. The proposal makes use of Feature Density, which consists of estimating the density of features from the calculation of a histogram. The obtained results report slight differences when different neural network architectures and uncertainty estimators are used.

**Keywords:** Feature Density · Mahalanobis distance · Jensen-Shannon distance · Uncertainty · Deep learning

## 1 Introduction

Machine Learning (ML) approaches are trying to be applied in the field of medicine as a tool to help in classification and diagnosis tasks of diseases like cancer and more recently COVID-19 by using medical images [1, 2]. Cancer is the first or second leading cause of premature death and breast cancer remains the leading cause of death in women worldwide, although it can also be diagnosed in



men [3]. In 2019, it was estimated that 268,600 new cases of invasive breast cancer were diagnosed among women and approximately 2,670 cases diagnosed in men [4]. To mitigate these numbers, it is necessary an early and accurate diagnosis. The analysis of imaging evaluation such as mammography or histopathological [5, 6] images may supply that diagnosis. Due to this, approaches like ML have been extensively studied to improve classification tasks and apply them to medical diagnosis.

In areas such as medicine, the main problem is the limited data set, its quality and the acquisition process, and it causes that not all approaches are suitable and not all methods provide optimal performance. ML algorithms usually face many problems in real-world deployment environments and several examples of this can be found [7–10]. According to [7] and [8] the labelled dataset can include a limited number of observations for each class, in the context of breast cancer, a more significant number of samples without cancer can be observed than with cancer, which can cause a tendency of the models to classify better (or recognize) the samples of the majority class, this is known as Data Imbalance. Also in [9] mentioned that the test dataset can include observations of patients with other types of pathologies than those observed in the training dataset, this is known as Out-Of-Distribution (OOD) data, and it can be potentially harmful to classifications models performance and cause a degradation in its accuracy. Another well-studied problem [10] is the mismatch distribution of the data. This usually happens when deploying the algorithms to a real-world environment. Training models with a specific dataset does not guarantee that testing the model in another setting (another hospital or clinic, usually called target dataset) will give the same performance results.

Experimental evidence shows that despite accuracy being harmed by the problems mentioned above and in [11] mentions that obtaining models that can generalize the characteristics of breast cancer is complicated since there is significant variability of anomalies which will always limit the efficiency of the algorithms, the ML techniques they remain an attractive approach for the detection, classification or segmentation of different types of anomalies. Hence, it is essential to continue their improvement and investigation.

In ML, uncertainty measures how reliable or accurate a model is in classifying the images in a test data set based on the supervised training that the model has performed. In this work, we evaluate feature density as a measure of uncertainty and compare this method with others proposed in state-of-the-art like Mahalanobis distances. To perform this investigation, we offer the following question: is it possible to obtain a statistically significant improvement between using Feature Histogram to improve the estimation of predictive uncertainty concerning other techniques that assume a Gaussian distribution of the data set?

## 2 State of the Art

In [12] they propose to combine two uncertainty measurements. The first one, based on subjective logic [13],  $u(p) : p \rightarrow \mathbb{R}$ , based on the information contained from the probabilistic predictions, while the second, a data closeness

measurement  $D_m(z) : z \rightarrow \mathbb{R}$  following a Mahalanobis approach [14] that measures the distance  $D_m$  of a sample to the training distribution cluster. They have observed that the Mahalanobis distance brings a complementary aspect, especially related to out-of-distribution cases [14]. For instance, when a classifier trained on breast images (ID) is fed with outliers from a flower dataset (OOD), the authors saw that the rejection criterion based on the Mahalanobis distance is quite effective. Despite the effectiveness of the combination, further research is required on automatic ways to find the optimal thresholds.

On the other hand, [15] their focus is on uncertainty estimation methods that are practical and straightforward to implement. Specifically, the Softmax and Monte Carlo Dropout (MCD) approaches were tested. The usage of a Softmax activation function in the output layer of a deep learning model can serve as a basic method for uncertainty estimation. The complete set of values for a Softmax output given an input  $x_j$  can also be used for uncertainty estimation. This is done by calculating the entropy over the corresponding output distribution  $p$  of Softmax. Softmax method alone can lead to poor representations of model uncertainty due to typical overconfidence in neural networks' predictions. The MCD approach aims at having more robust estimations while still being simple to implement [16], when compared to the usage of Softmax for uncertainty estimation. MCD is based on a Bayesian interpretation of the model's parameters. According to their results, an improvement with statistical significance was observed for SSDL models over supervised models.

To deal with data imbalance, [8] proposes to use the transfer learning approach. Multiple models were trained under different training configurations to evaluate the impact of SSDL on their Transfer learning (a simple Domain adaptation method) and loss function based class-imbalance correction were also tested. Deep learning models were first trained in a supervised manner with complete mammography datasets  $D_{s,INbreast}^l$  and  $D_{s,DDSM}^l$  in order to obtain source-trained models which were further fine-tuned on their target Costa Rican dataset in a Supervised manner, with limited amounts of labelled observations. In summary, models that were subject to do main adaptation from a source mammography dataset showed improved classification performance results in comparison to other experimental configurations tested there.

## 3 Methods

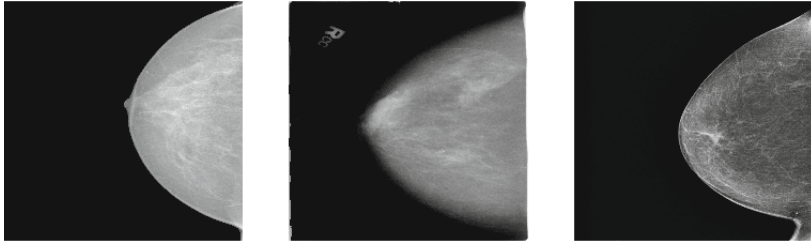
### 3.1 Mammography Datasets

Three different mammography datasets were used to carry out the experiments. The characteristics of those datasets are summarized in Table 1 and some samples of X-Ray images are illustrated in Fig. 1.

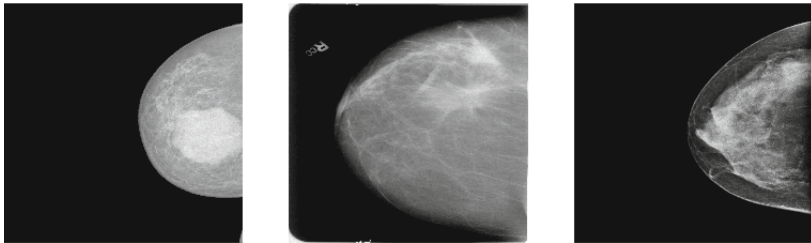
**INbreast.** The INbreast dataset introduced in [17] is a dataset containing a wide variety of breast anomalies such as masses, calcifications, architectural distortions, asymmetries and images with multiple anomalies at the same time, and usual patient samples. This dataset was built from 115 cases of X-ray images

**Table 1.** Summary of characteristics of the datasets.

	INbreast [17]	CBIS-DDSM [18]	CR-Chavarria 2020 [8]
Origin	Portugal	United States	Costa Rica
Year	2011	1997–2016	2020
Cases	115	1522	87
Images	410	3103	282



(a) Benign sample of INbreast (b) Benign sample of CBIS-DDSM (c) Benign sample of CR-Chavarria



(d) Malignant sample of INbreast (e) Malignant sample of CBIS-DDSM (f) Malignant sample of CR-Chavarria

**Fig. 1.** Mammogram samples from each dataset used according to a binary classification from a CC view (top-down view of the breast).

originating at Centro Hospitalar de São João at Porto, Portugal. Of the 115 cases, 90 cases have associated two images for each breast, belonging to each of the views (Craniocaudal (CC): which is a top to bottom view of the breast; and Mediolateral oblique (MLO): which is a side view of the breast); that is, 4 images associated with each patient; the remaining 25 cases only have related images for each of the views; giving a total of 410 X-ray images. The resolution of the images varies depending on the size of the patient’s breast. In addition, these images were evaluated and classified according to the categories of BI-RADS and according to their density measurement. For this case, the images were acquired digitally (Full-Field Digital Mammography) and stored in a DICOM (Digital Imaging and Communications in Medicine) format.

**CBIS-DDSM.** The Curated Breast Imaging Subset of Digital Database of Screening Mammography (CBIS-DDSM) [18] is an improved version of the Digital Database of Screening Mammography, which contained 2620 cases from different sources. This dataset has X-Ray images with standard samples, benign and malignant cases of breast cancer. The main problem with the original database was that some of the information attached to each case was limited or difficult to access. Due to this, a new dataset is created to improve the quality; to do this, inaccurate images or images that did not meet confidentiality standards are discarded. In [8] it is detailed that CBIS-DDSM contains a total of 3103 digitized images (scanned) belonging to 1566 cases, separated according to the anomaly presented in the X-Ray images (masses or calcifications) and classify according to the category of the BI-RADS system and according to its density measure. By classifying the dataset in a binary way, a total of 1728 images with benign cases were obtained and 1375 images with malignant cases.

**CR-Chavarria-2020.** Introduced in [8] the dataset from the Dr. Chavarria Estrada Medical Imaging private clinic located in Costa Rica. In [8] this dataset is used as out-of-distribution data as it comes to represent the conditions of a real-world deployment environment for the Machine Learning algorithms. The dataset was built from 87 cases, whose patients have an age range of 40 to 90 years. It contains 341 images, of which only 282 images are used, because in some cases the image does not have optimal quality or the patients have breast implants, which could produce noise in the classification models. When performing the classification in a binary way, the result is that 268 images are negative samples and 14 images are positive samples of cancer, showing a clear data imbalance in its classes. The images belonging to CR-Chavarria-2020 dataset were evaluated and classified according to the BI-RADS categories. Also, the images were acquired digitally form(FFDM).

### 3.2 Data Preprocessing

As part of the X-Ray image preprocessing from all three datasets described above, it was necessary to perform three operations on the datasets:

- A readjustment of the resolution of each image was performed, resulting in images of  $224 \times 224$  pixels, dimensions also used in the state-of-the-art literature in previous experiments, in order to reduce execution time, processing load and amount of disk space used.
- It was also necessary to change the file extension (image format) from DICOM to BMP (Windows Bitmap).
- This work was focused on the binary classification of the samples, because of this it was necessary a reclassification of the available datasets, similar to [8], where mammograms labelled with BI-RADS categories 4, 5 and 6 are defined as positive cases of breast cancer, while mammograms labelled with categories 1 and 2 are defined as negative cases of breast cancer. Image

samples labelled with categories 0 and 3 were discarded due to the peculiarity of their characteristics.

It was necessary to perform a second preprocessing stage on the dataset CBIS-DDSM since the X-ray images belonging to this set were digitized (scanned), thus their images were noisy. The anomalies observed are the following:

- In the pixels surrounding the breast it is observed as a blur (pixels in different shades of grey) similar to a shadow, which could cause the classification algorithms to take those areas as part of the image’s characteristics and cause a classification deficiency. To clean up noise, it was used the procedure described in [19].
- Despite the preprocessing that was given to the images described in the previous point, after a visual inspection it was found that in some images there were still remains of annotations of the type of view or data belonging to the X-ray, which could generate a bias within the classification model. To eliminate the remaining noise, it was necessary to make manual annotations of the area with noise and treat them using an algorithm.

After a second visual inspection of the images in the CBIS-DDSM dataset, it was possible to observe that in some exceptions the algorithm removed a considerable part of the breast. For these cases, manual cleaning of the image was carried out, similar to item two described above.

### 3.3 Training Process

For this work, the FastAI implementations of AlexNet and DenseNet architectures were chosen as classification models, were used a pre-trained version of the same and subsequently a Fine-Tuning process was performed on the dataset INbreast and CBIS-DDSM.

Initially, the configuration of hyperparameters used is the default configuration by the FastAI library, i.e. no modification was made to the algorithm to improve its accuracy when classifying images, with that a maximum of 70% accuracy was obtained on classification tasks, to improve that and achieve the accuracy reported in the state-of-art was resorted to using of Adam optimization function and data augmentation technique but was not obtain a statistical improvement.

Since the purpose of this work is not focused on obtaining models with the best possible accuracy in classification tasks, but to try uncertainty techniques, no further modifications were made to the classification models and left the default settings. To a certain extent, it is sought that the models are not perfect and that they make errors, in order to be able to evaluate the uncertainty estimators.

Initially, the models were trained from 857 X-ray images as shown in Table 2 for a maximum of 50 epochs. The selection of these images was done randomly. In order to improve the accuracy of the models, it was also experimented the

**Table 2.** Composition of images from the training dataset

Dataset	Number of images	Class balance
INbreast BI-RADS-1	47	242
INbreast BI-RADS-2	195	
INbreast BI-RADS-4	34	
INbreast BI-RADS-5	39	
INbreast BI-RADS-6	4	
CBIS-DDSM Benign Calcifications	140	
CBIS-DDSM Benign Masses	189	
CBIS-DDSM Malignant Calcifications	92	209
CBIS-DDSM Malignant Masses	117	

training of the models with more epochs (e.g. 200 epochs) and tried to use a more balanced training set, but it did not obtain an improvement of the performance.

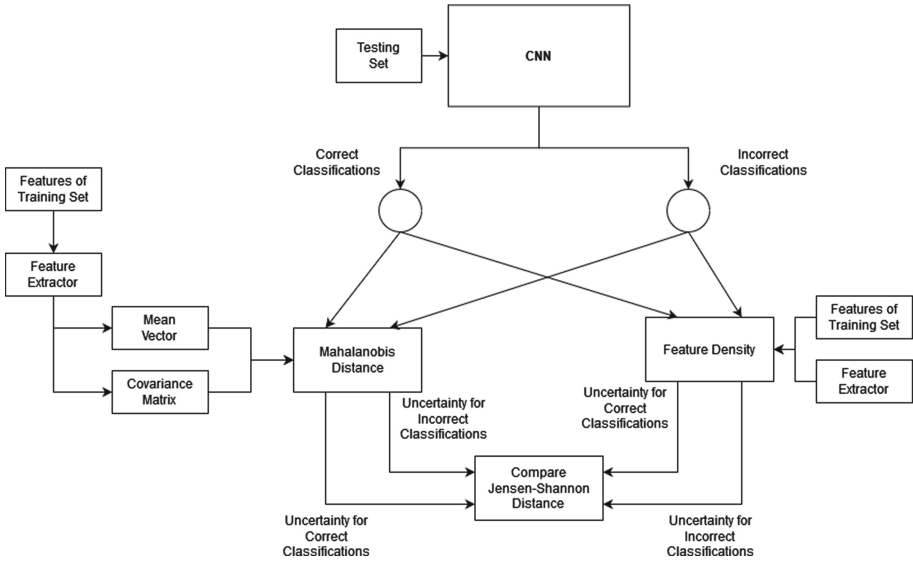
From the training process, the feature extractor was obtained, which in simple words are all those operations or mathematical processes that the network has used to extract the features of images. The feature extractor is used as part of the uncertainty estimators. The aim is to obtain the features of the correct and incorrect estimations and compare them with the features of the training images.

### 3.4 Uncertainty Estimation Process

Once the training of the models is finished, the uncertainty estimators were evaluated. For this, 10 test sets were used. Once the network has classified the test images, the confusion matrix and the network's predictions were used to find out the number of correct and incorrect estimations. From this information, representative subsets were created, these sets (correct and incorrect estimations) were subsequently processed by the uncertainty estimator models, together with the other necessary parameters. (similar to data flow shown in Fig. 2).

For the Mahalanobis Distance method, it was necessary to calculate the covariance matrix and the vector of means, from the training dataset, these elements are the basis that was used to estimate the uncertainty of the previously built image sets. For each image within the subsets mentioned above, an uncertainty measurement was obtained, thus creating two vectors of uncertainty, i.e. a vector with uncertainties of correct estimations and the other with uncertainties of incorrect estimations. Once this information was obtained, a PDF (Probability Density Function) was created for each of the uncertainty vectors, and it proceeded to calculate the distance between them (Jensen-Shannon Distance). The distance will be compared subsequently with the other estimator method.

For the Feature Density method, it was first necessary to estimate the feature histogram of the training dataset, this histogram is the basis for estimating



**Fig. 2.** Schema of the estimation of uncertainty

the uncertainty of the previously constructed image subsets. As in the previous method, for each subset (correct and incorrect estimations) a vector was obtained that contains each one of the uncertainty measurements corresponding to each image. Again, another PDF was created for each of the uncertainty vectors and the distance between them was calculated.

Once the Jensen-Shannon distance of the uncertainty vectors has been measured using each of the methods, a direct comparison was made as to which method is more accurate. As mentioned above, the Jensen-Shannon distance of the uncertainty distribution is intended to be as large as possible.

## 4 Experiment Results

To evaluate the performance of the uncertainty estimator models, 10 experiments (batches) were used, each of the test sets had 60 randomly selected X-ray images, covering each of the types of images available. It is important to mention that the network had never seen the images of test sets previously. In the first five experiments were used in-of-distribution images, i.e. images that belonged to the INbreast and CBIS-DDSM datasets with which the network was trained. In the remaining five experiments, different degrees of out-of-distribution data contamination were used, as shown in Table 3, belonging to the CR-Chavarria-2020 dataset.

The first experimental stage it was necessary to train the AlexNet architecture with the INbreast and CBIS-DDSM dataset with the number of images detailed in Table 2, 20% of the total images were used as a validation set. The

**Table 3.** Evaluation experiments for the uncertainty estimation methods

Experiments without contamination			Experiments with contamination		
N° of exp.	Number of images	Distribution percentage	N° of exp.	Number of images	Distribution percentage
1	60	100% IOD	6	60	75% IOD 25% OOD
2	60	100% IOD	7	60	50% IOD 50% OOD
3	60	100% IOD	8	60	50% IOD 50% OOD
4	60	100% IOD	9	60	25% IOD 75% OOD
5	60	100% IOD	10	60	100% OOD

**Table 4.** Number of correctly and incorrectly classified images, using INbreast and CBIS-DDSM as IOD data and CR-Chavarria as OOD data, with an Alexnet architecture for classification.

Experiments without contamination				Experiments with contamination			
N° of exp.	Correct. estimations	Incorrect. estimations	Acc	N° of exp.	Correct. estimations	Incorrect. estimations	Acc
1	33	27	0,5500	6	31	28	0,5254
2	31	29	0,5167	7	31	29	0,5167
3	32	28	0,5333	8	33	27	0,5500
4	33	27	0,5500	9	40	20	0,6667
5	28	32	0,4647	10	45	15	0,7500

neuronal network was trained for 50 epochs. The maximum accuracy obtained in the train validation was 70%.

Despite not obtaining high accuracy in the classification tasks, it was not taken as an impediment to continue with the experiments, since a perfect classification model was not sought. Table 4 shows the number of correct and incorrect estimations made by the neural network over the test dataset, as well as the accuracy with which it was made.

Not in all experiments can the capacity of the neural network to classify OOD data be determined with such precision, although experiment 10 of Table 4 can be taken as a basis, where there is 100% of OOD data and the model adequately classified 75% of the samples. In Tables 5 and 6 the averages of the uncertainty measurements were compiled for the ten experiments carried out in this stage.

Despite being hardly noticeable, when analyzing the averages of the uncertainty values, there are two tendencies:



**Table 5.** Average of uncertainty measurements over the correct and incorrect estimations, using INbreast and CBIS-DDSM as IOD data.

N° of exp.	Mahalanobis distance		FD method	
	Correct. estimations	Incorrect. estimations	Correct. estimations	Incorrect. estimations
1	9,7627	7,6000	388,2441	386,3513
2	7,9117	6,3012	384,3943	394,4933
3	8,9966	7,2569	336,9922	381,4414
4	7,6128	8,4158	385,8245	395,2873
5	9,4562	7,2151	385,8129	394,8537

**Table 6.** Average of uncertainty measurements over the correct and incorrect estimations, using INbreast and CBIS-DDSM as IOD data and CR-Chavarria as OOD data.

N° of exp.	Mahalanobis distance		FD method	
	Correct. estimations	Incorrect. estimations	Correct. estimations	Incorrect. estimations
6	9,0266	5,7338	416,1937	423,9887
7	8,6063	8,1578	491,3473	465,5490
8	8,0021	7,0746	459,0135	478,2890
9	9,2823	6,6258	520,9386	505,8273
10	11,5599	6,6558	548,6212	554,1428

- The difference between the uncertainty measurements for the correct and incorrect estimations is minimal in the case of the Mahalanobis Distance, whereas with the Feature Density method the uncertainty measurements for the incorrect estimations are a little greater than the uncertainty measurements for the correct estimations.
- The uncertainty measurements for the experiments with OOD data are a little greater than the uncertainty measurements for the experiments without OOD data, the most noticeable difference could be seen with the Feature Density method.

The observations above are not always met, especially using the Mahalanobis Distance method. Thus, it is necessary more experiments to determine the causes. All information about the comparison between both methods are showed in Table 7.

One aspect in which there is a big difference between both estimating methods is in the execution time and computational cost. With a convolutional layer belonging to the AlexNet architecture, the Mahalanobis Distance method takes an average of 0.3 ms to process an experimental batch, while with the Feature

**Table 7.** Jensen-Shannon distance between the uncertainties of correct and incorrect estimations, using INbreast and CBIS-DDSM as IOD data and CR-Chavarria as OOD data. Classification architecture: AlexNet.

Experiments without contamination			Experiments with contamination		
N° of exp.	JS distance with the Mahalanobis method	JS distance with the FD method	N° of exp	JS Distance with the Mahalanobis method	JS distance with the FD method
1	0,3639	0,3579	6	0,3865	0,3011
2	0,3883	0,3409	7	0,3639	0,3480
3	0,3573	0,3158	8	0,3469	0,4000
4	0,4419	0,3069	9	0,3666	0,3079
5	0,2932	0,4647	10	0,3896	0,5324
Avg	0,3689	0,3573		0,3707	0,3779
Std	0,0481	0,0566		0,0157	0,0849

Density method it takes an average of 41 s. The big difference between the execution times is due to the calculation of the Feature Histogram for each one of the dimensions of the training set when it is processed by the Feature Extractor. To calculate the execution time using the Mahalanobis Distance method, the computation time of the covariance matrix and the vector of means plus the batch processing time are added. In the case of Feature Density, the time it takes to calculate the Feature Histogram of the training set is added plus the batch processing time.

As a second experimental stage, a DenseNet architecture was used, the process of both training, validation and testing was similar to that used with the AlexNet architecture.

The results obtained for the Jensen-Shannon distance are shown in Table 8. As can be seen when using a feature extractor belonging to the DenseNet network, there is a more notable difference between both estimating methods; In this case, the Feature Density method is the one with the highest value for both the IOD and the OOD samples. This would indicate that the performance of the method is related to the type of Feature Extractor that is used.

When using a more complex Feature Extractor, the execution time and the computational cost increased significantly for both methods. For the Mahalanobis method the average time in the execution of the experiments was 3.6047s, while for the Feature Density estimator it was 1763.3704s (approximately 30 min), this difference between the times is due to the fact that with the Feature Extractor produced from the DenseNet architecture, 1024 dimensions are obtained as a result, at which The Feature Histogram must be calculated from the training data set. Therefore, the little gain obtained by estimating the uncertainty is overshadowed by the execution time invested.

**Table 8.** Jensen-Shannon distance between the uncertainties of the correct and incorrect estimations, using INbreast and CBIS-DDSM as IOD data and CR-Chavarria as OOD data. Classification architecture: DenseNet.

Experiments without contamination			Experiments with contamination		
N° of exp.	JS distance with the Mahalanobis method	JS distance with the FD method	N° de exp.	JS distance with the Mahalanobis method	JS distance with the FD method
1	0,2934	0,3479	6	0,1076	0,4151
2	0,2722	0,4098	7	0,3647	0,3779
3	0,2234	0,3988	8	0,3710	0,4193
4	0,3476	0,5553	9	0,4280	0,4163
5	0,3105	0,5180	10	0,3798	0,4209
Avg	0,2894	0,4460		0,3296	0,4099
Std	0,0412	0,0778		0,1135	0,0161

## 5 Conclusions and Recommendations

This research was carried out to evaluate the feature density method as an uncertainty estimator, applied to the binary classification of X-ray images (mammograms), using the AlexNet and DenseNet neural network architectures.

Based on the results of this work, no statistically significant improvement was found between the feature density method concerning the Mahalanobis Distance as an uncertainty estimator method when using an AlexNet architecture. In the case of the DenseNet architecture, a more notable difference can be observed, but the results are not entirely conclusive. This way, more experiments are needed to reach a more accurate answer.

If the execution time and the computational cost invested in estimating the uncertainty using both methods are taken into consideration, it can even be thought that the Mahalanobis Distance has some advantage from that perspective. It is necessary to emphasize that the execution time and computational cost is closely related to the type of architecture selected for the experiments.

Despite the conclusions reached in this research, this does not mean that the feature density method should be discarded entirely as an estimator of uncertainty. Like everything in Artificial Intelligence, more experiments must be carried out to reach an accurate conclusion about which method has a better performance.

As recommendations to continue with the work raised in this research, it proposes:

- Perform more experiments, with a more significant number of images for both training and testing. As there are few images and tests, no conclusive trend regarding improvement can be observed. Another recommendation is to experiment with data augmentation approaches and find the optimal combination of transformations on the images.

- Use other convolutional network architectures to investigate if there are architectures (and thus their feature extractor) where the performance of the feature density method might be better.
- Experiment with the hyperparameters of the architectures until finding an optimal configuration, which can reach the accuracy proposed in [15] and experiment if there is a variation in the estimation of the uncertainty.
- Experiment with other datasets of medical images, with the possibility that in different contexts, a significant improvement is obtained, since not necessarily when getting a low or high performance in a specific context means that it must work in the same way in others.

**Acknowledgments.** This work is partially supported by the following Spanish grants RTI2018-094645-B-I00 and UMA18-FEDERJA-084. All of them include funds from the European Regional Development Fund (ERDF). The authors acknowledge the computer resources, technical expertise and assistance provided by the SCBI (Supercomputing and Bioinformatics) center of the University of Malaga. The authors acknowledge the funding from the Instituto de Investigación Biomédica de Málaga - IBIMA and the Universidad de Málaga.

## References

1. Calderon-Ramirez, S., et al.: Improving uncertainty estimation with semi-supervised deep learning for Covid-19 detection using chest x-ray images. *IEEE Access* **9**, 85, 442–485 (2021)
2. Calderon-Ramirez, S., et al.: Dealing with scarce labelled data: semi-supervised deep learning with mix match for Covid-19 detection using chest x-ray images. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 5294–5301. *IEEE* (2021)
3. Wild, C., Weiderpass, E., Stewart, B.: World cancer report: cancer research for cancer prevention. International Agency for Research on Cancer, Lyon, France (2020)
4. A. C. Society: Breast Cancer Facts & Figures 2019–2020. American Cancer Society, Atlanta (2019)
5. Molina-Cabello, M.A., Accino, C., López-Rubio, E., Thurnhofer-Hemsi, K.: Optimization of convolutional neural network ensemble classifiers by genetic algorithms. In: Rojas, I., Joya, G., Catala, A. (eds.) IWANN 2019. LNCS, vol. 11507, pp. 163–173. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-20518-8\\_14](https://doi.org/10.1007/978-3-030-20518-8_14)
6. Molina-Cabello, M.A., Rodríguez-Rodríguez, J.A., Thurnhofer-Hemsi, K., López-Rubio, E.: Histopathological image analysis for breast cancer diagnosis by ensembles of convolutional neural networks and genetic algorithms. In: 2021 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. *IEEE* (2021)
7. Calderon-Ramirez, S., et al.: Correcting data imbalance for semi-supervised Covid-19 detection using x-ray chest images. *Appl. Soft Comput.* **111**, 107692 (2021)
8. Ramírez, S.C., Murillo-Hernández, D., Rojas-Salazar, K., Elizondo, D., Moemeni, A., Molina-Cabello, M.A.: A real use case of semi-supervised learning for mammogram classification in a local clinic of Costa Rica. *Med. Biol. Eng. Comput.* (2022)

9. Calderon-Ramirez, S., et al.: Mixmood: a systematic approach to class distribution mismatch in semi-supervised learning using deep dataset dissimilarity measures. arXiv preprint [arXiv:2006.07767](https://arxiv.org/abs/2006.07767) (2020)
10. Calderon-Ramirez, S., Yang, S., Elizondo, D., Moemeni, A.: Dealing with distribution mismatch in semi-supervised deep learning for Covid-19 detection using chest x-ray images: a novel approach using feature densities. arXiv preprint [arXiv:2109.00889](https://arxiv.org/abs/2109.00889) (2021)
11. Sun, W., Tseng, B., Zhang, J., Qian, W.: Enhancing deep convolutional neural network scheme for breast cancer diagnosis with unlabeled data. *Comput. Med. Imaging Graph.* (2016)
12. Tardy, M., Scheffer, B., Mateus, D.: Uncertainty Measurements for the Reliable Classification of Mammograms. In: Shen, D., Liu, T., Peters, T.M., Staib, L.H., Essert, C., Zhou, S., Yap, P.-T., Khan, A. (eds.) MICCAI 2019. LNCS, vol. 11769, pp. 495–503. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-32226-7\\_55](https://doi.org/10.1007/978-3-030-32226-7_55)
13. Jøsang, A.: Subjective Logic: A Formalism for Reasoning Under Uncertainty. International Series of Monographs on Physics. Springer, Cham (2016). <https://doi.org/10.1007/978-3-319-42337-1>
14. Denouden, T., Salay, R., Czarnecki, K., Abdelzad, V., Phan, B., Vernekar, S.: Improving reconstruction autoencoder out-of-distribution detection with Mahalanobis distance. *CoRR*, vol. abs/1812.02765 (2018). <http://arxiv.org/abs/1812.02765>
15. Calderón-Ramírez, S., et al: Improving uncertainty estimations for mammogram classification using semi-supervised learning. In: 2021 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2021)
16. Gal, Y., Ghahramani, Z.: Dropout as a Bayesian approximation: representing model uncertainty in deep learning (2016)
17. Moreira, I.C., Amaral, I., Domingues, I., Cardoso, A., Cardoso, M.J., Cardoso, J.S.: INbreast: toward a full-field digital mammographic database. *Acad. Radiol.* **19**(2), 236–248 (2012)
18. Lee, R.S., Gimenez, F., Hoogi, A., Miyake, K.K., Gorovoy, M., Rubin, D.: A curated mammography data set for use in computer-aided detection and diagnosis research. *Sci. Data* **4** (2017)
19. Beeravolu, A.R., Azam, S., Jonkman, M., Shanmugam, B., Kannoopatti, K., Anwar, A.: Preprocessing of breast cancer images to create datasets for deep-CNN. *IEEE Access* **9**, 438–463 (2021)



# Iterative Clustering for Differential Gene Expression Analysis

Olga Georgieva<sup>(✉)</sup>

Sofia University “St. Kliment Ohridski”, Sofia, Bulgaria

[o.georgieva@fmi.uni-sofia.bg](mailto:o.georgieva@fmi.uni-sofia.bg)

**Abstract.** The Next Generation Sequencing technologies provide large volumes of DNA-seq and RNA-seq data. A central part of their investigation is the task for selecting the differentially expressed genes. Different methods for RNA-seq data analysis that identify genes distinguished by their expression levels have been proposed basically on the statistical data analysis. There is no agreement among the applied methods as different results are produced by the distinct methods. The present paper proposes a new method for differential gene expression analysis based on machine learning approach. Difficulty of the selection due to the large number of indistinguishable genes is solved by iterative clustering procedure. The importance of the proper cluster distance measure is discussed. The visibility of the procedure results and ability to find different number of compact clusters is also underlined. The significance of the method is investigated and proved by application to the two mice strains dataset. The obtained results are compared with the results of the statistical methods applied to the same dataset. It is concluded that the proposed method is valuable and could be applied as standalone or for preliminary genes selection within a statistical algorithms pipeline for discovering differentially expressed genes.

**Keywords:** RNA-seq · Differential gene expression analysis · Cluster analysis

## 1 Introduction

The Next Generation Sequencing technologies provide large volumes of DNA-seq and RNA-seq data. These are gene expression data, which are more precise having higher resolution than the older technologies such as microarray. It significantly increases the opportunities for effective research and revealing knowledge about dependencies of the genes' activity. A central part of these investigations is the task for selecting differentially expressed genes. In this way, the genes responsible for certain disease states can be discovered or it is possible to find differences of two species strains.

Different methods for RNA-seq data analysis to identify genes distinguished by their expression levels have been proposed. These are basically statistical data analysis approaches. Systematic and deep comparison studies is provided to quantifying the proximity of methods' performance in solving the task [1–3]. The papers' conclusions state that there is no agreement among the applied methods and the respective software

used tools as different results are produced by the distinct methods. For instance, fifteen methods were assessed according to the obtained  $p$ -values of each gene [3] of a dataset consisting of two mice groups of different strains. The differentially expressed genes of the two strains have been selected according to a value of a predefined threshold using an appropriate software [4]. Only 570 genes in common were recognized as significant by four different methods namely *DESeq* [5], *DESeq2* [6], *edgeR* [7] and the *limma* method that is closely to *ttest*. Each of these methods has been chosen as a representative of a subgroup of similar methods for investigation. It is underlined that beside the common genes identified from all methods each method detects additional genes that are not found by the others. The amounts of these additional genes differ in large among the distinct methods.

The difficulties of the existing methods in reliable discovering differently express genes is an incentive to explore other approaches in solving the problem. Given the lack of reference information to assess the selected genes, the unsupervised approaches of machine learning analysis could be useful. The machine learning methods able to find structures within complex data sets as biosignal data are. By that they could increase the knowledge about the biological mechanisms and their application can surely improve patient outcomes [8, 9]. The obtained results indicate that machine learning algorithms can effectively differentiate healthy subjects and affected patients [10]. The successful implementation of these methods to tasks of differential expression analysis [11, 12] encourages such research.

The present paper proposes a new method for differential gene expression identification based on clustering analysis. The difficulty of the selection problem due to the large number of indistinguishable genes is solved by iterative clustering procedure. The significance of the approach is investigated by application to the two mice strains dataset. The obtained results are compared with the results of the well-known statistical methods applied to the same dataset. According to the comparison it is concluded that the approach is valuable and could be apply as stand alone or for preliminary gene selection within a pipeline of the algorithms for differential gene expression.

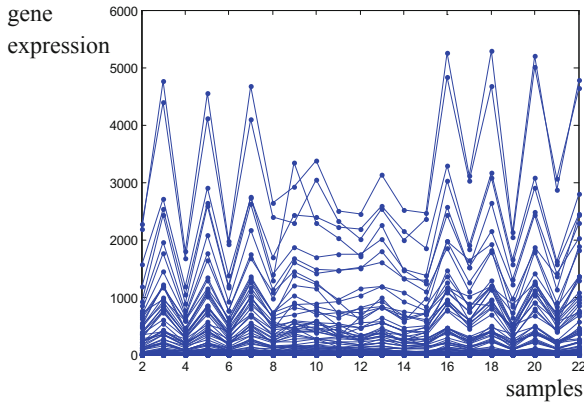
## 2 Problems of Differential Gene Expression Analysis

Differential expression problem tries to find genes that have significantly different activity represented by the expression levels of one strain compared to other. For this task RNA-seq dataset is used. Each entry of the dataset is a row with reads of a particular gene for samples given in the table columns.

The solution of the task is complicated due to difficulties in gene expression distinguishing assessed by statistical analysis. Due to the large number of genes and lack of significance in separation by the estimated  $p$ -values no valuable conclusion about the gene selection could be done. The task is complicated by the fact that usually number of interested genes is quite smaller than the whole their number. In this case the searched genes are appeared mostly as outliers than as a representative group that could be estimated and separated.

Other difficulty of the differential gene expression separation is a result of the large deviation of the activity levels among the samples of a certain strain. For instance, the

profiles of the gene expression values of mice of two different strains [13] shows larger deviation in the gene activities within the same strain than between the two strains (Fig. 1). Due to the impossibility of distinguishing genes within the distinct samples, a solution based on some aggregated measure of the gene activity of all samples could be useful.



**Fig. 1.** Gene expression profiles of 100 genes that are most differently in their average activity values of the two mice strains. First ten samples (from 2 to 10) are of one mice strain and the rest 11 (from 11 to 22) are of another mice strain

### 3 Unsupervised Analysis for Gene Expression Differentiation

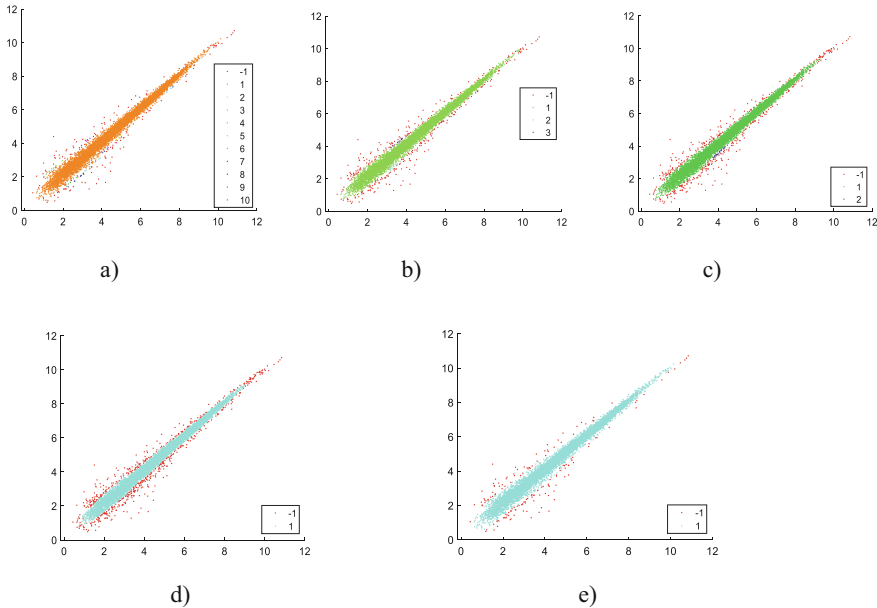
Here we propose a method of gene expression differentiation that uses the average value of the gene expression of the strain samples. Comparing the average values of the two investigated strains we could expect that for the genes that behave equally the respective average values remain close whereas for the differently expressed genes the mean values differ. Direct comparison will not give a reliable result as we do not have a threshold value to separate the genes group with equal behavior from the group with dissimilar behavior. It is necessary to apply a procedure that could distinguish the genes data that is not separable from those that are much different for the two strains. The latter are differently expressed genes which we are interested in. Due to the lack of a reference threshold the study could use an unsupervised machine learning method as cluster analysis. This machine learning technique does not need preliminary information about the data structure.

In accordance with this idea we are looking for a clustering algorithm applied to the data space formed by the average gene activity values of two strains that enables us to distinguish the two types of genes. An appropriate choice is the widely applicable DBSCAN algorithm [14], which groups data based on a density-based approach and finds clusters, as well as outliers. By this algorithm we can separate genes densely clustered around the equivalence area from the outliers that are away from this area. In fact, the outlier data are of genes that are differentially expressed and that we are searching for.



DBSCAN clustering uses a matrix of pairwise distances between data. It finds the number of outliers and core points. The clustering is accomplished based on a threshold radius  $r$  for neighborhood search and a minimum number of neighbors  $N_{min}$  required to identify a core point. The two parameters are subject of off-line investigation and fully depend on the structure of the data space. The default measure for data range estimation is Euclidian distance.

Clustering results of Bottomly reduced dataset [13] are presented at Fig. 2 where the two dimensions correspond to the two mice strains and dots' coordinates are the respective average expression values of a certain gene of the two strains. By varying the clustering parameters  $r$  and  $N_{min}$  different number of clusters is discovered. It should be underlined that we are interested not in the data of the clusters but the outliers that are surrounding the compact data group(s). They are marked by “-1” and are the data of differently expressed genes. For small cluster radius  $r$  and low minimum neighbors number  $N_{min}$  the number of discovered clusters is relatively high and the number of the outliers is small (Fig. 2a,b).



**Fig. 2.** a) Results of DBSCAN clustering applied to the average values of the genes activity of the two mice strains a)  $r = 0,2$ ,  $N_{min} = 5$ , discovers 153 differently expressed genes; b)  $r = 0,2$ ,  $N_{min} = 10$ , discovers 310 genes; c)  $r = 0,2$ ,  $N_{min} = 15$ , discovers 420 genes; d)  $r = 0,2$ ,  $N_{min} = 20$ , discovers 593 genes; e)  $r = 0,4$ ,  $N_{min} = 20$ , discovers 154 genes. Discovered clusters are enumerated in the picture legends.

By increasing  $N_{min}$  more outliers are identified (Fig. 2c,d). The radius  $r$  is sensitive parameter as by its doubling the number of identified differently expressed genes decreases (Fig. 2d,e). For some parameters values only one compact cluster is defined (Fig. 2d,e).

It should be underlined a problem of DBSCAN clustering applied to the entire genes dataset. Due to the large number of genes behaving equivalent the algorithm reveals limited amount of differently expressed genes compared to those discovered by statistical analysis [3]. In order to improve the distinguishability of the two types of genes and thus their disclosure, the clustering can be applied not to the whole dataset at once, but sequentially to distinct subsets of all data. These subsets of data must have the same volume and here we call them data batches.

## 4 Iterative Procedure of Gene Expression Differentiation

Aiming to improve gene selection, here we propose an iterative clustering procedure that applies DBSCAN at batches of genes. By that the outliers of each batch are added in order to form a common set of the differently expressed genes.

### 4.1 Data Preprocessing

A lot of zero values are a main feature of the RNA-seq data. This implies special attention in the developed methods for analysis of differential gene expression. In our case the problem is solved during the preprocessing stage as a part of the procedure pipeline.

Two important stages of data preprocessing are need. First, logarithm transformation of the data is obligatory to deal with the large differences in the expression values of the raw data that will distort the gene grouping. Second, in order to ensure the logarithm calculation, filtering for removing genes with zero activity value is a requisite.

### 4.2 Data Processing

Iterative implementation of DBSCAN requires to set clustering parameters in advance. These are the threshold radius  $r$  for neighborhood search, minimum number of neighbors  $Nmin$  and the number of genes' data in a batch. Once the parameters are found they are applied to each data batch.

Again, due to the similarity in the genes' behavior as in case of the whole data set (Fig. 2), it could be expected that the data of each batch form a large compact group along the equivalence area. This area is rather oblong than spherical one. This suggests that clusters are not Euclidean. This observation is assessed by exploiting different distance measure of DBSCAN search. In searching of a proper distance measure two distances—Euclidean and Mahalanobis, were applied and the results were compared in terms of data separation abilities.

The number of data that form a batch is also a subject of an advance choice. The larger number could make impossible to detect the outliers due the effect already discussed in case of whole data set processing. The low number could embarrass detection of real clusters and thus the right selection of the interested genes.

The search for proper radius  $r$  and neighbors  $Nmin$  could be such to maximize the number of the discovered differentially expressed genes.

## 5 Application Results and Discussion

The proposed procedure for finding genes that are differently expressed is applied for the samples' set of two mice strains—ten of strain C57BL/6J and eleven of strain DBA/2J. Raw data available from the ReCount online resource [15] were filtered to represent data of 13932 genes having non-all-zero rows in the dataset [13].

By applying the preprocessing transformations discussed in Sect. 4.1, the number of genes is reduced to 9196. They were divided in 18 batches of 511 genes each except the last one consisting 508 genes. By that the amount of a batch is set to be comparable to the number of genes found by statistical analysis. DBSCAN algorithm was set by parameters given in Table 1.

**Table 1.** DBSCAN initialization parameters

Distance	<i>r</i>	<i>Nmin</i>
Euclidian	0.15	10
Mahalanobis	0.2	5

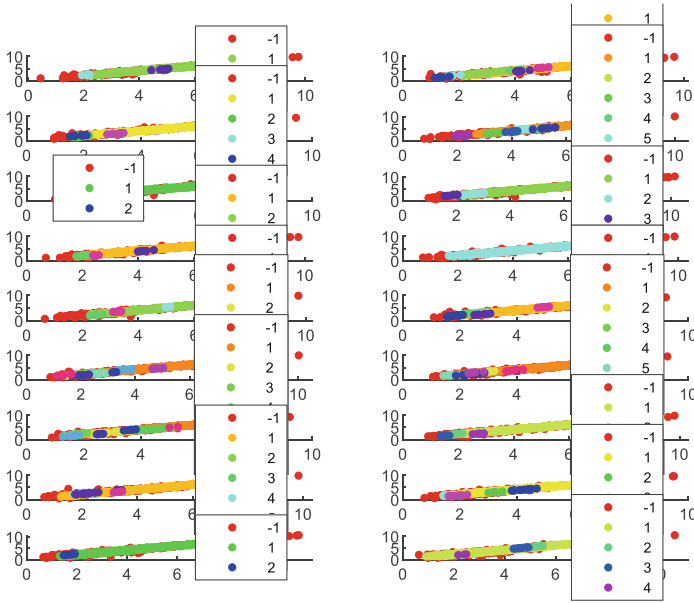
For the purposes of analyzing the results and comparing the proposed method with the existing ones the found group of differently expressed genes marked by *ML* is compared with genes' groups defined through statistical data analysis (Table 2). Four statistical methods—*ttest*, *edgeR*, *limma*, *DESeq2* have been explored to the same dataset. The number of discovered genes by each statistical method of the filtered dataset provided by [3] is presented at the first (sub)column of the respective method column. The number of genes discovered by our procedure that are common for the corresponding statistical method is given at the respective second (sub)column. The last column of the table “*ML* all data” consists the total amount of differently expressed genes that are identified by the proposed iterative clustering method.

**Table 2.** Number of differentially expressed genes selected by the different methods

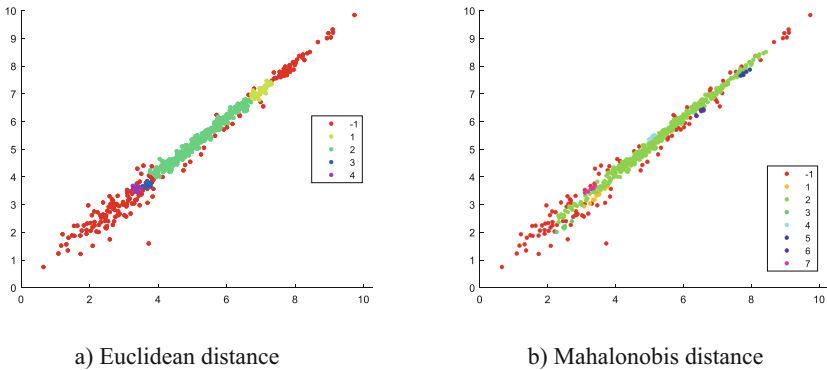
Method	<i>ttest</i>		<i>edgeR</i>		<i>limma</i>		<i>DESeq2</i>		<i>ML</i> all data
	<i>ttest</i>	<i>ML</i>	<i>edgeR</i>	<i>ML</i>	<i>limma</i>	<i>ML</i>	<i>DESeq2</i>	<i>ML</i>	
Distance									
Euclidean	71	71	915	647	736	537	982	648	2848
Ma- halanobis	71	71	915	738	736	611	982	735	1905

It could be seen that the preference is given for searching by Mahalanobis distance. By searching oblong clusters through Mahalanobis distance smaller number of 1905 is selected against 2848 through Euclidean distance. By that, by Mahalanobis distance



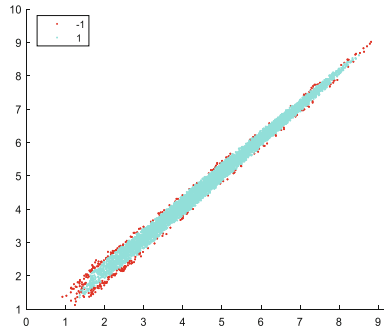


**Fig. 4.** Results of Iterative DBSCAN clustering by Mahalanobis distance measure



**Fig. 5.** Clustering results of the 9-th batch

The proposed procedure can be applied recursively. For this aim new dataset of 7 291 genes is obtained after subtraction the selected differentially expressed genes from the initial dataset. They are clustered by DBSCAN algorithm and in result additional 317 differentially expressed genes are identified (Fig. 6).



**Fig. 6.** Result of DBSCAN clustering,  $r = 0.2$ ,  $Nmin = 20$  of newly form dataset

Some additional observations have to be marked. By varying the clustering parameters different splitting in two groups could be found—the group of outliers (differently expressed) and rest genes equivalent in its behavior. The clustering determines several clusters along the equivalent area (Figs. 3 and 4), which captures genes with specific (very close) behavior. For instance, in case of clustering of 9-th batch (Fig. 5b) clusters except 2-nd should be interpreted additionally as they show close behavior but, in some sense, different from the large compact group. All these observations are prerequisites for further knowledge extraction and an opportunity to find genes with different level of significance revealing new useful information.

## 6 Conclusion

The proposed method introduces a new procedure for gene differential expression identification that is applicable to RNA-seq dataset. It is build based on machine learning approach that implements iterative clustering on data space defined by the averaged sample data of two strains.

It is shown that by varying the values of the clustering parameters the procedure discovers different number of significant genes. The important improvement is found by proper choice of the cluster distance measure. It is underlined that oblong clustering finds more genes common for the proposed method and for the statistical analysis for genes expression identification. The visibility of the procedure results and ability to find different number of compact clusters is other method advantage. This is an object of additional information, which could gain new knowledge about the gene activity.

The procedure could be used in combination with statistical methods in order to stick their search in a smaller number of genes. However, it is applicable as a standalone method by exploring some further procedure improvements as defining proper number of genes in a batch, optimal clustering parameters values and procedure optimization.




**Acknowledgement.** The result presented in this paper is part of the GATE project. The project has received funding from the European Union’s Horizon 2020 WIDESPREAD-2018–2020 TEAMING Phase 2 programme under Grant Agreement No. 857155 and Operational Programme Science and Education for Smart Growth under Grant Agreement No. BG05M2OP001–1.003–0002–C01.

## References

1. Spies, D., Renz, P.F., Beyer, T.A., Ciaudo, C.: Comparative analysis of differential gene expression tools for RNA sequencing time course data. *Brief. Bioinform.* **20**(1), 288–298 (2019)
2. Wang, T., Li, B., Nelson, C.E., et al.: Comparative analysis of differential gene expression analysis tools for single-cell RNA sequencing data. *BMC Bioinform.* **20**, 40 (2019)
3. Palejev, D.: Comparison of RNA-seq differential expression methods. *Cybern. Inf. Technol.* **17**(5), 60–67 (2017)
4. Law, C.W., Chen, Y., Shi, W., Smyth, G.: Voom: Precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biol.* **15**, 1–17 (2014). R29
5. Anders, S., Huber, W.: Differential expression analysis for sequence count data. *Genome Biol.* **11** (2010). R106
6. Love, M.I., Huber, W., Anders, S.: Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.* **12** (2014). 550
7. Robinson, M.D., Mccarthy, D.J., Smyth, G.K.: EdgeR: A bioconductor package for differential expression analysis of digital gene expression data. *Bioinform.* **26**(1), 139–140 (2010)
8. Chousiadass, D., Menychtas, A., Tsanakas, P., Maglogiannis, I.: Advancing quantified-self applications utilizing visual data analytics and the internet of things. In: Iliadis, L., Maglogiannis, I., Plagianakos, V. (eds.) *AIAI 2018. IAICT*, vol. 520, pp. 263–274. Springer, Cham (2018). [https://doi.org/10.1007/978-3-319-92016-0\\_24](https://doi.org/10.1007/978-3-319-92016-0_24)
9. Sevakula, R.K., Au-Yeung, W.T.M., Singh, J.P., Heist, E.K., Isselbacher, E.M., Armoundas, A.A.: State-of-the-Art machine learning techniques aiming to improve patient outcomes pertaining to the cardiovascular system. *J. Am. Heart Assoc.* **9**(4), e013924 (2020)
10. Poddar, M.G., Birajdar, A.C., Virmani, J., Kriti: Automated classification of hypertension and coronary artery disease patients by PNN, KNN, and SVM classifiers using HRV analysis. In: Dey, N., Borra, S., Ashour, A.S., Shi, F. (eds.) *Proceedings of the Machine Learning in Bio-Signal Analysis and Diagnostic Imaging*, pp. 99–125. Academic Press (2019)
11. van IJzendoorn, D.G.P., Szuhai, K., Briaire-de Bruijn, I.H., Kostine, M., Kuijjer, M.L., et al.: Machine learning analysis of gene expression data reveals novel diagnostic and prognostic biomarkers and identifies therapeutic targets for soft tissue sarcomas. *PLoS Comput. Biol.* **15**(2) (2019)
12. Abbas, M., El-Manzalawy, Y.: Machine learning based refined differential gene expression analysis of pediatric sepsis. *BMC Med. Genomics* **13**, 122 (2020)
13. Bottomly, D., Walter, N.A.R., Hunter, J.E., et al.: Evaluating gene expression in C57BL/6J and DBA/2J mouse striatum using RNA-seq and microarrays. *PLoS ONE* **6**(3), e17820 (2011)
14. Ester, M., Kriegel, H.-P., Sander, J., Xiaowei, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the Second International Conference on Knowledge Discovery in Databases and Data Mining*, pp. 226–231. AAAI Press, Portland (1996)
15. Frazee, A.C., Langmead, B., Leek, J.T.: ReCount: a multi-experiment resource of analysis-ready RNA-seq gene count datasets. *BMC Bioinform.* **12**, 449 (2011)



# Comparison of Batch Effect Removal Methods for High Dimensional Mass Cytometry Data

Aleksandra Suwalska<sup>1</sup> , Nelita du Plessis-Burger<sup>2</sup> , Gian van der Spuy<sup>2</sup> ,  
and Joanna Polanska<sup>1</sup> 

<sup>1</sup> Department of Data Science and Engineering, Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology, Gliwice, Poland  
{Aleksandra.Suwalska, Joanna.Polanska}@polsl.pl

<sup>2</sup> Division of Molecular Biology and Human Genetics, Faculty of Medicine and Health Sciences, University of Stellenbosch, Cape Town, South Africa  
{nelita, gvds}@sun.ac.za

**Abstract.** Mass cytometry is one of the most popular single-cell technology that can measure over 50 markers simultaneously for millions of cells. Due to the high-dimensional nature of the dataset, manual analysis is difficult. Furthermore, during biological sample preparation, some problems may arise that need to be dealt with. One of the problems is a batch effect that can be introduced to the data because of experimental design or different experimental settings and techniques used. There are several bioinformatical solutions to correct the batch effect. Some of them require technical replicates (CytofBatchAdjust, CytoNorm and CytofRUV), others can work for a limited number of cells only (iMUBAC). An interesting aspect is how the batch correction method affects the results in terms of the number and quality of identified cell groups and to what extent the batch effect was removed. In the study, the two batch effect methods were compared, that do not require technical replicates, cyCombine and iMUBAC, applied to a real dataset with over 2 million bronchoalveolar lavage cells. Results were presented with the original mISO plots. Cells were clustered based on the original and corrected marker profiles with the PARC algorithm. After the correction, the number of clusters decreased from 24 to 22 (iMUBAC) and 18 (cyCombine). The homogeneity of clusters expressed as an effect size measure increased after the cyCombine correction ( $p$ -value =  $4.38 \cdot 10^{-7}$ ) in contrast to iMUBAC ( $p$ -value = 0.4628). The results indicate the superiority of cyCombine over iMUBAC for the real dataset if the within-cluster marker profile similarity is considered.

**Keywords:** CyTOF · Batch effect · Clustering

## 1 Introduction

One of the most popular single-cell analysis technology is mass cytometry (CyTOF) which combines mass spectrometer with inductively coupled plasma and Time-of-flight detector (TOF). This technique uses stable isotopes of rare metals to label the antibodies, therefore, enabling the measurement of over 50 parameters [1]. The measured markers



indicate functional and phenotypic traits of the cells making it possible to identify specific subtypes.

Mass cytometry overcomes many of the flow cytometry limitations like spectra overlap. However, measuring more parameters translates to a more difficult analysis because of the higher data dimensionality. Moreover, mass cytometers are used to measure millions of cells but have lower throughput than flow cytometry [2]. This implies that to process the same number of cells, the mass cytometer has to run longer (for example several days) than the flow cytometer. This may lead to some problems, including signal fluctuations and a presence of a batch effect.

The batch effect is a technical variance introduced to the data during experimenting and it makes it difficult to reveal the real biological variance, therefore a lot of effort is made to find methods that can remove batch effect accurately. The batch effect not only results from the experimental design but is also present between datasets from different experiments or experimental techniques (like CITE-seq) that could be combined and analysed together to get a better view of the biological problem [3].

Some methods for the batch effect correction in mass cytometry data have been proposed so far. The most popular ones are CytobatchAdjust, CytoNorm, CytobatchRUV, iMUBAC and cyCombine. CytobatchAdjust [4] is a method that uses technical replicates that are included in each run to appropriately adjust all samples to a reference batch without manual intervention. CytoNorm [5] requires an identical control sample to be included in each batch to perform batch-to-batch correction of variability. First, the control sample is clustered with FlowSOM [6] to find cell subpopulations and goal distributions are determined based on the quantiles calculated for the clusters. The original values are translated to follow the goal distribution. Then the rest of the samples are normalized using learned models leaving only a biological variation in the data. CytobatchRUV [7] is based on RUV-III method, which was applied mainly for technologies like RNA-Seq or microarrays. The method uses pseudo-replicates to estimate and remove the artificial variation from protein expression. iMUBAC [8] uses only healthy controls for batch correction. The data are downsampled to a fixed number of cells per batch to reduce the computations. Then, the expression values are corrected with Harmony [9] with the default parameters. cyCombine [3] allows integration of cytometry data from different batches, experiments and experimental techniques. This technique uses Combat [10], which was introduced to remove batch effects from microarray expressions, to remove technical variation. Similar cells are grouped by a self-organizing map (SOM) and the groups are batch corrected.

The methods propose a lot of diagnostic plots and measurements that will indicate the presence of batch effect in data and evaluate the effectiveness of batch correction. Some of the techniques are Earth Mover's Distance (EMD) for batch-to-batch comparisons presented on different types of plots; distribution of each marker in each batch; multidimensional scaling plot (MDS); comparison of EMD values before and after correction [3] and visualization of the batches on two-dimensional plots after dimensionality reduction. However, these methods, especially the last one, are not clear indicators of the batch effect when the number of cells exceeds millions of cells.

In the study, two of the mentioned batch effect removal methods: iMUBAC and cyCombine were applied and compared. A real dataset was used that does not have

technical replicates, therefore the correction with the other three techniques (CytobatchAdjust, CytoNorm and CytobatchRUV) is impossible since they require the replicates. To evaluate the effectiveness of the methods, after batch correction the dataset was clustered with PARC [11]. An interesting aspect is how the batch correction method affects the results in terms of the number of identified cell subtypes. Since iMUBAC uses only healthy samples for the correction the cyCombine was also applied to the same healthy samples. A proposed new visualization technique can help visualize the batch effect and if it decreased after correction. The technique is also helpful to visualize the results of clustering.

## 2 Materials and Methods

### 2.1 Dataset

Data used in the study contained healthy control samples of bronchoalveolar lavage cells (BALC) from studies on drug-resistant tuberculosis. Bronchoscopies were performed in the bronchoscopy theatre, ward A5, Tygerberg Hospital (TBH) from Cape Town, South Africa. The dataset was measured in seven batches with CyTOF2 instrument, located at the SATVI institution (South African Tuberculosis Vaccine Initiative) at the University of Cape Town. For each cell, a set of 32 parameters (markers) was collected, where 19 of them were extracellular (phenotype features) and 13 intracellular (functional features).

Before using the batch effect removal methods, the dataset was normalized with MATLAB Normalizer v0.3 software and the samples were filtered during manual gating to discard debris, dead cells, beads or doublets from the analysis. Each marker expression was arcsinh transformed with a co-factor of 5 for visualization purposes. The total number of cells used in the study was equal to 4,145,712. Table 1 presents the number of cells in each batch.

**Table 1.** The number of cells in each batch.

Batch no	Number of cells
Batch 1	761,230
Batch 2	598,492
Batch 3	205,958
Batch 4	329,228
Batch 5	341,007
Batch 6	1,449,084
Batch 7	460,713
<b>Total</b>	<b>4,145,712</b>

### 2.2 Batch Effect Correction

Cells were corrected with the iMUBAC method using the default parameters set by the authors except for maxN that was set to 300,000. This means a maximum of 300,000

cells were randomly selected for each batch to correct, that is 2,005,958 cells (about 50% of data). To make a fair comparison, the same set of cells was used for cyCombine correction, although this method is not limited to the number of cells and the patient’s status. This algorithm was also run with the default parameters.

**2.3 Cell Subtypes Identification**

For the cell subtypes identification on corrected data, the PARC [11] algorithm was used with the default parameters. Although the FlowSOM method is the most often used clustering tool for mass cytometry data because it is fast and gives satisfactory results in most cases, it has no automatic way to precisely estimate the number of cell types. FlowSOM often results in an overestimated number of clusters that have to be merged manually after examination of the clusters’ content. Therefore it was decided to apply a newer method, which automatically finds the number of clusters that is consistent with the experts’ opinion. PARC constructs a nearest-neighbour graph with a hierarchical navigable small world and prunes the edges based on their edge-weight distribution. Then, the community detection is performed with the Leiden algorithm. The method is fast and applicable to many single-cell technologies. The algorithm works well for high-dimensional data and efficiently detects rare cell subpopulations. The clustering was applied to cells described by the 32 markers after batch effect correction. The resulting cluster assignments were then transferred to the corresponding uncorrected cells for comparison purposes.

**2.4 Statistical Comparison of Methods**

Having observations before and after batch effect correction with PARC clusters, the effect of iMUBAC and cyCombine on the resulting clusters was checked with the ANOVA post-hoc Q Tukey test, for each marker. For the pairwise comparisons of marker’s expression between clusters, an effect size was calculated (1) where  $m_A$  and  $m_B$  are mean values of marker expression in clusters A and B;  $N_{ps}$  (2) is a pooled sample size;  $N_A$  and  $N_B$  are sample sizes in the clusters;  $N$  is a total number of samples;  $SS_{within}$  is a sum of squares within  $k$  groups (all clusters).

$$d_{AB} = \frac{m_A - m_B}{SE * \sqrt{N_{ps}}} = \frac{m_A - m_B}{\sqrt{\frac{SS_{within}}{N-k} * \frac{1}{N_{ps}} * \sqrt{N_{ps}}}} = \frac{m_A - m_B}{\sqrt{\frac{SS_{within}}{N-k}}} \tag{1}$$

$$N_{ps} = \frac{2}{\frac{1}{N_A} + \frac{1}{N_B}} \tag{2}$$

The pairwise comparison resulted in a set of  $d_{AB}$  effect size values for each marker and the median value was calculated as the global effect size. Therefore, for each experiment (iMUBAC, cyCombine) 32 values of  $d_{AB}$  were collected before and after batch correction (64 values in total per experiment). The effect sizes were then compared with the Wilcoxon signed-rank test. The assumption was, that after batch effect removal the heterogeneity of marker expression between clusters will be higher than before the correction. The Wilcoxon test was performed to check if the homogeneity of markers in the

clusters after correction is greater than before at a 5% significance level. The median shift of values was also included next to the p-value.

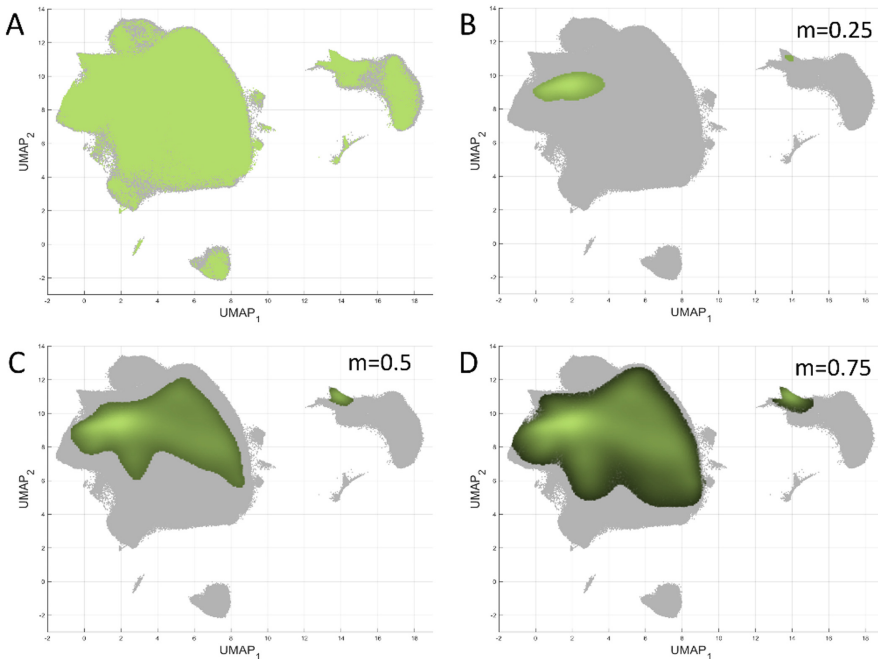
Moreover, the centroids of clusters (vectors of 32 mean marker values) after iMUBAC and cyCombine correction were grouped using agglomerative clustering with Spearman's rank correlation coefficient as the distance metric to find cluster equivalents in both methods. If a cyCombine cluster's marker profile is similar to the iMUBAC cluster's profile, then the two clusters probably describe the same type of cells.

## 2.5 UMAP Transformation

One of the most frequently used visualization methods for high dimensional mass cytometry data is Uniform Manifold Approximation and Projection (UMAP) [12]. With this algorithm, it is possible to see high dimensional data structures projected on the two-dimensional space. It is convenient to present analysis results with the use of UMAP plots, for example, cell assignments after grouping. In the study, generated UMAP embeddings from the batch corrected samples and the same transformation was used to present results for raw data. The assumption was, that each cell will be placed in a different position in the UMAP space before and after correction. The transformation is applied to the same set of features (markers) so it is possible to observe the change in cell placement and the effect of correction. To transform new data with the learned model, a simple neural network for regression tasks with three fully connected layers was proposed. The number of neurons in the layers was 100, 50 and 25 and the activation function was ReLU. The network took as input a vector of 32 markers and the output was a learned UMAP embedding. The performance of the network was evaluated with the coefficient of determination.

## 2.6 mISO Plots—Visualization of Results in High Dimensional Space

Because of the large number of observations (about 2 mln), presenting cell assignments to specific groups by colouring the cells on the plot may not be the best choice of visualization, especially when the cells on UMAP plots densely occupy a specific region. Due to this, the results could be misinterpreted. In the paper, a proposed new type of visualization is used—median isoline plot (mISO) that can be superimposed on UMAP plots. It is based on isolines that determine the density of the points. mISO plot has a parameter  $m$  that defines the density level above which the data will be displayed. The default value of  $m$  is 0.5 (median) and this value is used in the study. This method clearly shows the cell assignments by presenting the area of the highest concentration of the group of observations, a “core” of the group at the specified level (Fig. 1.).



**Fig. 1.** mISO plot as a proposition of high-dimensional data visualization technique that reveals the most densely populated regions. A) One batch of samples (green) looks as if it is evenly spreading over the data space (grey points). B) mISO plot showing region above 25% quartile isoline. C) mISO plot showing region above 50% quartile (median) isoline. D) mISO plot showing region above 75% quartile isoline.

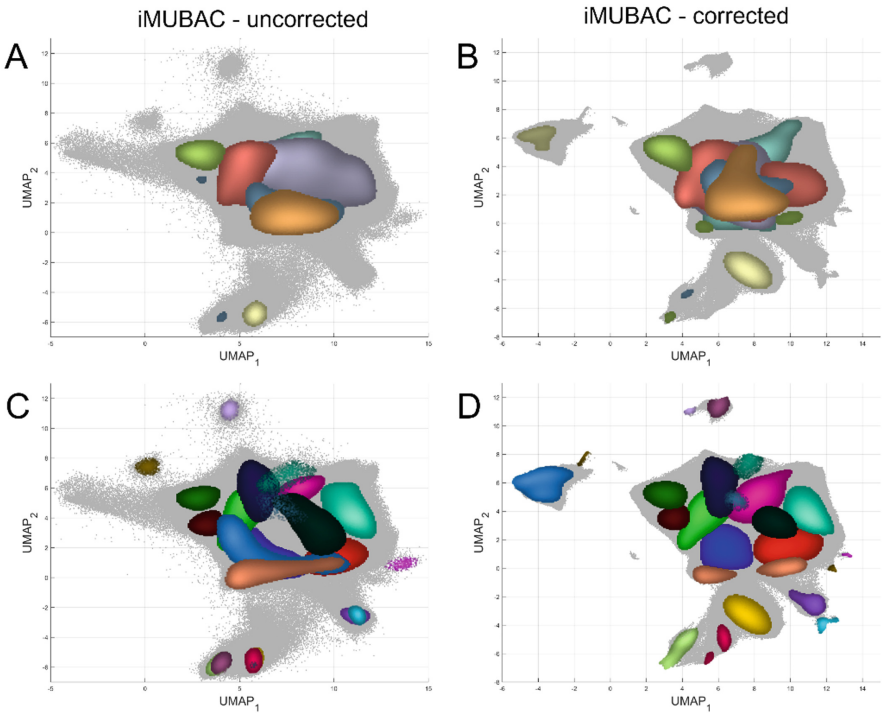
## 2.7 Technical Details

The analysis was conducted in Python, R and MATLAB 2020a environments. The system implementation was carried out using GeCONiI server (Intel Xeon Gold 6226R CPU 64 threads, 2.9GHz, GPU: 3x NVIDIA Tesla V100-PCIE with 1x 16GB and 2x 32GB).

## 3 Results

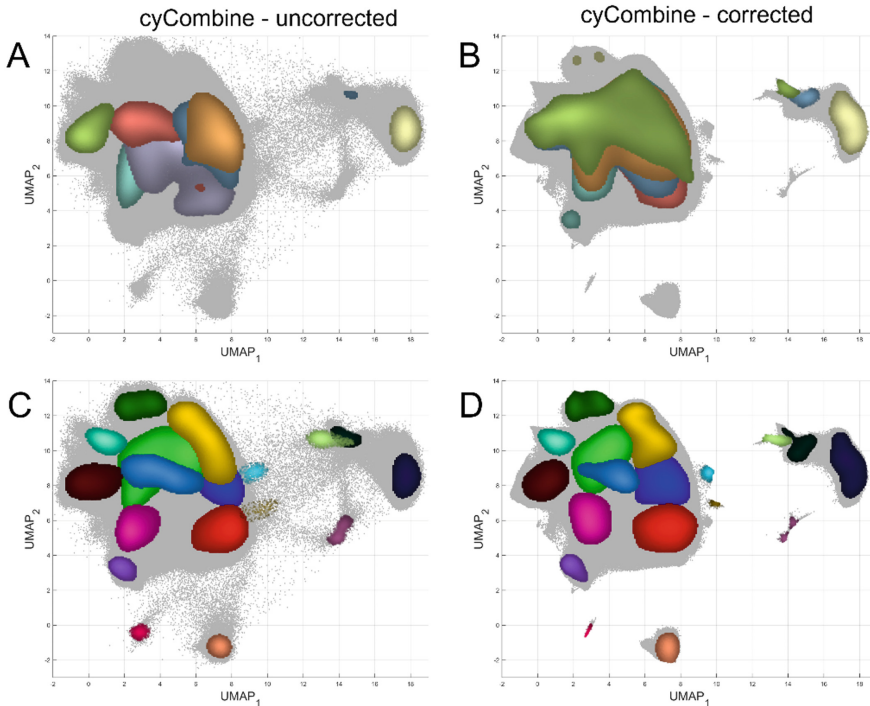
The same set of 2 mln healthy cells from seven samples were corrected with iMUBAC and cyCombine to remove technical variances present in the data. Using mISO plots, the effect of batch correction was visualized on UMAP. Figure 2 shows the effect of the iMUBAC method on each sample distribution in the UMAP space. The uncorrected samples (Fig. 2.A.) are located in different regions on the plot, while the corrected samples (Fig. 2.B.) share the space to a large extent - is visible, that they overlap. The same behaviour can be observed for samples corrected with cyCombine (Fig. 3.). Despite the differences in the location of the cells (different UMAP shape), the highest density of each sample before correction (Fig. 3.A.) occupy a separate space than after applying cyCombine (Fig. 3.B.). It can be also noticed that in the cyCombine case the overlap

is bigger and more regular than in the iMUBAC. The overlap of samples is desirable because it indicates the removal of technical variance in data.



**Fig. 2.** Visualization of results after iMUBAC correction. A) Samples before batch correction. B) Samples after batch correction. C) Clusters found with PARC before correction. D) Clusters after batch correction.

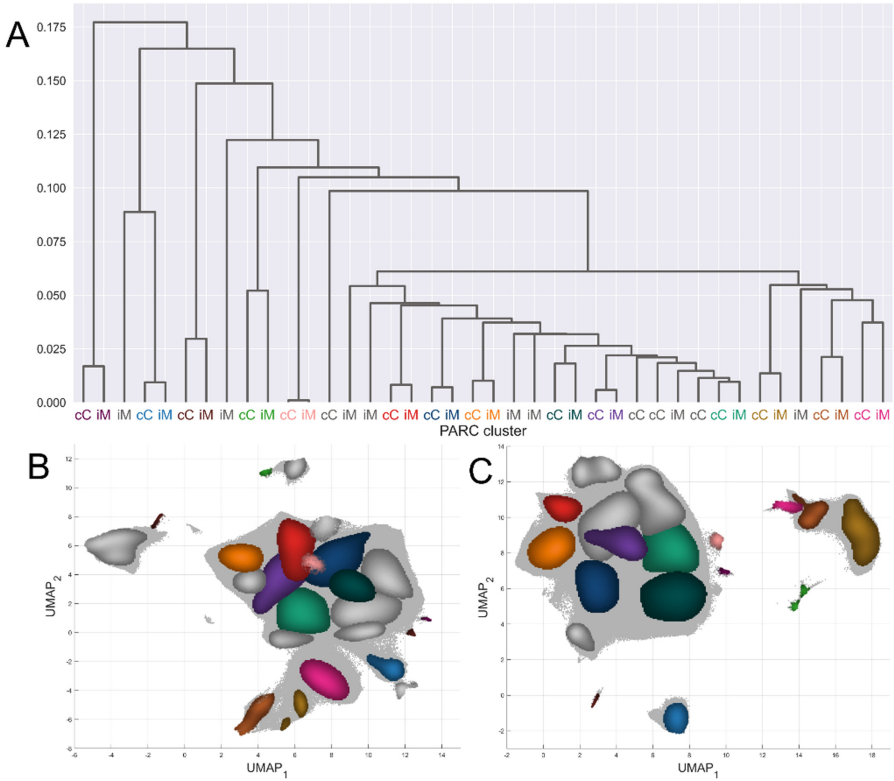
Cells before and after batch removal were subjected to cell subtypes identification with the PARC algorithm. The results were presented on the mISO plots to visualize the impact of the correction on the cluster location. The clustering of cells was conducted after the batch correction and the same cluster assignments were transferred to the raw cell expressions. For the iMUBAC corrected cells, the PARC algorithm found 22 clusters (Fig. 1.D.) that overlap strongly when transferred to raw data (Fig. 2.C.). For the cyCombiner, 18 clusters were found (Fig. 3.C-D.) and the results are similar to iMUBAC, however, the differences before and after correction are smaller.



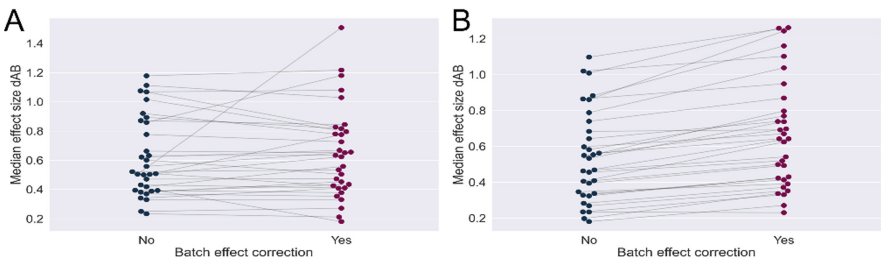
**Fig. 3.** Visualization of results after cyCombine correction. A) Samples before batch correction. B) Samples after batch correction. C) Clusters found with PARC before correction. D) Clusters after batch correction.

The cluster centroids were grouped with agglomerative clustering and visualized with a dendrogram (Fig. 4.). It can be noticed, that most of the clusters (14 clusters) from one experiment have a similar cluster from the second experiment. The pairs are presented in the same colour on the dendrogram (Fig. 4.A.) as well as the mISO plots (Fig. 4.B-C.). Clusters that do not have a direct pair are presented in grey.

To better understand the differences between the clusters from both experiments, each marker expression was compared among all the clusters with the ANOVA post-hoc Q Tukey test and the  $d_{AB}$  effect size measure. The visualization of the median  $d_{AB}$  values for each marker before and after correction is presented in Fig. 5. The effect size before iMUBAC correction did not change much after the correction, in contrast to the cyCombine method, where the values have increased. The observation was validated with the Wilcoxon test. For the iMUBAC correction, the p-value was 0.4628 with a median shift of 0.0011. For the cyCombine correction, the p-value was  $4.38 \cdot 10^{-7}$  (p-value  $\approx 0$ ) with a median shift of 0.0961.



**Fig. 4.** Comparison of the clustering results after batch correction. A) Dendrogram showing similar clusters (colour-coded) between the two correction methods; iM - cluster created after iMUBAC correction; cC—cluster created after cyCombine correction. B) mISO plot of iMUBAC clusters which have a direct counterpart among cyCombine clusters. C) mISO plot of cyCombine clusters which have a direct counterpart among iMUBAC clusters. The clusters that are most similar according to the dendrogram share the same colour. Clusters that do not have a similar pair from the other experiment are presented in grey.



**Fig. 5.** Median effect sizes from post-hoc ANOVA test before and after batch correction. A) iMUBAC. B) cyCombine.



## 4 Discussion

Mass cytometry is a very popular technology used for deep phenotyping, that overcomes the limitations of traditional flow cytometry. However, the ability to measure dozens of parameters and millions of cells makes the analysis of CyTOF data more difficult and impossible to conduct manually. Therefore, automatic tools that accurately and quickly perform the relevant parts of the analysis are constantly being sought. Difficulties in the implementation of these tools are the presence of artefacts, contamination and other problems arising during the preparation of the sample and the measurement itself, as the batch effect.

There are many proposed solutions for the batch effect problem and how to decrease or remove it, with the most popular five: CytofBatchAdjust, CytoNorm, CytofRUV, iMUBAC and cyCombine. Unfortunately, the first three solutions require technical replicates that are not always available. There is also the question of which method is the best to use for our data. Also, how much can the results differ after applying a specific method of correcting the batch effect? To address this question, a comparative analysis of the results of cell identification after data correction by cyCombine and iMUBAC methods was conducted. For the comparison, the same set of observations and preprocessing was used, as well as the default parameters of each method.

For the visual comparison, it was decided to generate UMAP plots, where for corrected data the UMAP was created and then, the learned transformation was applied to uncorrected data to see how the cells changed their position. The effect of transformation is visible in Figs. 2 and 3, as some blurring of the grey shapes that were formed on the UMAP plot after the correction. Because the effect size introduces an artificial variance to the data, it is assumed that before batch correction each of the batches (samples) will take a separate place in the UMAP space. Therefore after batch correction, the samples should overlap more than before the correction, since this issue is reduced. This is reflected in the UMAP transformation (the blurring indicating movement of points depending on the marker values) as well as in the mISO plots (Fig. 2.A. and Fig. 3.A.).

With mISO plots, it can be seen that iMUBAC correction had less impact on the placement of the cells than cyCombine, where the samples overlap almost completely. This would be impossible to observe on a regular UMAP plot with a specified colour of the points corresponding to each batch. The proposed mISO plot makes it easier to observe the change in expression values. It is expected that cells with different values will lie further apart in the UMAP space and with the batch effect, this behaviour can be seen on the mISO plot for samples. After the correction, samples should overlap because the technical variance is decreased and cells of the same type lie close to each other among all batches. Visually, cyCombine (Fig. 3.A.) has a better correction of values than iMUBAC (Fig. 2.A.).

After batch effect removal, clustering of the dataset was conducted with the PARC method and default parameters. The algorithm works fast (about an hour) with over 2 million observations and automatically finds the number of clusters. After iMUBAC correction, the number of clusters was 22 and after cyCombine correction, the number was 18. The results are presented in Fig. 2.D. and 3.D. It can be seen especially for iMUBAC correction, that the clusters transferred to the raw data are overlapping (for example the six clusters at the bottom of Fig. 2.C. overlap completely). This indicates

that without the batch effect correction the cell types that overlap would not be found but joined to other clusters. It is expected that after the correction cores of the clusters will take a separate place which is visible for both methods.

Clustering visualization with the mISO plots is not sufficient to infer the superiority of any method but someone can wonder which division into clusters is better since the different number of clusters results only from the batch correction method. Dendrogram (Fig. 4.A.) shows that 14 of the clusters are similar between the methods and the clusters lie in a similar place despite the different UMAP transformations (Fig. 4.B-C.). It can be concluded that the clusters represent the same type of cells. However, there are 8 clusters after iMUBAC and 4 after cyCombine corrections that do not have a direct counterpart and because there are more iMUBAC clusters, they probably contain a mixture of different cell subtypes.

The cluster assignments were transferred to the corresponding raw expression values to examine the differences in the values after correction and measure how the homogeneity of markers within clusters has changed. For each experiment (iMUBAC, cyCombine) an effect size was calculated for each marker in pairwise comparisons between cell subtypes. A median of the pairwise effect sizes was computed to get one overall value for each marker. This resulted in 32 effect size measures (Fig. 5.) for each case (32\*4 in total): for data before and after iMUBAC correction grouped into 22 clusters and data before and after cyCombine correction grouped into 18 clusters. The Wilcoxon signed-rank test revealed that the differences in effect size measures after correction are not greater than before for the iMUBAC method ( $p\text{-value} = 0.4628$ ). This may indicate that the batch effect was not significantly removed and the clusters may contain a mixture of several cell types. The raw dataset was also clustered with PARC (rather than transferring the clusters gained after correction) and the algorithm found 24 clusters. After iMUBAC correction, it was 22 clusters so the reduction from 24 to 22 is minor compared to cyCombine (18 clusters). This may suggest that the iMUBAC correction is not sufficient to remove the batch effect.

Taking all of the above into consideration, it can be concluded that cyCombine is a better method for batch effect removal than iMUBAC. It is not limited to healthy (control) patients—cyCombine removes the batch effect from the whole dataset without interfering with biological differences. cyCombine also does not need downsampling—it works fast and efficiently for large datasets with millions of cells so the correction can be more accurate. The measures of effect size after correction increased significantly ( $p\text{-value} = 4.38 \times 10^{-7}$ ) which means the homogeneity of the clusters is higher therefore the clusters differ more.

However, the obtained results should be further analysed and compared with other methods on different datasets. It is possible that cyCombine may be better for the used dataset but another problem may need another batch effect removal method to effectively reduce the technical variance.

In the future, the proposed mISO plot visualization method will be shared as a Python package for general use in the problems of high-dimensional data analysis.

## 5 Conclusions

In the study, two batch effect removal methods, iMUBAC and cyCombine, were compared. Both methods significantly decrease the batch effect. Cell clustering in the domain of corrected marker profiles resulted in a decreased number of detected cell groups for both methods compared to uncorrected data. The cell-type marker homogeneity increased after applying cyCombine in contrast to iMUBAC, where the one-vs-other effect size analysis did not reveal significant improvement. The results indicate the superiority of cyCombine over iMUBAC for the real dataset if the within-cluster cell-type marker profile similarity is considered.

**Acknowledgment.** AS benefits from the European Union scholarship through the European Social Fund (grant POWR.03.05.00–00-Z305). JP was financed by 02/070/BK\_22/0033 project. Calculations were carried out using GeCONiI infrastructure funded by NCBiR project no. POIG.02.03.01–24-099/13.

## References

1. Bjornson, Z.B., Nolan, G.P., Fantl, W.J.: Single-cell mass cytometry for analysis of immune system functional states. *Curr. Opin. Immunol.* **25**(4), 484–494 (2013)
2. Atkuri, K.R., Stevens, J.C., Neubert, H.: Mass cytometry: A highly multiplexed single-cell technology for advancing drug development. *Drug Metab. Dispos.* **43**(2), 227–233 (2015)
3. Pedersen, C.B., et al.: Robust integration of single-cell cytometry datasets. *bioRxiv* (2021)
4. Schuyler, R.P., et al.: Minimizing batch effects in mass cytometry data. *Front. Immunol.* 2367 (2019). <https://doi.org/10.3389/fimmu.2019.02367>
5. Van Gassen, S., Gaudilliere, B., Angst, M.S., Saeys, Y., Aghaeepour, N.: CytoNorm: A normalization algorithm for cytometry data. *Cytometry A* **97**(3), 268–278 (2020)
6. Van Gassen, S., et al.: FlowSOM: Using self-organizing maps for visualization and interpretation of cytometry data. *Cytometry A* **87**(7), 636–645 (2015)
7. Trussart, M., Teh, C.E., Tan, T., Leong, L., Gray, D.H., Speed, T.P.: CytofRUV: Removing unwanted variation to integrate multiple CyTOF datasets. *bioRxiv* (2020)
8. Ogishi, M., et al.: Multibatch cytometry data integration for optimal immunophenotyping. *J. Immunol.* **206**(1), 206–213 (2021)
9. Korsunsky, I., et al.: Fast, sensitive and accurate integration of single-cell data with harmony. *Nat. Methods* **16**(12), 1289–1296 (2019)
10. Johnson, W.E., Li, C., Rabinovic, A.: Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* **8**, 118–127 (2007)
11. Stassen, S.V., Siu, D.M., Lee, K.C., Ho, J.W., So, H.K., Tsia, K.K.: PARC: Ultrafast and accurate clustering of phenotypic data of millions of single cells. *Bioinformatics* **36**(9), 2778–2786 (2020)
12. McInnes, L., Healy, J.: UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv e-prints* 1802.03426 (2018)

# **Next Generation Sequencing and Sequence Analysis**



# Evaluating Performance of Regression and Classification Models Using Known Lung Carcinomas Prognostic Markers

Shrikant Pawar<sup>1</sup>(✉), Karuna Mittal<sup>2</sup>, and Chandrajit Lahiri<sup>3</sup>

<sup>1</sup> Department of Computer Science and Biology, Claflin University, Orangeburg, USA  
spawar@claflin.edu

<sup>2</sup> Department of Biomedical Sciences, Emory University, Atlanta, USA

<sup>3</sup> Department of Biological Sciences, Sunway University, Petaling Jaya, Malaysia  
chandrajitl@sunway.edu.my

**Abstract.** Differential expression study between tumor and non-tumor cells aids lung cancer diagnostic classifications and prognostic prediction at various stages. Support vector machine (SVM) learning is used to categorize the morphology of lung cancer. Logistic regression, random forest, and group lasso-based models are used to model dichotomous outcome variables. The purpose is to take groups of observations and design boundaries to forecast which group future observations belong to base measurements. The performance of these selected regression and classification models using lung cancer prognostic indicators is evaluated in this article. The presented results might guide for further regularizations in classification techniques using known lung carcinoma marker genes.

**Keywords:** Regression · Lung carcinomas · Predictions

## 1 Introduction

Among all malignancies, lung cancer caused the most considerable loss of pay, totaling \$21.3 billion in year 2018–19 [1]. However, the specific environmental and genetic etiology of a person's lung cancer is unknown, and it can be described as a tumor forming in the lung when altered cells escape the immune system and grow out of control. Despite the fact that many lung cancer research findings have been published, scientific advancement in lung cancer research is still limited. Lung cancer diagnostic classifications and prognosis prediction at various stages are aided by differential expression analysis between tumor and non-tumor cells. Attempts have been undertaken to find genes linked to lung cancer symptoms. Lung cancer morphology categorization has been performed using support vector machine learning techniques [2]. Alanni et al. devised a deep gene selection technique for cancer classification from microarray datasets [3]. The results of their experiments revealed an average sensitivity of 95.22% and a specificity of 77.39%. Several machine learning methods have also been utilized to identify 13 top genes in lung adenocarcinoma and lung squamous cell cancer [4]. To learn cancer

type classification based on TCGA data, Mohammed et al. employed the least absolute shrinkage and selection operator (LASSO) as a feature selection approach [5]. In addition to cancer classification and biomarker identification, overlapping feature selection strategies have been used [6]. Squamous cell lung cancer (LUSC) has been associated to four genes CCNA2 (890), AURKA (6790), AURKB (9212), and FEN1 (2237) [7], while lung adenocarcinoma (LUAD) has been linked to four genes (CD44 (960), CCND3 (896), NCALD (83988), MACF1 (23499), and RAMP2-AS1 (10266)). In a comprehensive genomic study of squamous cell lung tumors [9], one gene, TP53 (7157), was found to be altered in virtually all cases. To model dichotomous outcome variables, logistic regression, random forest, support vector machines (SVM), and group lasso-based models are utilized [10, 11]. The purpose is to take groups of observations and design boundaries to forecast which group future observations belong to base on their measurements. The performance of these selected regression and classification models using lung cancer prognostic indicators is evaluated in this article.

## 2 Dataset and Methodology

We chose to test performance of each of the 4 techniques on 3 different datasets with lung LUAD (517 tumor, 59 normal) [12], LUSC (501 tumor, 51 normal) [9] and non-small cell lung carcinomas (NSCLC) (91 tumor, 65 normal subjects) [13]. Libraries randomForest, caret was used for random forest application, library kernlab and e1071 for SVM, and glmnet for regression. Functions svm(kernel = "radial", cost = 10, gamma = 1), predict(), glm(), wald.test(), and glmnet() were utilized for performing k-fold cross-validation to find optimal lambda value that minimizes test mean squared error (MSE) [14–16]. Cross validations were performed with 70:30 training to testing splits. Response value was considered 0/living and 1/death status. Sum of squares total (SST), sum of squares error (SSE) and R-squared value on a response variable (y) were calculated as follows:

```
sst <- sum((y-mean(y))^2).
sse <- sum((y_predicted-y)^2).
rsq <- 1 - sse/sst.
```

All the code for accessing data and methodology can be found at authors GitHub account: <https://github.com/spawar2/Regression-Lung-Carcinoma/tree/main>.

## 3 Results

### 3.1 Prediction Performance of Random Forest

Test classification accuracy of 55% was obtained on selected 10 genes expression values with an 30–78 range for 95% CI. The P value was seen insignificant with sensitivity and specificity of 14 and 81% respectively. The 10 genes were not found to exclusively classify the survival response status. We also tested this classification approach on different combinations of these 10 marker genes, and results were consistent. Table 1 provides details of test and training metrics of random forest.

**Table 1.** Test and training metrics of random forest.

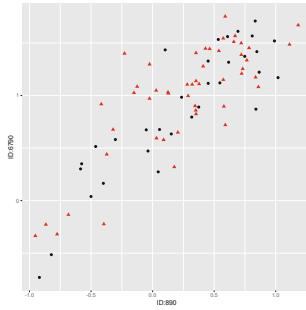
	Train: Type of random forest: regression Number of trees: 500 No. of variables tried at each split: 3 Mean of squared residuals: 0.2553676 % Var explained: -6.72	Test: Type of random forest: classification Number of trees: 500 No. of variables tried at each split: 3 OOB estimate of error rate: 40.62%
Accuracy	1	0.5556
95% CI	(0.944, 1)	(0.3076, 0.7847)
No information rate	0.6094	0.6111
P-value [Acc > NIR]	1.709e-14	0.7680
Kappa	1	-0.0435
Sensitivity	1	0.14286
Specificity	1	0.81818
Pos pred value	1	0.33333
Neg pred value	1	0.60000
Prevalence	0.3906	0.38889
Detection rate	0.3906	0.05556
Detection prevalence	0.3906	0.16667
Balanced accuracy	1	0.48052
'Positive' class	0	0

### 3.2 Prediction Performance of SVM

Testing SVM with 10 marker gene expression on a survival response variable predicted 85% subjects living/0 correctly ( $n = 20$ ), and 24% subjects dead/1 correctly ( $n = 62$ ) (Table 2). The test group was randomly selected with Fig. 1 showing dispersion of 2 groups for genes 890 and 6790. We found similar dispersion patterns for other genes and throughout all the 3 separate datasets. SVM poorly classifies survival response status with known marker genes.

**Table 2.** SVM classification of test data.

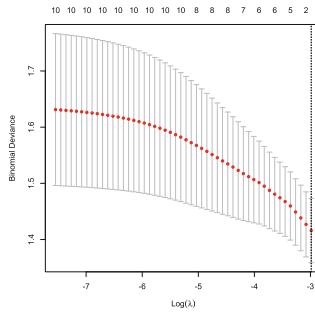
	0	1
0	17	3
1	15	47



**Fig. 1.** Dispersion of survival and dead subjects for genes 890 and 6790.

### 3.3 Prediction Performance of Logistic Regression and LASSO

Testing prediction probabilities from LASSO ranged from 0.3–0.7 (Table 3). A weighted distance between the unrestricted estimate (Wald test) P value was found to be insignificant. The Chi-squared value of 0.89 with a P value > 0.05 also states insignificant prediction probabilities. The least squares regression tries to find coefficient estimates that minimize the sum of squared residuals (RSS). It can be presented with function:  $RSS = \sum (y_i - \hat{y}_i)^2$ ,  $y_i$ : is actual response value for the  $i^{th}$  observation and  $\hat{y}_i$ : is the predicted response value based on the multiple linear regression model. Figure 2 depicts calculates the binomial deviance (binomial log-likelihood) in the test dataset. The test data R square value of  $-6.70$  was obtained stating the selected model does not follow the trend of the data, therefore leading to a worse fit than the horizontal line.



**Fig. 2.** Calculation of binomial deviance (binomial log-likelihood) in the test dataset.

**Table 3.** Prediction probabilities from LASSO.

Status	Predicted probability
0	0.6545150
0	0.6875263

(continued)



**Table 3.** (continued)

Status	Predicted probability
0	0.5171204
1	0.6935557
1	0.6536800
1	0.7114294
1	0.7818345
1	0.3633772
1	0.8720003
1	0.6644866
1	0.6111339
1	0.6527981

## 4 Discussion and Future Scope

The biological literature of the selected 10 key genes is enriched by their new roles associated to lung cancer, which have moved from an indirect to a direct association, i.e., to become new biomarkers. In many cases, indirect impacts are more important than direct effects because direct effects can be seen and controlled, whereas indirect effects are difficult to detect and control. We wanted to test their effects on response variable using selected regression and classification techniques. We find insignificant correlations with response variable. These findings are consistent for all the three cancer types. There can be several reasons of these outcomes. Growing more than one type of lung cancer is uncommon among all known lung cancer types. As a result, competing risk factor models can be extremely effective at modeling a variety of lung cancer forms. Further, confounding factors (age, gender, preexisting conditions, etc.) also significantly affect regression predictions. The expression data is rarely linearly separable, and prone to noise and overfitting. Although we did take care of limiting outliers, regression techniques are oversensitive to nominal outliers. One limitation of this study is multicollinearity, dimensionality reduction techniques are needed to be implemented to address issue of multicollinearity apart from above confounding factors. The presented results might guide for further regularizations in classification techniques using known lung carcinoma marker genes.

**Author Contributions.** SP and CL conceived the concepts, planned, and designed the article. SP and CL primarily wrote and edited the manuscript.

**Funding Source.** No external funding has been utilized for this study.




**Competing Interests.** The authors declare that they have no competing interests.

## References

1. Islami, F., et al.: National and state estimates of lost earnings from cancer deaths in the united states. *JAMA Oncol.* **5**(9), e191460 (2019). <https://doi.org/10.1001/jamaoncol.2019.1460>
2. Podolsky, M.D., Barchuk, A.A., Kuznetsov, V.I., Gusarova, N.F., Gaidukov, V.S., Tarakanov, S.A.: Evaluation of machine learning algorithm utilization for lung cancer classification based on gene expression levels. *Asian Pac. J. Cancer Prev.* **17**(2), 835–838 (2016). <https://doi.org/10.7314/apjcp.2016.17.2.835>. PMID: 26925688
3. Alanni, R., Hou, J., Azzawi, H., Xiang, Y.: Deep gene selection method to select genes from microarray datasets for cancer classification. *BMC Bioinformatics* **20**(608), 1–15 (2019). <https://doi.org/10.1186/s12859-019-3161-2>
4. Yuan, F., Lu, L., Zou, Q.: Analysis of gene expression profiles of lung cancer subtypes with machine learning algorithms. *Biochim. Biophys. Acta (BBA)–Mol. Basis of Dis.* **1866**(8), 165822 (2020). doi: <https://doi.org/10.1016/j.bbadis.2020.165822>. ISSN 0925–4439. <https://www.sciencedirect.com/science/article/pii/S0925443920301678>
5. Mohammed, M., Mwambi, H., Mboya, I.B., Elbashir, M.K., Omolo, B.: A stacking ensemble deep learning approach to cancer type classification based on TCGA data. *Sci. Rep.* **11**(1), 15626 (2021). <https://doi.org/10.1038/s41598-021-95128-x>
6. Chen, J.W., Dhahbi, J.: Lung adenocarcinoma and lung squamous cell carcinoma cancer classification, biomarker identification, and gene expression analysis using overlapping feature selection methods. *Sci. Rep.* **11**(1), 13323 (2021). <https://doi.org/10.1038/s41598-021-92725-8>
7. Gao, M., Kong, W., Huang, Z., Xie, Z.: Identification of key genes related to lung squamous cell carcinoma using bioinformatics analysis. *Int. J. Mol. Sci.* **21**(8), 2994 (2020). doi: <https://doi.org/10.3390/ijms21082994>. ISSN 1422-0067. <https://www.mdpi.com/1422-0067/21/8/2994>
8. Song, Z., Zhang, Y., Chen, Z., Zhang, B.: Identification of key genes in lung adenocarcinoma based on a competing endogenous RNA network. *Oncol. Lett.* **21**(1), 60 (2021). <https://doi.org/10.3892/ol.2020.12322>
9. Cancer Genome Atlas Research Network: Comprehensive genomic characterization of squamous cell lung cancers. *Nature* **489**(7417), 519–525 (2012). <https://doi.org/10.1038/nature13385>
10. Hosmer, D., Lemeshow, S.: *Applied Logistic Regression*, 2nd edn. John Wiley & Sons Inc., New York (2000)
11. Long, J.S.: *Regression Models for Categorical and Limited Dependent Variables*. Sage Publications, Thousand Oaks (1997)
12. Cancer Genome Atlas Research, Network: Comprehensive molecular profiling of lung adenocarcinoma. *Nature* **511**(7511), 543–550 (2014). <https://doi.org/10.1038/nature13385>
13. Hou, J., et al.: Gene expression-based classification of non-small cell lung carcinomas and survival prediction. *PLoS ONE* **5**(4), e10312 (2010). <https://doi.org/10.1371/journal.pone.0010312>
14. Breiman, L.: Random forests. *Mach. Learn.* **45**, 5–32 (2001). <https://doi.org/10.1023/A:1010933404324>
15. Karatzoglou, A.: kernlab—An S4 package for kernel methods in R. *Kernel-Based Machine Learning Lab* (2019)
16. Friedman, J.: Regularization paths for generalized linear models via coordinate descent. *Lasso and Elastic-Net Regularized Generalized Linear Models* (2009)



# Approximate Pattern Matching Using Search Schemes and In-Text Verification

Luca Renders<sup>(✉)</sup>, Lore Depuydt<sup>(ID)</sup>, and Jan Fostier<sup>(✉)</sup>

Ghent University - imec, Technologiepark 126, 9052 Ghent, Belgium  
{luca.renders,lore.depuydt,jan.fostier}@ugent.be

**Abstract.** Search schemes enable the efficient identification of all approximate occurrences of a search pattern in a text. Using a bidirectional FM-index, search schemes describe how to explore the search space in such a way that runtime is minimized. Even though in-index matching has an optimal time complexity, relatively expensive random memory access is required for elementary operations on the FM-index. We analyze to what extent in-index matching can be complemented with in-text verification where a candidate occurrence is directly validated in the text using a bit-parallel, pairwise alignment procedure. We find that hybrid in-index/in-text matching can reduce the running time by more than a factor of two, compared to pure in-index matching. We present Columba 1.1, an open-source (AGPL-3.0 license) software tool written in C++ that efficiently implements these ideas. Using a single CPU core, Columba 1.1 can identify, within a maximum edit distance of four, all occurrences of 100 000 Illumina reads (150 bp) in the human reference genome in roughly half a minute. This significantly outperforms existing, state-of-the-art tools.

**Keywords:** Lossless sequence alignment · FM-Index · Bit-parallel alignment · In-text validation

## 1 Introduction

Approximate pattern matching is a well-studied problem in computer science and central to many bioinformatics applications. It involves identifying occurrences of a search pattern  $P$  in a (much) larger text  $T$ . For example, in a typical setting,  $P$  could be a short DNA fragment (a read) and  $T$  a (collection of) reference genome(s). Due to sequencing errors and genetic diversity among individuals, one is often interested in finding *approximate* occurrences of  $P$  in  $T$ .

Historically, *lossy* approximate pattern matching algorithms gained a lot of popularity. Such algorithms rely on heuristics to quickly identify *some* (but not necessarily *all*) approximate matches of  $P$  in  $T$ . By sacrificing some sensitivity, significant performance gains can be obtained. As such, lossy algorithms are used in many state-of-the-art alignment tools such as BLAT [8], BLAST [2], BWA [12], etc. In contrast, in this paper, we focus on *lossless* algorithms which

are guaranteed to retrieve *all* approximate matches of  $P$  in  $T$  under a certain error distance metric. Specifically, the  $k$ -mismatch problem involves identifying all occurrences of  $P$  in  $T$  with up to  $k$  errors. Under the Hamming distance metric, only substitutions are allowed whereas the Levenshtein/edit distance metric allows substitutions, insertions, and deletions. In this work, we focus on the edit distance metric.

Full-text indexes such as suffix trees [5], enhanced suffix arrays [1] and FM-indexes [4] are used within numerous bioinformatics tools [18]. They allow for unidirectional, exact pattern matching, one character at a time, with a runtime proportional to the length of the search pattern and independent of the size of  $T$ . A naive approach to lossless approximate pattern matching would be to explore all possible branches in the index (called *backtracking*) within the maximum allowed Hamming/Levenshtein distance of search pattern  $P$ . This approach has two problems: a) the number of branches to explore increases rapidly with  $k$  and b) the vast majority of branches that are explored eventually turn out not to be matches.

A bidirectional index (such as the affix tree [13], the affix array [24] and the bidirectional FM-index [11]) augments the functionality of its unidirectional counterpart by allowing patterns to be matched in both directions: left-to-right and right-to-left. Using, e.g., a bidirectional FM-index, a query pattern can be searched by starting at any arbitrary position of that pattern and extending the match either to the left or to the right in arbitrary order. More formally, a (partial) match  $P$  can be extended by a single character  $c$  to either  $cP$  or  $Pc$ . In 2009, Lam et al. were the first to note that this added functionality opens up new possibilities for faster lossless approximate matching [11]. Leveraging the classical pigeonhole principle, they partitioned  $P$  into  $k + 1$  parts, from which immediately follows that any approximate occurrence with at most  $k$  errors, must have an exact match with at least one of these parts. By first performing an exact search for one part of  $P$  (which maps to a single branch of the index) and then extending this partial match with an approximate search (backtracking), significant computational gains are obtained. This idea was generalized by Kucherov et al. who introduced the concept of *search schemes* [10]. Informally, search schemes define how a pattern  $P$  is matched using a bidirectional index, such that unsuccessful branches are discarded as quickly as possible and, hence, the runtime is minimized. Kucherov et al. also proposed a number of efficient search schemes with  $k + 1$  and  $k + 2$  parts for up to  $k = 4$  errors. Kianfar et al. [9] further extended this work and used integer linear programming (ILP) to generate additional search schemes for the Hamming distance metric. Additionally, they show that related work on lossless approximate pattern matching by Vroland et al. on  $01^*0$  seeds [26] can also be expressed as search schemes. Therefore, search schemes represent a flexible framework for lossless approximate pattern matching in which a multitude of algorithmic ideas can be expressed.

Recently, we proposed Columba [21], an efficient software tool for lossless approximate pattern matching using arbitrary search schemes. We proposed an algorithm for the dynamic partitioning of search patterns to further reduce the

search space and used an efficient memory layout for the data structures that underlie the FM-index. In this paper, we further build upon this work and we make the following contributions:

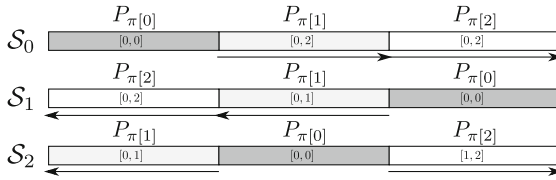
1. We adapted the search schemes by Kucherov et al. with  $k+1$  parts by imposing more stringent lower bounds on the cumulative number of errors in the different parts of the search pattern while maintaining the guarantee that all possible error distributions are covered. These adapted search schemes reduce the runtime by nearly 15%.
2. We adopt the bit-parallel, pairwise alignment algorithm by Hyyrö [7]. This algorithm is used to accelerate edit distance computations during in-index matching. Additionally, it is applied to in-text verification where a candidate occurrence of the search pattern is assessed directly in the text  $T$ . We show that using hybrid in-index matching/in-text verification can reduce the runtime by half compared to using only in-index matching.
3. We developed Columba 1.1, an open-source implementation in standard C++11 in which the above techniques were implemented. We demonstrate that our implementation is several times faster than other state-of-the-art lossless alignment algorithms such as GEM [14] and Bwolo [26] for the task of identifying all occurrences of 150 bp Illumina reads in the human reference genome within an edit distance of  $k = 4$ . We show that Columba 1.1 is faster than BWA in mem mode for  $k = 1, 2$  and 3 and has a similar runtime for  $k = 4$ . Columba 1.1 is available at <https://github.com/biointec/columba> under AGPL-3.0 license.

This paper is organized as follows. In Sect. 2, we briefly describe the (bidirectional) FM-index and search scheme functionality. Section 3 introduces the adapted search schemes that are used throughout this work. In Sects. 4 and 5, we provide the key algorithms for bit-parallel edit distance computations and their application to in-text verification, respectively. Finally, Sect. 6 provides performance benchmarks of Columba as well as existing state-of-the-art tools.

## 2 Preliminaries

### 2.1 Bidirectional FM-Index

In this paper, we use zero-based array indexing, half-open intervals  $[..)$  and standard notation on strings. A text  $T[0, n)$  of size  $n$ , which ends with a unique sentinel character  $\$$  (defined as the lexicographically smallest character), has a Burrows-Wheeler transform  $\text{BWT}[0, n)$ , which is defined as  $\text{BWT}[i] = T[\text{SA}[i] - 1]$  if  $\text{SA}[i] > 0$  and  $\text{BWT}[i] = \$$  otherwise [3]. Here, SA denotes the suffix array of text  $T$ , defined as a permutation over  $\{0, 1, \dots, n-1\}$ , such that  $\text{SA}[i]$  is the starting position of the lexicographically  $i$ -th suffix of  $T$ . To perform exact and approximate matching, we need support for  $\text{occ}(c, i)$  queries on the BWT, that return the number of occurrences of a character  $c$  in the prefix  $\text{BWT}[0, i)$ . This is realized through  $|\Sigma|$  (where  $\Sigma$  denotes the alphabet) bit vectors with



**Fig. 1.** Search scheme for  $k = 2$  errors and 3 parts proposed by Kucherov et al. The parts are processed from darkest to lightest shade of gray. In each part, the lower and upper bound to the cumulative number of errors up to and including that part, are indicated. The arrows indicate the search direction (left-to-right or right-to-left).

constant-time rank support. Exact matching can then be performed by matching character by character from right to left. Consider an interval  $[b, e)$  over the suffix array for which the corresponding suffixes are prefixed by  $P$ . In order to do exact matching backwards, we want to find interval  $[b', e')$  whose corresponding suffixes are prefixed by  $cP$ . This can be computed as follows:  $b' = C(c) + \text{occ}(c, b)$  and  $e' = C(c) + \text{occ}(c, e)$ , where  $C(c)$  denotes the number of characters in  $\text{BWT}[0, n)$  that are smaller than  $c$ . These are pre-computed and stored in a small array of size  $|\Sigma|$ . Since  $\text{occ}$  queries rely only on constant-time rank operations, exact matching of a pattern  $P$  takes  $O(|P|)$  time. The number of occurrences of  $P$  in  $T$  is equal to the size of the interval  $[b, e)$ , i.e.,  $e - b$ . The positions of these occurrences in  $T$  are then found using the suffix array. One can opt to use a sparse version of the suffix array, where  $\text{SA}[i]$  is stored only when  $\text{SA}[i]$  is a multiple of a pre-defined sparseness factor  $s$ . A length- $n$  bit vector  $B$  is stored alongside the sparse suffix array to indicate for each index  $i$  if  $\text{SA}[i]$  is stored. The value  $\text{SA}[i]$  for arbitrary  $i$  can be inferred in  $O(s)$  time. For details of this procedure, we refer to e.g. [20]. The FM-index is a full-text index that comprises a BWT representation and auxiliary tables and that may occupy as little as 2–4 bits of memory per character for DNA sequences [4].

In 2009, the *bidirectional* FM-index was introduced [11]. By also storing  $\text{BWT}_r$ , the Burrows-Wheeler transform of the reverse of  $T$ , and keeping track of both the range  $[b, e)$  over the BWT as well as the range  $[b', e')$  over  $\text{BWT}_r$  in a synchronized manner,  $P$  can be extended backwards (to  $cP$ ) or forwards (to  $Pc$ ). By replacing the ‘occ’ data structure with a so-called ‘Prefix-Occ’ structure, both can be done in  $O(1)$  time [19].

## 2.2 Search Schemes

To perform lossless approximate pattern matching with up to  $k$  errors one needs to explore all the branches of the FM-index that could potentially be matches. Using a naive backtracking approach, an excessive number of unsuccessful branches near the dense root of the search tree will be explored, rendering backtracking computationally unfeasible even for modest values of  $k$ . To alleviate this, Kucherov et al. proposed *search schemes* [10]. We adopt their notation. A pattern  $P$  is partitioned into  $p$  parts  $P_i$  ( $i = 0 \dots p - 1$ ). A search  $\mathcal{S}$  is a

triplet of arrays  $(\pi, L, U)$  of size  $p$ . Here,  $\pi$  is a permutation over  $\{0, \dots, p-1\}$  that defines the order in which the parts  $P_i$  are processed. In order to constitute a valid search scheme,  $\pi$  must satisfy the connectivity property, i.e., a partial match can only be extended in a contiguous manner, either to the left or to the right. The arrays  $L$  and  $U$  respectively define the lower and upper bound to the cumulative number of errors after each part is processed. The core idea of search schemes is that the number of allowed errors is only gradually increased. This significantly reduces the search space near the dense root of the search tree. To cover all possible error distributions over the length of a pattern, multiple *searches* are required that collectively form a search scheme. We denote an error distribution for  $p$  parts and at most  $k$  errors as  $e_0 e_1 \dots e_{p-1}$ , with  $\sum_{i=0}^{p-1} e_i \leq k$ , where  $e_i$  is the number of errors in part  $P_i$ . In order for a search scheme for  $p$  parts and at most  $k$  errors to be valid, all possible error distributions need to be covered by at least one search.

For example, for  $k = 2$  errors, Kucherov et al. proposed a search scheme with three searches:  $\mathcal{S}_0 = (012, 000, 022)$ ;  $\mathcal{S}_1 = (210, 000, 012)$ ;  $\mathcal{S}_2 = (102, 001, 012)$  (see Fig. 1). In the  $\mathcal{S}_0$  search, exact matching is first performed for the leftmost part  $P_0$ . Next, this exact match is extended to the right, thus processing parts  $P_1$  and  $P_2$ , using a backtracking procedure that allows up to two errors. In the  $\mathcal{S}_1$  search, exact matching is first performed for the rightmost part  $P_2$ , and extended to the left by first allowing up to a single error in  $P_1$ , and then two errors in  $P_0$ . Indeed, occurrences of  $P$  with two errors in the middle part were already covered by search  $\mathcal{S}_0$ . Finally, search  $\mathcal{S}_2$  first involves an exact matching of  $P_1$ , which is then extended to the left allowing a single error, and finally to the right with at least one, and at most two errors. This search also explains the need for bidirectional matching functionality. Kucherov et al. [10] and Kianfar et al. [9] proposed search schemes for up to  $k = 4$  errors.

### 3 Adapted Search Schemes

In earlier work [21], we concluded that the search schemes by Kucherov et al. with  $p = k + 1$  parts showed the best performance for the task of identifying occurrences of Illumina reads in the human reference genome under an edit distance constraint. However, it appears that for some searches  $\mathcal{S} = (\pi, L, U)$ , the lower bound array  $L$  can be made more stringent, while maintaining the guarantee that collectively, all searches within the search scheme cover all possible error distributions over a pattern. Recall that when part  $P_i$  has been processed, the cumulative number of errors must be between  $L[i]$  and  $U[i]$ . The benefit of the adapted search schemes is twofold: 1) if fewer error distributions of a search pattern are covered by multiple searches, the number of redundant occurrences decreases, reducing the time to filter them and 2) by making the lower bounds more stringent, the search space that needs to be explored decreases. The original and adapted search schemes are presented in Table 1.

**Table 1.** The original search schemes by Kucherov et al. for  $p = k + 1$  parts and our adapted search schemes for  $k = \{1, 2, 3, 4\}$  errors. Changes are highlighted in bold.

$k$	Original	Adapted
1	(01, 00, 01); (10, 01, 01)	(01, 00, 01); (10, 01, 01)
2	(012, 000, 022); (210, 000, 012); (102, 001, 012)	(012, <b>012</b> , 022); (210, 000, 012); (102, 001, 012)
3	(0123, 0000, 0133); (1023, 0011, 0133) (2310, 0000, 0133); (3210, 0011, 0133)	(0123, <b>0002</b> , 0133); (1023, <b>0113</b> , 0133) (2310, 0000, 0133); (3210, <b>0111</b> , 0133)
4	(01234, 00000, 02244); (43210, 00000, 01344); (10234, 00133, 01334); (01234, 00133, 01334); (32410, 00011, 01244); (21034, 00013, 01244); (10234, 00124, 01244); (01234, 00034, 00444);	(01234, <b>00002</b> , 02244); (43210, 00000, 01344); (10234, <b>01334</b> , 01334); (01234, <b>00334</b> , 01334); (32410, <b>00111</b> , 01244); (21034, <b>00113</b> , 01244); (10234, <b>01224</b> , 01244); (01234, <b>00344</b> , 00444)

### 4 Bit-Parallel Edit Distance Computation

To enable approximate pattern matching, we rely on edit distance computations. The edit distance between two sequences  $S_1$  and  $S_2$  of lengths  $m$  and  $n$ , respectively, can be computed in  $O(mn)$  time using a dynamic programming algorithm. This entails computing an  $(m + 1) \times (n + 1)$  matrix  $D$  such that each element  $D(i, j)$  represents the edit distance between prefix  $S_1[0 \dots i]$  and prefix  $S_2[0 \dots j]$ . The values  $D(i, j)$  are efficiently computed by following recurrence relation:

$$\begin{aligned}
 D(i, 0) &= i; D(0, j) = j && \forall i, j \geq 0 \\
 D(i, j) &= \min \begin{cases} D(i - 1, j - 1) + \delta(S_1[i - 1], S_2[j - 1]) \\ D(i - 1, j) + 1 \\ D(i, j - 1) + 1 \end{cases} && \forall i, j > 0
 \end{aligned}$$

where  $\delta(a, b)$  is 0 if  $a = b$  and 1 otherwise. The oldest description of this algorithm is by Vintsyuk [25] in 1968; it has been independently rediscovered by others (see e.g. [17] and the references therein). Myers [16] improved the time complexity to  $O(mn/w)$ , where  $w$  denotes the computer word size ( $w = 64$  for most CPU architectures). The core idea is to leverage bit-level parallelism to compute multiple values of matrix  $D$  simultaneously. Inspired by Myers work, Hyyrö [6] proposed a slightly more efficient bit-parallel algorithm. We first provide a brief description of this algorithm. Next, we describe our specific adaptations.

#### 4.1 Hyyrö’s Bit-Parallel Algorithm

Adjacent elements within any row or column of matrix  $D$  differ by at most a value of 1, i.e., for all  $i, j$ :  $D(i, j) - D(i, j - 1) \in \{-1, 0, 1\}$  and  $D(i, j) - D(i - 1, j) \in \{-1, 0, 1\}$  (see [15], lemma 3). Similarly, for adjacent elements on a diagonal, it holds that  $D(i, j) - D(i - 1, j - 1) \in \{0, 1\}$ . Rather than computing the values of  $D$  directly, each row  $i$  is encoded by five delta vectors  $VP_i, VN_i, HP_i, HN_i,$



and  $D0_i$ . These delta vectors are stored as bit vectors (i.e., a sequence of 0s and 1s) and are defined as follows:

1. The vertical positive delta vector:  $VP_i[j] = 1 \iff D(i, j) - D(i - 1, j) = 1$
2. The vertical negative delta vector:  $VN_i[j] = 1 \iff D(i, j) - D(i - 1, j) = -1$
3. The horizontal positive delta vector:  $HP_i[j] = 1 \iff D(i, j) - D(i, j - 1) = 1$
4. The horizontal negative delta vector:  $HN_i[j] = 1 \iff D(i, j) - D(i, j - 1) = -1$
5. The diagonal zero delta vector:  $D0_i[j] = 1 \iff D(i, j) - D(i - 1, j - 1) = 0$

The bits  $HP_i[j]$  and  $HN_i[j]$  encode the value  $D(i, j) - D(i, j - 1)$ . The latter equals either 1 (when  $HP_i[j] = 1$ ), -1 (when  $HN_i[j] = 1$ ), or 0 (when both  $HP_i[j] = 0$  and  $HN_i[j] = 0$ ). Similarly,  $VP_i[j]$  and  $VN_i[j]$  encode the value  $D(i, j) - D(i - 1, j)$ . Therefore, because  $D(0, 0)$  is known (often 0), all other values  $D(i, j)$  can be inferred from the delta vectors.

The key advantage of using the delta vectors is that they can be computed in a bit-parallel manner as shown in Algorithm 1:

---

**Algorithm 1:** Bit-parallel computation of the delta vectors at row  $i$  from those at row  $i - 1$

---

```

D0i ← (((MS1[i-1] & HPi-1) + HPi-1) ^ HPi-1) | MS1[i-1] | HNi-1
VPi ← HNi-1 | ~(D0i | HPi-1)
VNi ← D0i & HPi-1
HPi ← (VNi << 1) | ~(D0i | (VPi << 1))
HNi ← (D0i & (VPi << 1))
    
```

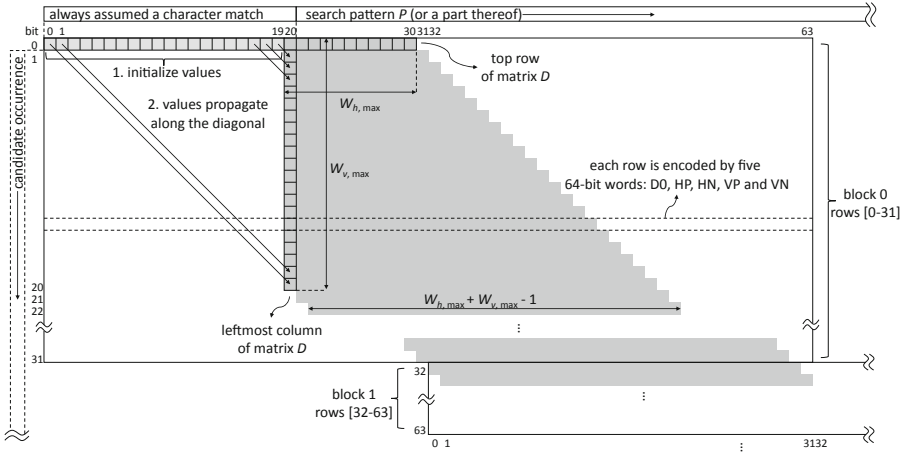
---

Here, the symbols  $\&$ ,  $|$ ,  $\wedge$ ,  $\sim$  and  $\ll$  respectively denote the bitwise AND, OR, XOR, NOT and left shift operators.  $M_{S_1[i-1]}$  is a match vector (again a bit vector) that indicates which positions in  $S_2$  match character  $S_1[i - 1]$ . The four match vectors  $M_c$  (with  $c \in \{A, C, G, T\}$ ) are pre-computed. For the exact details of Algorithm 1, we refer to [6].

### 4.2 Bit-Parallel Banded Alignment

In the context of this work, we want to identify approximate occurrences within a distance of at most  $k$  edit operations of search pattern  $P$ . Therefore, computations can be restricted to those elements  $D(i, j)$  for which  $|i - j| \leq k$ , i.e., within a band along the diagonal. Each row (or column) of matrix  $D$  thus contains at most  $2k + 1$  values to compute. For this problem of banded alignment, Hyyrö proposed a bit-parallel algorithm [6]. Our implementation is heavily influenced by these ideas but uses a different layout of bit vectors. It is described below.

The global layout of the banded dynamic programming matrix  $D$  is depicted in Fig. 2. Search pattern  $P$  is the ‘horizontal’ sequence while candidate occurrence  $O$  is the ‘vertical’ sequence. The FM-index spells out candidate occurrences character by character, therefore, we leverage bit-parallel computations at the level of rows of  $D$ . During in-index searching, candidate occurrences are generated by a depth-first exploration of the search tree. To support backtracking, the delta vectors of each row are kept in a stack data structure.



**Fig. 2.** Layout of the banded dynamic programming matrix  $D$  as 64-bit words.

Our implementation can compute edit distance values up to  $k = 10$  for sequences of arbitrary length. Because  $k$  is sufficiently small, a single 64-bit word can be used to represent a delta vector and all computations per row are done in  $O(1)$  time. Support for larger values of  $k$  could easily be achieved by representing a delta vector by multiple words, at the cost of some loss of performance. Rows are grouped into *blocks* of 32 rows each. At each next block, the delta vectors are shifted by 32 bit positions such that they overlap all relevant values of the banded dynamic programming matrix (gray-shaded cells in Fig. 2). For each block, four match vectors  $M_c$  (with  $c = \{A, C, G, T\}$ ) are pre-computed to indicate character matches between  $c$  and the overlapping positions of  $P$ . At each row  $i$ , we also keep track of the value  $D(i, i)$ . Using the  $D0_i$  delta vector,  $D(i, i)$  can easily be computed from  $D(i - 1, i - 1)$ . The knowledge of  $D(i, i)$  and the  $HP_i$  and  $HN_i$  delta vectors allows for the computation of any value  $D(i, j)$ . By using population count (‘popcount’) instructions, this can be achieved in  $O(1)$  time. Finally, we adopted Hyr ro’s algorithm to evaluate in a bit-parallel manner whether all values on a row exceed the maximum edit distance threshold  $k$ . This is important to signal the backtracking algorithm that the current candidate occurrence  $O$  should no longer be extended and that the search procedure should backtrack and explore a different branch of the search tree. For details on this algorithm, we refer to [7].

### 4.3 Matrix Initialization

Traditionally, the first row and column of matrix  $D$  are initialized with gap penalties (i.e.,  $D(i, 0) = i$  and  $D(0, j) = j$ ) in the case of global alignment, or with zero values (i.e.,  $D(i, 0) = 0$  and/or  $D(0, j) = 0$ ) in case of semi-global alignment. For our use case of search schemes, we need to be able to initialize the leftmost column of  $D$  with  $2k + 1$  arbitrary values. Indeed, using search

schemes, search pattern  $P$  is matched part by part. Therefore, assuming left-to-right matching, when matching part  $P_i$ , the first column of  $D$  should be initialized with the values from the last column of the matrix of part  $P_{i-1}$  in order to continue the alignment.

In the bit-parallel implementation, the initialization of the first row of  $D$  is straightforward: we set the appropriate value for  $D(0, 0)$  (e.g.,  $D(0, 0) = 0$ ) and encode the other values  $D(0, j)$  using the  $HP_0$  and  $HN_0$  delta vectors. For example, to encode  $D(0, j) = j$ , we set  $HP_0[j] = 1$  and  $HN_0[j] = 0$  for  $j = 1 \dots k$ .

To initialize the first column of  $D$  with arbitrary values, we append dummy columns with a ‘negative’ column index to  $D$  (illustrated in a lighter shade of gray in Fig. 2). Again, we use the  $HP_0$  and  $HN_0$  delta vectors to encode the part of the first row of  $D$  with negative column indexes such that  $D(0, -i)$  equals the desired value for  $D(i, 0)$ . By always assuming a character match at negative column indexes, each value  $D(0, -i)$  will effectively propagate along a diagonal and ultimately set  $D(i, 0)$  to its correct value. This is easily achieved by setting 1-bits in the corresponding part of  $M_c$  for all  $c = \{A, C, G, T\}$ . Even in the presence of backtracking, the elements  $D(i, 0)$  will always be computed correctly. Because the computations for the negative column indexes are handled within the same 64-bit word as the regular column indexes, this procedure imposes no computational overhead.

Because we support a maximum allowed edit distance of 10, we require at most  $W_{h, \max} = 11$  elements at the top row of  $D$  (e.g., to encode the values  $\{0, 1, 2, \dots, 10\}$ ) and at most  $W_{v, \max} = 21$  elements at the leftmost column of  $D$  (e.g., to encode the values  $\{10, \dots, 1, 0, 1, \dots, 10\}$ ). Thus, the parts of the delta vectors that *could* contain relevant values are indicated in a darker shade of gray in Fig. 2. Depending on the use-case (the actual allowed edit distance  $k \leq 10$ , and how precisely matrix  $D$  is initialized) only a subset of these cells will effectively contain relevant data.

## 5 In-Text Verification

In principle, search schemes rely purely on in-index matching: using the bidirectional FM-index, candidate occurrences  $O$  of a search pattern  $P$  are spelled character by character. Extending a candidate occurrence by a single character ultimately translates into rank operations on bit vectors. Collectively, these rank operations lead to a random memory access pattern. The expression *random memory access* refers to the fact that the memory access pattern is unpredictable, and hence, will suffer from a large number of cache misses. Therefore, extending a candidate occurrence by a character is a relatively expensive operation: Pockrandt et al. estimated at least 100 CPU clock cycles per character [20].

At all times during the spelling of a candidate occurrence  $O$ , a range  $[b, e)$  over the suffix array is maintained that refers to the starting positions of each instance of  $O$  in  $T$ . Thus, at any point, the size of the range  $e - b$  corresponds to the number of times  $O$  occurs in  $T$ . This number of instances decreases monotonically when more characters are added to  $O$ . When the value  $e - b$  becomes small, it can be

beneficial to abandon the in-index matching procedure and to verify each of the instances of  $O$  directly in  $T$  using the previously described pairwise alignment procedure. As detailed in Sect. 4, pairwise alignment can be performed efficiently using bit-parallel techniques in a cache-friendly manner. In contrast, when the value  $e - b$  is large, in-index matching is more computationally advantageous, because all instances of  $O$  in  $T$  are handled simultaneously by the FM-index.

In our implementation, part  $P_{\pi[0]}$  is always matched using the FM-index. In practice, matching  $P_{\pi[0]}$  always entails an exact pattern matching procedure (see search schemes in Table 1). From that point onwards, whenever the value  $e - b$  becomes smaller than or equal to a pre-defined threshold  $t$  (referred to as the ‘tipping point’), candidate occurrence  $O$  is no longer extended using the index and the search procedure switches to in-text verification. When  $O$  has been fully evaluated, the search procedure will backtrack and explore other candidate occurrences, again using the FM-index.

This idea of hybrid in-index matching/in-text verification within the context of search schemes has been explored previously by Pockrandt et al. for the Hamming distance metric. The authors report speed-ups between  $1.6\times$  and  $2.1\times$  and an optimal tipping point of 25 [20]. Performing in-text verification for the edit distance metric is more complex because 1) pairwise alignment is computationally more expensive and thus needs to be highly optimized to have overall performance gains; 2) the precise start and end positions of each approximate occurrence of  $P$  in  $T$  are not known in advance. To this end, the bit-parallel alignment algorithm from Sect. 4 is easily modified to support semi-global alignment.

## 6 Results and Discussion

All benchmarks were performed using a dataset of 100 000 Illumina NovaSeq 6000 reads (150 bp), randomly sampled from a larger whole genome sequencing dataset (accession no. SRR9091899). We identified all approximate read occurrences up to an edit distance of  $k = \{1, 2, 3, 4\}$  on both strands of the human reference genome (GRCh38) [22]. We recall that we consider only lossless algorithms that are guaranteed to report all occurrences. We replaced non-ACGT characters in the reference genome (e.g., Ns) by a randomly chosen nucleotide. The different chromosomes were concatenated into a single string. As such, a read can be mapped across the borders of adjacent chromosomes. Such spurious matches can easily be filtered during post-processing.

All results were obtained using a single core of a 32-core Intel® Xeon® E5-2698 v3 CPU running at a base clock frequency of 2.30 GHz. To quantify variability in runtime, each benchmark run was repeated 20 times. We report both the average wall clock time as well as the standard deviation. Redundant occurrences (as defined in [21]) were filtered.

**Table 2.** Comparison of the original search schemes by Kucherov et al. and our adapted search schemes, for different values of the maximum allowed edit distance  $k$ . In both cases, 100 000 Illumina reads of length 150 bp are mapped to both strands of the human reference genome.

Search scheme	Wall clock time $\pm$ SD	No. of nodes visited (search space)	No. of redundant occurrences
$k = 2$ , unique occurrences = 676 528, reads mapped 90.5%			
Original	15.91 $\pm$ 1.58 s	62 035 887	267 541
Adapted	14.73 $\pm$ 1.44 s (-7.4%)	57 263 477 (-7.7%)	264 671 (-1.1%)
$k = 3$ , unique occurrences = 1 416 632, reads mapped 93.1%			
Original	30.89 $\pm$ 1.80 s	128 708 469	719 576
Adapted	26.82 $\pm$ 0.60 s (-13.2%)	116 965 983 (-9.1%)	648 817 (-9.8%)
$k = 4$ , unique occurrences = 2 579 745, reads mapped 94.8%			
Original	72.07 $\pm$ 2.54 s	364 385 491	1 492 806
Adapted	61.35 $\pm$ 0.59 s (-14.9%)	305 476 323 (-16.2%)	1 420 668 (-4.8%)

## 6.1 Original Versus Adapted Search Schemes

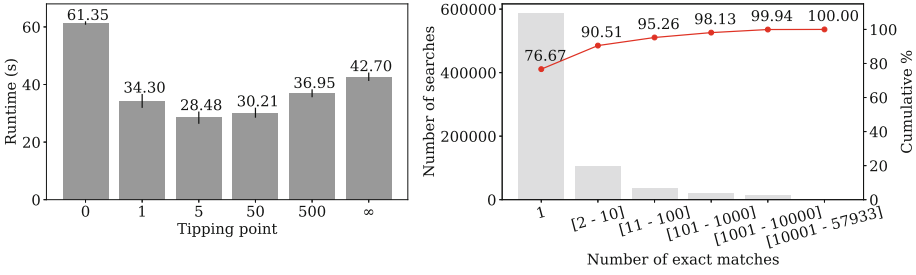
In Table 2, the original and adapted search schemes (as defined in Table 1) are compared for edit distance values of  $k = \{2, 3, 4\}$  as for  $k = 1$ , both search schemes are identical. We report the average runtime and standard deviation on a single CPU core and the number of nodes visited in the search tree. The latter equals the number of times a partial match is extended by a single character  $c$  (in either direction). In practice, this involves expensive random memory access that largely determines the runtime. It is therefore a clear indication of intrinsic performance, regardless of the quality of implementation. It is clear that both in the size of search space (number of nodes visited) and runtime the adapted search schemes are superior. This is no surprise, as the adapted search schemes have tighter bounds and thus reduce the search space.

Table 2 also reports the total number of unique and redundant (filtered out) occurrences for the different values of  $k$ . Because search schemes are lossless, the number of unique occurrences does not differ between the original and adapted search scheme. Clearly, the tighter lower bounds also reduce the number of redundant occurrences (i.e., occurrences reported by multiple searches in the search scheme).

Finally, Table 2 reports the fraction of reads that have at least one occurrence in the reference genome ('reads mapped'), for the different values of  $k$ .

## 6.2 In-Index Versus In-Text Verification

We compared the runtime for matching 100 000 Illumina patterns to both strands of the human reference genome with up to  $k = 4$  edit operations for different values of the tipping point  $t = 0, 1, 5, 50, 500$  and  $\infty$ . A value of  $t = 0$  means that all patterns are entirely matched using the FM-index and that no in-text



**Fig. 3.** Left: the runtime for mapping 100 000 Illumina reads of length 150 bp to both strands of the human reference genome ( $k = 4$ ) as a function of the tipping point  $t$ . Right: histogram of the number of matches for part  $P_{\pi[0]}$  across all searches.

verification is performed whereas  $t = \infty$  denotes that after the initial matching of the first part  $P_{\pi[0]}$ , all candidate occurrences are verified directly in  $T$  and that no further in-index extension takes place. For the intermediate tipping point values, the search procedure switches to in-text verification when  $e - b \leq t$ .

Figure 3 (left) shows the runtime as a function of tipping point  $t$ . Clearly, using purely in-index matching shows the worst performance for this particular dataset. This is because in-index matching involves expensive random memory access in the FM-index for each character that is added to a candidate occurrence. Switching to in-text verification when there is only a single candidate occurrence in  $T$  ( $t = 1$ ) reduces runtime by almost half. This is because bit-parallel, pairwise alignment between the appropriate substring of  $T$  and  $P$  can be performed very efficiently. This effect increases with larger tipping point values and for  $t \approx 5$ , runtime is minimized. For larger tipping point values ( $t \geq 50$ ), the increasing overhead of suffix array lookup operations and pairwise alignments associated with in-text verification (that often turn out to be unsuccessful) dominates the gains. Remarkably, for this dataset, never performing in-index extension beyond the exact matching of the first part  $P_{\pi[0]}$  ( $t = \infty$ ) is still significantly faster than pure in-index matching ( $t = 0$ ). For  $t = \infty$ , the matching process degenerates to a very simple procedure: exact pattern matching of part  $P_{\pi[0]}$  followed by in-text verification of each of the candidate occurrences. For our dataset, the largest suffix array range size encountered was 57 933. This range was encountered for a single read for which  $P_{\pi[0]}$  consists of 29 consecutive characters A.

Collectively over all reads, a tipping point  $t$  between 2 and 10 yields the best performance. Within this range and for our dataset, the runtime is largely insensitive to the precise choice of  $t$  (data not shown). Only for larger values of the tipping point ( $t \geq 10$ ), we again observe an increase in runtime. For other values of  $k$ , a similar conclusion is reached: hybrid in-index matching/in-text verification reduces runtime by 38.43% for  $k = 1$ , 45.24% for  $k = 2$  and 51.30% for  $k = 3$ .

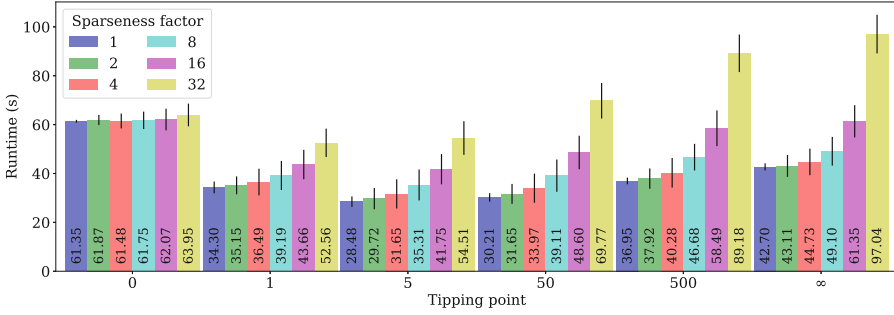
**Breakdown of Reads.** The search scheme for  $k = 4$  errors consists of eight searches (see Table 1). Therefore, for the task of identifying all approximate occurrences of 100 000 reads on both strands of the reference genome, 1 600 000 searches are executed in total. For more than half of these searches (834 198), the first part  $P_{\pi[0]}$  has no exact match in  $T$  and, hence, the search will immediately be terminated. This is no surprise, as most reads have approximate occurrences on only one strand of the reference genome. For the remaining 765 802 searches, Fig. 3 (right) shows a breakdown as a function of the number of (exact) occurrences of part  $P_{\pi[0]}$ . Remarkably, 76.67% (587 103) of those searches yield only a single occurrence in  $T$  for  $P_{\pi[0]}$ . In other words, for most reads, matching only a single part of  $P$  already suffices to point to a unique position in  $T$ . For such cases, in-text verification of that sole candidate occurrence outperforms a further in-index character-by-character extension. This explains the large performance difference between tipping point values  $t = 0$  and  $t = 1$ . Additionally, 13.84% (106 022) of the searches yield between 2 and 10 occurrences in  $T$  for part  $P_{\pi[0]}$ . Also for these cases, in-text verification at each of these candidate positions in  $T$  is superior to in-index matching.

In contrast, only a relatively small fraction of 9.49% (72 677) of the searches deal with patterns for which  $P_{\pi[0]}$  has more than 10 occurrences in  $T$ . In certain cases, this number of instances is vast. For example, 14 329 searches yield more than 1 000 instances of  $P_{\pi[0]}$  in  $T$ , seven of which amount to more than 50 000 instances. The latter all correspond to low-complexity poly-A/T or poly-CA/GT patterns which are highly repeated in the human genome. Here, in-index matching has a clear advantage as all repeated candidate occurrences are handled simultaneously by the FM-index.

We conclude that in-text verification is beneficial for those searches for which the number of occurrences of  $P_{\pi[0]}$  in  $T$  (and hence, the number of candidate occurrences of  $P$  itself), is limited ( $\leq 10$ ). For our dataset, this holds for roughly 90% of the searches. In contrast, the remaining searches (10%) deal with search patterns with many potential occurrences in  $T$ , a task which is best performed using in-index matching and the search scheme. We find that these ‘difficult’ searches, although limited in number, account for roughly two-thirds of the total runtime. In total, these complex searches account for 96.0% of unique matches over the entire dataset.

**SA Space-Time Tradeoff.** In-text verification requires a lookup operation in the suffix array (SA) to retrieve, for each candidate occurrence, its position in  $T$ . The number of candidate occurrences for which in-text verification is performed, and hence, the number of required lookup operations in the SA, increases with higher values of the tipping point  $t$ .

To reduce the memory footprint of the FM-index, a sparse version of the SA is often used. In our implementation, every  $s$ -th suffix of the SA is stored, where  $s$  denotes the sparseness factor, i.e.,  $SA[i]$  is stored if and only if  $SA[i] \bmod s = 0$ . It is well-known that a suffix at an arbitrary index  $i$  can then be inferred in  $O(s)$  time [20]. Thus, the sparseness factor  $s$  controls the space-time tradeoff. As each



**Fig. 4.** Runtime for mapping 100 000 Illumina reads (150 bp) to both strands of the human reference genome as a function of the tipping point  $t$  and sparseness factor  $s$ .

in-text verification requires a lookup operation in the SA, a larger sparseness factor  $s$  will diminish the gains in the runtime of in-text verification.

Figure 4 shows the runtime for different sparseness factors  $s$  and tipping points  $t$ . The results for  $s = 1$  (dense SA) are identical to those of Fig. 3 (left). For all values of  $t$ , the runtime increases with the sparseness factor  $s$ , as lookup operations in the SA become more expensive. For  $t = 0$ , the increase in runtime from  $s = 1$  to  $s = 32$  is limited to only 4.2% whereas for  $t = \infty$ , the runtime more than doubles.

Therefore, especially for larger values of the sparseness factor  $s$ , the tipping point  $t$  should not be set to (too) high values for good performance. In our experience, up to  $s = 16$ , a choice of  $t \approx 5$  appears appropriate. For sparseness factors of  $s = 32$  and larger, a tipping point of  $t = 1$  or  $t = 2$  showed the best performance.

### 6.3 Comparison to State-of-the-Art Tools

In earlier work [21], we presented Columba 1.0, a fast software implementation for lossless approximate pattern matching using search schemes. Columba 1.0 implements the ideas outlined in [21] such as a cache-friendly BWT representation and dynamic partitioning of search schemes.

The techniques described in this paper (bit-parallel edit distance computations, in-text verification, and the adapted search schemes) are implemented in Columba 1.1. In this section, we benchmark Columba 1.1 against state-of-the-art lossless pattern matching tools, including Columba 1.0. We use the adapted search schemes proposed in Table 1, a tipping point  $t = 5$  and a SA sparseness factor  $s = 1$  (dense SA).



**Table 3.** Runtime comparison of state-of-the-art lossless alignment tools, with the exception of BWA in ‘mem’ mode, which is a lossy alignment algorithm.

Tool	Language	Reference	$k = 1$	$k = 2$	$k = 3$	$k = 4$
Columba 1.1 <sup>a</sup>	C++	This paper	5.15 ± 0.44 s	8.66 ± 1.00 s	13.06 ± 1.31 s	28.48 ± 2.13 s
Columba 1.0 <sup>b</sup>	C++	[21]	7.05 ± 0.16 s	13.10 ± 0.26 s	25.62 ± 0.33 s	67.75 ± 0.51 s
BWA <sup>c</sup>	C	[12]	14.73 ± 0.23 s	133.11 ± 2.39 s	1454.40 ± 24.64 s	DNC (> 3h)
Bwolo	C++	[26]	12.53 ± 0.55 s	25.24 ± 0.86 s	63.67 ± 1.32 s	189.78 ± 2.25 s
GEMv3 <sup>d</sup>	C	[14]	9.0 ± 1.5 s	18.6 ± 2.4 s	38.5 ± 4.6 s	84.6 ± 4.9 s
Yara v0.9.11 <sup>e</sup>	C++	[23]	4.49 ± 0.13 s	21.00 ± 0.34 s	81.90 ± 0.84 s	537.26 ± 7.65 s
BWA mem (lossy)	C	[12]	32.42 ± 0.67 s (independent of $k$ )			

<sup>a</sup> -e  $k$  -i 5 -ss ../search\_schemes/kuch.k+1.adapted/

<sup>b</sup> -e  $k$  -ss ../search\_schemes/kuch.k+1/

<sup>c</sup> aln -N -n  $k$  -i 0 -l 150 -k  $k$

<sup>d</sup> -t 1 -e [ $k$ ] -s [ $k$ ] -alignment-model edit -mapping-mode complete -M 1000

<sup>e</sup> -e [ $k$ ] -s [ $k$ ] -y full -t 1

In Table 3, we compare the performance of Columba 1.1 to Columba 1.0, Bwolo [26], GEM [14], Yara [23] and BWA [12] in all-mapping mode. Note that Columba 1.0 and Bwolo do not report the CIGAR string of the alignments in their output whereas the other tools do (including Columba 1.1). For the GEM aligner, not all occurrences could be reported as the tool failed when using the `all` parameter. Therefore, GEM was configured to report at most 1000 occurrences per read.

Columba 1.1 outperforms Columba 1.0 for all values of  $k$ , even though Columba 1.0 does not compute the CIGAR string. Gains are achieved through the tighter lower bounds as specified in the adapted search schemes and bit-parallel, in-text verification. Clearly, these gains outweigh the extra computations required to generate the CIGAR string.

Both Columba 1.1 and 1.0 outperform all other lossless alignment tools for  $k \geq 2$ . For  $k = 1$ , both are slightly slower than Yara. This is likely due to the overhead imposed by the use of the bidirectional FM-index, whereas Yara relies on a unidirectional index. For  $k \geq 2$ , Columba 1.1 is at least twice as fast as other tools. For  $k = 4$ , Columba 1.1 appears roughly 3× faster than GEM, 6× faster than Bwolo, and even 18× times faster than Yara. Clearly, BWA was not designed to run in lossless mode for higher values of  $k$ .

We also compare Columba 1.1 with BWA in (lossy) mem mapping mode. In mem mode, BWA does not require a maximum number of errors  $k$  to be specified and it will typically report only a single candidate alignment position for each read. Note that the time to read the index structure from disk is included in BWA’s runtime, which is not the case for Columba 1.1. Also note that BWA outputs SAM format and is able to handle paired-end reads, which is not the case for Columba 1.1. Columba 1.1 appears faster than BWA for  $k = 1, 2$  and 3. For  $k = 4$ , the runtime of Columba 1.1 is similar to that of BWA. This indicates that the performance gap between lossless and lossy alignment tools is closing for practical bioinformatics applications such as read mapping.

## 7 Conclusion

We introduced Columba 1.1, a tool for lossless approximate pattern matching using search schemes under the edit distance metric. Columba 1.1 implements hybrid in-index matching/in-text verification using a bit-parallel, pairwise alignment algorithm. It is demonstrated that this technique reduces runtime by more than a factor of two, compared to pure in-index matching. We provided an analysis of the effect of in-text verification for different types of reads. For reads with a limited number of occurrences, switching to in-text verification greatly reduces the runtime. In contrast, for reads with many potential occurrences, in-index matching appears the better option. We showed that the use of a sparse suffix array somewhat diminishes the performance gains of using in-text verification. Nevertheless, for all practical values of the suffix array sparseness factor, in-text verification proves beneficial. Finally, Columba 1.1 shows superior performance to state-of-the-art lossless aligners.

**Acknowledgments.** Luca Renders and Lore Depuydth are funded by the Research Foundation - Flanders (FWO), through a PhD Fellowship SB (1SE7822N) and a PhD Fellowship FR (1117322N), respectively.








## References

1. Abouelhoda, M.I., Kurtz, S., Ohlebusch, E.: Replacing suffix trees with enhanced suffix arrays. *J. Discrete Algorithms* **2**(1), 53–86 (2004). [https://doi.org/10.1016/S1570-8667\(03\)00065-0](https://doi.org/10.1016/S1570-8667(03)00065-0)
2. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic local alignment search tool. *J. Mol. Biol.* **215**(3), 403–10 (1990)
3. Burrows, M., Wheeler, D.: A block-sorting lossless data compression algorithm. Technical report, Digital Systems Research Center (1994)
4. Ferragina, P., Manzini, G.: Opportunistic data structures with applications. In: *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pp. 390–398, February 2000. <https://doi.org/10.1109/SFCS.2000.892127>
5. Gusfield, D.: *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, Cambridge (2007)
6. Hyvärinen, H.: A bit-vector algorithm for computing Levenshtein and Damerau edit distances. *Nord. J. Comput.* **10**(1), 29–39 (2003)
7. Hyvärinen, H., Navarro, G.: Faster bit-parallel approximate string matching. In: Apostolico, A., Takeda, M. (eds.) *CPM 2002*. LNCS, vol. 2373, pp. 203–224. Springer, Heidelberg (2002). [https://doi.org/10.1007/3-540-45452-7\\_18](https://doi.org/10.1007/3-540-45452-7_18)
8. Kent, W.J.: BLAT - the BLAST-like alignment tool. *Genome Res.* **12**(4), 656–64 (2002)
9. Kianfar, K., Pockrandt, C., Torkamandi, B., Luo, H., Reinert, K.: FAMOUS: fast approximate string matching using optimum search schemes. *CoRR* (2017). <http://arxiv.org/abs/1711.02035>
10. Kucherov, G., Salikhov, K., Tsur, D.: Approximate string matching using a bidirectional index. In: Kulikov, A.S., Kuznetsov, S.O., Pevzner, P. (eds.) *CPM 2014*. LNCS, vol. 8486, pp. 222–231. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-07566-2\\_23](https://doi.org/10.1007/978-3-319-07566-2_23)

11. Lam, T., Li, R., Tam, A., Wong, S., Wu, E., Yiu, S.: High throughput short read alignment via bi-directional BWT. In: IEEE International Conference on Bioinformatics and Biomedicine, pp. 31–36, December 2009. <https://doi.org/10.1109/BIBM.2009.42>
12. Li, H., Durbin, R.: Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* **25**(14), 1754–1760 (2009). <https://doi.org/10.1093/bioinformatics/btp324>
13. Maaß, M.G.: Linear bidirectional on-line construction of affix trees. In: Giancarlo, R., Sankoff, D. (eds.) CPM 2000. LNCS, vol. 1848, pp. 320–334. Springer, Heidelberg (2000). [https://doi.org/10.1007/3-540-45123-4\\_27](https://doi.org/10.1007/3-540-45123-4_27)
14. Marco-Sola, S., Sammeth, M., Guigó, R., Ribeca, P.: The gem mapper: fast, accurate and versatile alignment by filtration. *Nat. Methods* **9**(12), 1185–1188 (2012). <https://doi.org/10.1028/nmeth.2221>
15. Masek, W.J., Paterson, M.: A faster algorithm computing string edit distances. *J. Comput. Syst. Sci.* **20**(1), 18–31 (1980)
16. Myers, G.: A fast bit-vector algorithm for approximate string matching based on dynamic programming. In: Farach-Colton, M. (ed.) CPM 1998. LNCS, vol. 1448, pp. 1–13. Springer, Heidelberg (1998). <https://doi.org/10.1007/BFb0030777>
17. Navarro, G.: A guided tour to approximate string matching. *ACM Comput. Surv.* **33**(1), 31–88 (2001). <https://doi.org/10.1145/375360.375365>
18. Navarro, G., Baeza-Yates, R.: A hybrid indexing method for approximate string matching. *J. Discrete Algorithms* **1**(1), 205–239 (2000)
19. Pockrandt, C., Ehrhardt, M., Reinert, K.: EPR-dictionaries: a practical and fast data structure for constant time searches in unidirectional and bidirectional FM-indices (2016)
20. Pockrandt, C.M.: Approximate string matching: improving data structures and algorithms. Ph.D. thesis, Freien Universität Berlin (2019). <https://doi.org/10.17169/refubium-2185>
21. Renders, L., Marchal, K., Fostier, J.: Dynamic partitioning of search patterns for approximate pattern matching using search schemes. *iScience* **24**(7), 102687 (2021). <https://doi.org/10.1016/j.isci.2021.102687>
22. Schneider, V., et al.: Evaluation of GRCh38 and de novo haploid genome assemblies demonstrates the enduring quality of the reference assembly. *Genome Res.* **27** (2017). <https://doi.org/10.1101/gr.213611.116>
23. Siragusa, E.: Approximate string matching for high-throughput sequencing. Ph.D. thesis (2015)
24. Strothmann, D.: The affix array data structure and its applications to RNA secondary structure analysis. *Theoret. Comput. Sci.* **389**(1), 278–294 (2007). <https://doi.org/10.1016/j.tcs.2007.09.029>
25. Vintsyuk, T.K.: Speech discrimination by dynamic programming. *Cybernetics* **4**(1), 52–57 (1968). <https://doi.org/10.1007/bf01074755>
26. Vroland, C., Salson, M., Bini, S., Touzet, H.: Approximate search of short patterns with high error rates using the 01\*0 lossless seeds. *J. Discrete Algorithms* **37**, 3–16 (2016). <https://doi.org/10.1016/j.jda.2016.03.002>



# KFinger: Capturing Overlaps Between Long Reads by Using Lyndon Fingerprints

Paola Bonizzoni<sup>1</sup> , Alessia Petescia<sup>1</sup> , Yuri Pirola<sup>1</sup> , Raffaella Rizzi<sup>1</sup>  ,  
Rocco Zaccagnino<sup>2</sup> , and Rosalba Zizza<sup>2</sup> 

<sup>1</sup> Dip. di Informatica, Sistemistica e Comunicazione, University of Milano-Bicocca,  
viale Sarca 336, 20126 Milan, Italy

{paola.bonizzoni,yuri.pirola,raffaella.rizzi}@unimib.it,  
a.petescia@campus.unimib.it

<sup>2</sup> Dip. di Informatica, University of Salerno,  
via Giovanni Paolo II 132, 84084 Fisciano, Italy

{rzaccagnino,rzizza}@unisa.it

**Abstract.** Detecting common regions and overlaps between DNA sequences is crucial in many Bioinformatics tasks. One of them is genome assembly based on the use of the overlap graph which is constructed by detecting the overlap between genomic reads. When dealing with long reads this task is further complicated by the length of the reads and the high sequencing error rate. This paper proposes a novel alignment-free method for detecting the overlaps in a set of long reads which exploits a signature (called *fingerprnt*) of reads built from a factorization of the read based on the notion of Lyndon words. The method has been implemented in the tool **KFinger** and tested over a simulated and a real PacBio HiFi dataset of genomic reads; its results have been compared with the well-known aligner **Minimap2**. **KFinger** is available at <https://github.com/AlgoLab/kfinger>.

**Keywords:** Lyndon word · Factorization · Fingerprint · Overlap graph · Long reads

## 1 Introduction

Lyndon word is a concept of combinatorics on words and a well-known notion in Bioinformatics [1, 2], where it has been used to find short motifs [3] and more recently in the notion of the extended BWT [4]. Most notably, a recent work suggests that Lyndon factorizations can be used to detect overlaps between reads [5], which is the fundamental task to build the overlap graph in genome assembly.

---

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 872539.

Note that a factorization (as notion of combinatorics on words) expresses a string as a concatenation of factors and factors in a Lyndon factorization are Lyndon words. Lyndon factorization [1, 6] is one of the most well-known factorizations and has two main properties: (i) it is unique for a given string and (ii) can be computed in linear time. Moreover, it satisfies the following crucial property, which is the foundation of our proposed method: two strings sharing a common overlap also share a set of consecutive common factors in their factorizations [5].

Detecting the overlap between sequences is the fundamental step in de-novo assembly based on the Overlap-Layout-Consensus (OLC) strategy [7], which is the main approach used for assembling long reads [8, 9]. Since unfortunately such reads are long and error-prone, detecting overlaps is often a bottleneck from a computational point of view, mainly when a pairwise comparison is adopted, due to the fact that long reads have high sequencing errors and contain repetitive regions. Several methods for discovering overlaps in long reads, which are based on a representation of the input reads, are present in literature, achieving good performance in terms of computation time and accuracy. For example, [10] proposes an algorithm combining minimizers and MinHash algorithm [11] for mapping long reads to a reference database; sourmash [12] and MHAP [8] use MinHash algorithm (MHAP relies on  $k$ -mers); sourmash estimates sequence similarity between very large data sets whereas MHAP is a tool for discovering overlaps between long reads and is used by Canu assembler [13]. Minimap2 [14] is an aligner of DNA or mRNA long reads against a large reference database and uses minimizers.

We propose an alignment-free approach for discovering the overlaps in a set of noisy long reads, exploiting a compact representation (or signature) given by the sequence of lengths of the Lyndon factors (instead of the factors themselves) in Lyndon factorizations. The sequence of factor lengths, called *fingerprint*, has been first introduced in [15] as a mean to discover common regions between reads and applied for classifying RNA-Seq reads by origin gene. Read fingerprints provide a compact representation of the reads and unexpectedly they are effective in preserving sequence similarities, thus being extremely useful in an alignment-free approach for discovering similarities. The main idea is that a factorization of a read is computed while reading the reads and the factorization splits the reads based on their content in terms of Lyndon-words: we keep the sequence of the distances between consecutive splitting positions (that is, the sequence of the factor lengths) to use as read fingerprint (read signature). The  $k$ -mers of a fingerprint (called  $k$ -fingers) are the sub-pieces able to capture the similarity regions between the reads in a more flexible way with respect to the  $k$ -mers of a sequence: indeed the length  $k$  of a  $k$ -mer is fixed. Furthermore, fingerprints (numerical sequences) are shorter than the represented nucleotide sequences and we expect that they are also resilient to errors occurring in long reads and common  $k$ -fingers can be discovered. In the paper we show that  $k$ -fingers provide anchors for computing common regions between reads of an input set  $S$  and present an algorithm performing factorization of the reads in  $S$  and (next) a linear scanning of the read signatures (or fingerprints); by hashing the  $k$ -fingers,

the common regions, shared by the currently processed read  $s$  and all the reads previously considered, are computed in  $O(LN)$ , where  $L$  is the read length and  $N$  is the maximum number of occurrences of a unique (occurring once)  $k$ -finger of  $s$  in the reads considered by the previous iterations. At the end of the iterations, the algorithm has computed all the common regions between the input reads. Observe that comparing reads in a reference-free approach often requires a pairwise comparison and is computationally demanding (refer for example to the problem of the identification of the relationships between metagenomic reads [16]). We have implemented our method in the Python prototype **KFinger** taking as input a set of reads and producing as output the pairs of reads in overlap. We have tested it over an error-free dataset of long reads simulated from a 2M-long region of the human chromosome 21 by using **DeepSimulator** [17] and a real PacBio HiFi set. We have compared the results from **KFinger** and **Minimap2** [14].

Overall, **Minimap2** produces more overlapping pairs than **KFinger** and the percentage of overlaps with high error rate (error rate over 3.0%) is higher for **Minimap2** than **KFinger**. Observe that pair of reads, that share a short overlap, are expected to be missed by our method, but, on the other hand, with the purpose of reconstructing an assembly, these pairs of reads may be discarded. The obtained results suggest that **KFinger** is less sensitive than **Minimap2** in the face of a quite high specificity. To test this hypothesis, we also compared the results (from **KFinger** and **Minimap2**) with the overlaps obtained by mapping the input reads to the reference genome.

## 2 Preliminaries

Let  $s = c_1 \cdots c_p$  be a string over a finite alphabet  $\Sigma$ . The *length* of  $s$  (that is, the number  $p$  of its characters) will be denoted by  $|s|$ . A *prefix* of  $s$  is a string composed of its first  $i$  characters (that is,  $c_1 \cdots c_i$ ). Similarly, a *suffix* is a string composed of the last  $i$  characters of  $s$  (that is,  $c_{n-i+1} \cdots c_n$ ).

A prefix (or suffix) is *proper* if it does not cover the whole string  $s$ . In the following, notation  $s < s'$  (resp.  $s \leq s'$ ) will specify that string  $s$  is lexicographically smaller than  $s'$  (resp.  $s = s'$ ). Furthermore,  $s \ll s'$  will specify that  $s < s'$  and additionally  $s$  is not a proper prefix of  $s'$ .

Now, we introduce the two main ingredients for capturing common regions between two strings (or reads): the definitions of *factorization* and *fingerprint*. Precisely, a *factorization* of a string  $s$  is a sequence  $F(s) = \langle f_1, f_2, \dots, f_n \rangle$  of factors (strings over  $\Sigma$ ), such that  $s = f_1 f_2 \cdots f_n$  and the *fingerprint*, with respect to  $F(s)$ , is the sequence  $\mathcal{L}(s) = \langle |f_1|, |f_2|, \dots, |f_n| \rangle$  of the factor lengths.

Given a fingerprint  $\mathcal{L}(s) = \langle l_1, l_2, \dots, l_n \rangle$ , a *k-finger* is a  $k$ -mer of  $\mathcal{L}(s)$ , that is, any substrings  $\langle l_i, l_{i+1}, \dots, l_{i+k-1} \rangle$  composed of  $k$  consecutive elements of  $\mathcal{L}(s)$ . The sum  $l_i + l_{i+1} + \dots + l_{i+k-1}$  will be referred as *supporting length* of the  $k$ -finger. Moreover, the index  $i$  and the sum  $l_1 + l_2 + \dots + l_{i-1}$  of the upstream elements (lengths) of the fingerprint will be referred as *index offset* and *length offset* of the  $k$ -finger with respect to the fingerprint.

The substrings  $f_i f_{i+1} \cdots f_{i+k-1}$  will be the *supporting string* of the  $k$ -finger.

*Example 1.* Let  $F(s) = \langle aaaaa, cccc, \mathbf{aaaaaa}, \mathbf{cccc}, \mathbf{ttt}, a \rangle$  be the factorization of  $s$  and let  $\mathcal{L}(s) = \langle 5, 4, \mathbf{6}, \mathbf{5}, \mathbf{3}, 2, 1 \rangle$  be the fingerprint. The three bold consecutive integers  $\langle \mathbf{6}, \mathbf{5}, \mathbf{3} \rangle$  are a 3-finger, whose supporting length is 14 and supporting string is the concatenation of the three bold factors of the factorization. The *index offset* of the 3-finger is 3, since its first element is the third in the whole fingerprint, and the *length offset* is 9, which the sum of the upstream elements 5 and 4. The *length offset* gives the offset of the supporting string in  $s$ .

In order to obtain read fingerprints, in this work we will exploit special kinds of factorizations, named *Lyndon based factorizations* [15] since they are defined starting from the well-known Lyndon factorization of a string  $s$  [1]. We firstly recall that each string  $s$  can be uniquely factorized into *Lyndon words* [1], where a Lyndon word is a word which is strictly smaller than any of its non empty proper suffixes. For example, it is easy to see that *accgctct* is a Lyndon word, whereas *cac* is not a Lyndon word. Formally, given a string  $s$ , its Lyndon factorization is denoted by  $CFL(s) = \langle f_1, f_2, \dots, f_n \rangle$ , where  $f_1 \geq f_2 \geq \dots \geq f_n$  and each  $f_i$  is a Lyndon word. For example, given  $s_1 = gcatcaccgctctacagaac$ , we have that  $CFL(s_1) = \langle g, c, atc, accgctct, acag, aac \rangle$ . In [18], the *Canonical Inverse Lyndon factorization*  $ICFL(s) = \langle f_1, f_2, \dots, f_n \rangle$  is a factorization of  $s$  such that  $f_1 \ll f_2 \ll \dots \ll f_n$  and each  $f_i$  is an *inverse Lyndon word* [18], that is, each non empty proper suffix of  $f_i$  is strictly smaller than  $f_i$ . For example, *cac*, *tcaccgc* are inverse Lyndon words. Let us consider again  $s_1 = gcatcaccgctctacagaac$ . We have that  $ICFL(s_1) = \langle gca, tcaccgc, tctacagaac \rangle$ . Such factorizations are unique and can be computed in linear time and constant space [18].

A property of  $CFL(s) = \langle f_1, f_2, \dots, f_n \rangle$ , which is crucial in our framework, is the following *Conservation Property* [19]. Suppose that  $CFL(s) = \langle f_1, f_2, \dots, f_n \rangle$  and let  $z = f'_l f_{l+1} \dots f_t f'_{t+1}$  be a non simple factor w.r.t.  $CFL(s)$  (i.e., it properly contains at least one factor), for some indexes  $l, t$  with  $1 \leq l < n$ ,  $1 < t < n$ , and  $f_l = f''_l f'_l$ ,  $f_{t+1} = f'_{t+1} f''_{t+1}$ . A main consequence of the conservation property proved in [18] is that, given two strings  $s$  and  $s'$  sharing a common overlap  $z$ , there exist factors that are in common between  $CFL(w)$  and  $CFL(w')$ . Thus,  $s$  and  $s'$  will have fingerprints sharing  $k$ -fingers for a suitable size  $k$ . For example, consider again  $s_1 = gcatcaccgctctacagaac$  and  $s_2 = ccaccgctctacagaagcatc$ . We know that  $CFL(s_1) = \langle g, c, atc, accgctct, acag, aac \rangle$  and we have that  $CFL(s_2) = \langle c, c, accgctct, acag, aagcatc \rangle$ . Hence, we have  $\mathcal{L}(s_1) = \langle 1, 1, 3, 8, 4, 3 \rangle$  and  $\mathcal{L}(s_2) = \langle 1, 1, 8, 4, 7 \rangle$ . The two common consecutive elements  $\langle 8, 4 \rangle$  are related to the same factors in the strings (8 is related to *accgctct* and 4 is related to *acag*) and capture the common substring *accgctctacag* given by their concatenation.

Our method exploits the previous result and is based on the following assumption: a  $k$ -finger occurring in different read fingerprints has the same supporting string. This assumption is fundamental in order to capture common regions between reads by using fingerprints and  $k$ -fingers while ignoring the string characters. We define  $CFL.ICFL$  the factorization obtained by applying first the Standard Lyndon Factorization  $CFL$ , and then the Canonical Inverse Lyndon factorization  $ICFL$  to factors (of  $CFL$ ) longer than a given threshold. In other



words, given  $\text{CFL}(s) = \langle f_1, f_2, \dots, f_n \rangle$ , we obtain  $\text{CFL\_ICFL}(s)$  by replacing with  $\text{ICFL}(f_i)$  each  $f_i$  longer than the threshold.

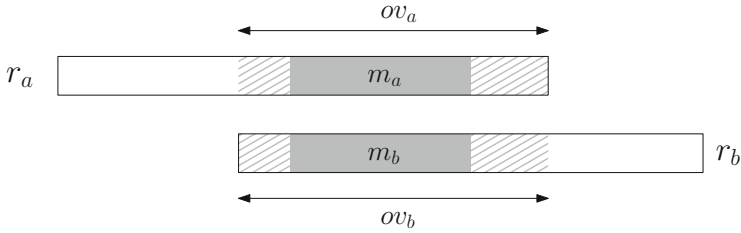
Observe that  $\text{CFL\_ICFL}$  has the main advantage of producing many factors, thus enriching the set of  $k$ -fingers to use for detecting the common regions between reads. In [15], in order to deal with the double-stranded nature of sequencing reads it is proposed a factorization algorithm  $F^d(s) = \langle f_1, f_2, \dots, f_n \rangle$  such that  $F^d(\bar{s})$  is equal to  $\langle \bar{f}_n, \bar{f}_{n-1}, \dots, \bar{f}_1 \rangle$ , where  $\bar{f}_i$  is the reverse and complement of  $f_i$ . Recall that the *reverse and complement* of a string  $s$  over the DNA alphabet  $\{A, C, G, T\}$  is the string  $\bar{s}$ , such that its  $i$ -th character is the complement of the  $(|s| - i + 1)$ -th character of  $s$ , where the *complement* is the operation transforming the DNA symbol  $A$  into the DNA symbol  $T$  (and vice versa) and the DNA symbol  $C$  into the DNA symbol  $G$  (and vice versa). This double-stranded factorization relies on a basic algorithm  $F$  such as  $\text{CFL}$ ,  $\text{ICFL}$  or  $\text{CFL\_ICFL}$ , and is obtained by combining  $F(s)$  with  $F(\bar{s})$ , with the result of reducing the length of the factorization factors [15].

Observe that the fingerprint of  $s$  will be equal to the reverse of the fingerprint of  $\bar{s}$  and, as a consequence, the same genomic region on the two opposite strands will be supporting two  $k$ -fingers, which are one the reverse of the other.

### 3 Detecting Reads in Overlap

In our framework, we consider in overlap two reads  $s$  and  $s'$  between which, one of the following relations occurs: (i) a proper suffix of  $s$  has a match with a proper prefix of  $s'$  (or vice versa), (ii)  $s$  has a match with a substring of  $s'$  (or vice versa). In absence of sequencing errors, the suffix of  $s$  will be equal to the prefix of  $s'$  (or vice versa) in case (i) and  $s$  will be equal to the substring of  $s'$  (or vice versa) in case (ii). Clearly, when sequencing errors are present, the equality relation must be transformed into a similarity relation. Observe that the above relation (i) holds for two reads sequenced from the same genomic strand. When the reads come from opposite strands, then relation (i) must be turned into the following one: a proper suffix (resp. prefix) of  $s$  has a match with a proper suffix (resp. prefix) of  $s'$  (or vice versa). Obviously, in both cases the matching occurs except for a reverse and complement operation of one of the two involved read substrings. Our aim is to use fingerprints and  $k$ -fingers obtained from Lyndon-based factorizations for capturing common regions between reads in an input set and inferring pairs of reads in overlap. Given an input set of reads, our method duplicates each input read. In other words, we expand the input set by adding the reverse and complement version of each input read. Then, it computes for each read (of the expanded set) a Lyndon factorization from which to obtain the fingerprint (read signature) and extract the  $k$ -fingers. Next, it exploits the obtained  $k$ -fingers to detect common regions between reads and infers the pairs of reads in overlap (or overlapping pairs) in the expanded input set. Observe that, following the duplication approach to handle the double-stranded nature of the reads, we only have to deal with suffix-prefix overlaps as if the reads originated from the same strand. Next, a post-processing step obtains the overlapping pairs of the original input set and (if needed) converts suffix-prefix overlaps into suffix-suffix (or prefix-prefix) overlaps between reads from opposite strands.





**Fig. 1.** Suffix-prefix overlap between reads  $r_a$  and  $r_b$  having common region  $(m_a, m_b)$ . A suffix of  $r_a$  has a match with a prefix of  $r_b$ .

### 3.1 The Method

Let  $S = \{s_1, s_2, \dots, s_r\}$  be the set of the input reads (strings over the DNA alphabet) and let  $\bar{s}_i$  be the reverse and complement of  $s_i$ . The set  $S$  is first expanded into the set  $S_e = \{s_1, \bar{s}_1, s_2, \bar{s}_2, \dots, s_r, \bar{s}_r\}$ . Then, (first step), each read in  $S_e$  is split into segments of a given length  $\mathcal{X}$  (observe that the last segment may be smaller) and each segment is factorized by using a factorization algorithm (among the ones described in the previous section). The fingerprint of a read will be the concatenation of the fingerprints of its segments. The read segmentation has the advantage of producing richer fingerprints in terms of number of elements and therefore in terms of  $k$ -fingers to use to capture similarities. Next (second step), the read fingerprints are exploited to obtain the pairs of reads (of the expanded set  $S_e$ ) sharing a common region. Observe that we are not interested in overlapping pairs  $(s_i, \bar{s}_i)$  composed of a read and its reverse and complement. This step considers pairs  $(r_a, r_b)$  such that  $r_a$  is  $s_i$  or  $\bar{s}_i$  and  $r_b$  is in  $\{s_{i+1}, \bar{s}_{i+1}, \dots, s_r, \bar{s}_r\}$  (the vice versa is indeed redundant). This step basically finds two common unique  $k$ -fingers (occurring uniquely in the two reads) to use as anchors of the common region between  $r_a$  and  $r_b$ . For each computed common region (third step), the suffix-prefix overlap is obtained by extending the common region to the left endpoint of a read and to the right endpoint of the other read (as depicted in Fig. 1). When the common region does not cover a certain percentage  $P$  of the putative overlap, then the pair  $(r_a, r_b)$  is not an overlapping pair and will not be produced as output.

Finally (fourth step), after computing all the suffix-prefix overlaps of the expanded set  $S_e$ , a post-processing step computes the overlapping pairs of the original input set  $S$ . Precisely, let  $s_i, s_j$  and  $\bar{s}_i, \bar{s}_j$  be two input reads and their reverse and complement versions. Assuming  $i < j$ , then the overlapping pairs  $(s_i, s_j), (s_i, \bar{s}_j), (\bar{s}_i, s_j)$  and  $(\bar{s}_i, \bar{s}_j)$  may be coexist in the output of the third step. Hence, a trivial strategy is applied to only retain just one among those pairs, which consists in selecting the first pair produced by the algorithm. Observe that sophisticated strategies have been tested (using some criteria based on the read strand) but we did not obtain a significant improvement in the results. Observe that when the selected pair is  $(s_i, \bar{s}_j)$  or  $(\bar{s}_i, s_j)$  (that is, it involves reads from opposite strands), then the suffix-prefix overlap is converted into a suffix-suffix or

prefix-prefix overlap. When the selected pair is  $(\bar{s}_i, \bar{s}_j)$ , the suffix-prefix overlap is reported onto the original reads  $s_i$  and  $s_j$ .

The following paragraphs are devoted to detail the second step which is the core of our method and works in two sub-steps: first, the *candidate pairs* are computed (see Algorithm 1) and then, the common regions are obtained. Basically, Algorithm 1 performs a linear scanning of the reads of  $S_e$  and, for each read fingerprint, the  $k$ -fingers are considered from the leftmost to the rightmost. The goal is to compute a hash table  $C$  storing the pairs  $(r_a, r_b)$  sharing at least  $U$  unique  $k$ -fingers (that is, occurring only once in both reads), which are referred as *candidate pairs*. The leftmost (unique)  $k$ -finger shared by  $r_a$  and  $r_b$  is stored in  $C$  for each candidate pair  $(r_a, r_b)$  together with its *length offsets* and *index offsets* in the fingerprints of the two reads. The returned hash table  $C$  gives for a key  $(r_a, r_b)$  (candidate pair) the tuple  $(f_l, \omega_a^l, i_a^l, \omega_b^l, i_b^l)$ , where  $f_l$  is the common leftmost  $k$ -finger,  $\omega_a^l$  and  $\omega_b^l$  are the length offsets for  $r_a$  and  $r_b$  (respectively) and  $i_a^l$  and  $i_b^l$  are the index offsets for  $r_a$  and  $r_b$  (see Example 1). The algorithm uses a support hash table  $H$  storing the  $k$ -fingers and their localization in the reads (length offset and index offset): for each  $k$ -finger  $f$ , the value  $H(f)$  is a list of tuples  $(r, \omega, i)$ , where each tuple gives the localization of  $f$  in the fingerprint of a read  $r$ . For each considered read  $r_b$  (see the main **foreach** cycle at line 3) and for each  $k$ -finger  $f$ , its localization in  $r_b$  is stored in the hash table  $H$  (see **foreach** cycle at line 5). Then,  $H$  is updated such that it contains only the localizations of the unique  $k$ -fingers of  $r_b$  (see **foreach** cycle at line 9) and at the same time such unique  $k$ -fingers are stored in the list *unique.list*. The **if** condition at line 10 checks whether the  $k$ -finger  $f$  is unique in  $r_b$ . In fact, if  $f$  is not unique, then the  $n > 1$  trailing tuples of list  $H(f)$  will be related to  $r_b$ . At each iteration of the main **foreach** cycle, the support hash table  $H$  contains, for each read already processed before  $r_b$ , only the localizations of its unique  $k$ -fingers. The last **foreach** cycle at line 15 considers each unique  $k$ -finger of  $r_b$  and finds its localizations in the other reads (processed before  $r_b$ ) in order to compute all the candidate pairs involving  $r_b$  as second read. Observe that the  $k$ -fingers are always considered from left to right in the read fingerprints and the two **foreach** cycles at lines 9 and 15 guarantee that the  $k$ -finger  $f$ , stored in  $C$  for a candidate pair  $(r_a, r_b)$ , is the leftmost unique  $k$ -finger shared by the two reads. Algorithm 1 performs a linear scanning of the read fingerprints and the three **foreach** cycles at lines 5, 9 and 15 perform a linear scanning of the read  $k$ -fingers whose number is asymptotically equal to the read length. Finally, observe that the **foreach** cycle at line 16 only checks the tuples in the list  $H(f)$  (of the support hash table  $H$ ) whose size is the number of reads (among the ones already processed) containing a unique occurrence of the  $k$ -finger  $f$ . Even though it is not specified by the algorithm, only  $k$ -fingers whose *supporting length*, i.e. the sum of the lengths in the  $k$ -finger, is at least a given threshold  $\tau$  are considered. The parameter  $\tau$  is the threshold we use to consider a  $k$ -finger reliable and avoid collisions (that is, the same  $k$ -finger which is supported by different strings in different reads).

---

**Algorithm 1:** Compute the candidate pairs
 

---

**Input :** Fingerprints of the reads of the expanded set  $S_e$   
**Output:**  $C$ , hash table of the candidate pairs

```

1  $H \leftarrow$  empty hash table;
2  $C \leftarrow$  empty hash table;
3 foreach fingerprint  $\mathcal{L}$  do
4    $r_b \leftarrow$  read whose fingerprint is  $\mathcal{L}$ ;
5   foreach  $k$ -finger  $f \in \mathcal{L}$  do // From the leftmost to the rightmost
6      $(\omega, i) \leftarrow$  length offset and index offset of  $f$ ;
7     Add  $(r_b, \omega, i)$  to the list  $H(f)$ ;
8    $unique\_list \leftarrow$  empty list;
9   foreach  $k$ -finger  $f \in \mathcal{L}$  do
10    if the last  $n > 1$  tuples of  $H(f)$  are related to  $r_b$  then
11      | Remove from  $H(f)$  the last  $n$  tuples;
12    else //  $f$  is unique in  $r_b$ 
13      | Append  $f$  to  $unique\_list$ ;
14     $already\_processed \leftarrow$  empty set;
15    foreach  $f \in unique\_list$  do
16      foreach  $(r_a, \omega_a^l, i_a^l) \in H(f)$  do
17        if  $r_a \neq r_b$  and  $r_a \notin already\_processed$  and  $(r_a, r_b) \notin C$  then
18          if  $r_a$  and  $r_b$  share at least  $U$  unique  $k$ -fingers then
19            |  $(\omega_b^l, i_b^l) \leftarrow$  length offset and index offset of  $f$  in  $\mathcal{L}$ ;
20            |  $C(r_a, r_b) \leftarrow (f, \omega_a^l, i_a^l, \omega_b^l, i_b^l)$ ;
21          else
22            | Add  $r_a$  to  $already\_processed$ ;
23 return  $C$ 
    
```

---

For each candidate pair  $(r_a, r_b)$  in the hash table  $C$ , the algorithm uses the tuple  $(f_l, \omega_a^l, i_a^l, \omega_b^l, i_b^l)$  returned by  $C$  to localize the two longest subsequences (consecutive elements) of fingerprints  $\mathcal{L}(r_a)$  and  $\mathcal{L}(r_b)$  of  $r_b$  which satisfy the following three conditions: (1) both subsequences have  $k$ -finger  $f_l$  as prefix, (2) they share at least the last  $k'$  elements (where  $k'$  is an input parameter) and the  $k'$ -finger corresponding to such elements uniquely occurs in the two reads and has a minimum supporting length (to avoid collisions), and (3) the sum of the elements (integer values) of the first subsequence differs from the sum of the elements of the second subsequence by an upper threshold, we call *length tolerance*. The algorithm further extends as much as possible on the right the two subsequences while maintaining the equality of the corresponding elements.

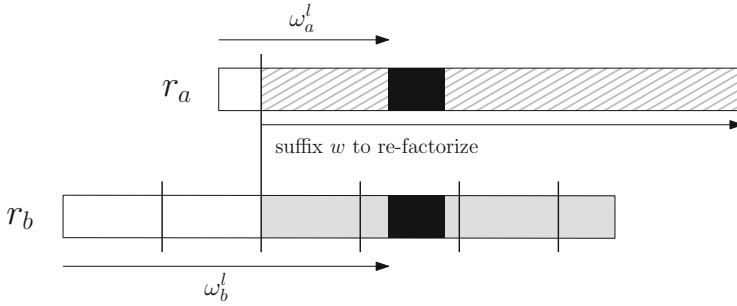
*Example 2.* Let  $\mathcal{L}(r_a) = \langle 5, 4, 3, \mathbf{10}, \mathbf{6}, \mathbf{5}, \mathbf{3}, \mathbf{2}, \mathbf{7}, \mathbf{3}, 4 \rangle$  be the fingerprint of  $r_a$  and let  $\mathcal{L}(r_b) = \langle 2, 2, \mathbf{10}, \mathbf{6}, \mathbf{3}, \mathbf{2}, \mathbf{2}, \mathbf{2}, \mathbf{7}, \mathbf{3}, 3, 9 \rangle$  be the fingerprints of  $r_b$ . Assuming  $k = 2$ ,  $k' = 2$ , a *length tolerance* set to 3 and a minimum supporting length set to 10 for  $k$ -fingers, then the bold subsequences satisfy the above conditions. Indeed, both ones start with the 2-finger  $\langle 10, 6 \rangle$  which is the leftmost common  $k$ -finger (occurring just once in the reads) having a supporting length at least 10. Moreover, they share the last  $k'$ -finger (the last  $k'$  elements)  $\langle 7, 3 \rangle$  having a

supporting length at least 10 (assuming that 10 is also the minimum supporting length for the  $k'$ -finger terminating such subsequences). Finally, the sum of the bold subsequence of  $r_a$  is equal to 36, while the sum of the bold subsequence of  $r_b$  is equal to 35 and their difference satisfies the assumed length tolerance. Hence, the common region between  $r_a$  and  $r_b$  (computed by our method) will be composed of the 36-long substring starting at position  $5 + 4 + 3 + 1 = 13$  of  $r_a$  and the 35-long substring starting at position  $2 + 2 + 1 = 5$  of  $r_b$ . At this point, the common region between the reads is obtained by retrieving the two read substrings, referred in the following as *common region*, supporting the two computed fingerprint subsequences. The *length tolerance* admitted in condition (3) takes into account possible sequencing errors of the reads and is the maximum difference between the length of the two detected common read substrings. Observe that the two fingerprint subsequences may share only a prefix (the leftmost  $k$ -finger  $f_l$ ) and a suffix and the equality of the corresponding integers may be interrupted because of sequencing errors or read segmentation (see the first step). Our method also allows to perform a *re-factorization* of one of the two reads  $r_a$  and  $r_b$  before computing the core common region, motivated by the fact that the read segmentation (see step one) may lead to a misalignment in the segment fingerprints, thus inducing to lose common factors in the overlapping regions between two reads.

During re-factorization, the read (between  $r_a$  and  $r_b$ ), where the common leftmost  $k$ -finger  $f_l$  has the smallest length offset, is selected; the suffix  $w$  to re-factorize is computed as described by Fig. 2. Note that  $w$  is aligned with a factorization segment of the other read and therefore the fingerprint of  $w$  will be compared with the suffix of the fingerprint of the other read starting from such segment. The common region between  $r_a$  and  $r_b$  will be computed, as described before, starting from the common leftmost  $k$ -finger shared by the two new fingerprints. In case of re-factorization, the common region between the two reads will be the longest between the ones computed before and after re-factorization.

## 4 Experimental Results

The method has been implemented in the Python prototype `KFinger` and it is available at <https://github.com/AlgoLab/kfinger> along with all the scripts needed to replicate the experiments. The tests have been performed on an Ubuntu 20.04 laptop with a single Intel® Core™ i5-8250U CPU and 16 GB of RAM over the following datasets: (1) a dataset of 10K error-free long reads simulated from the region of the human chromosome 21 between positions 32 000 000 and 34 000 000 (2 000 000bp), by using `DeepSimulator` [17] and (2) a dataset of 6141 real PacBio HiFi reads extracted from PacBio Sequel II HiFi sequencing of sample HG00731 of a Puerto Rican Trio. Precisely, the reads were mapped against the human genome GRCh38 (GCA\_000001405.15, no ALT contigs) using `Minimap2` (version 2.17) with preset `asm20` (as suggested in its documentation for aligning PacBio HiFi/CCS genomic reads). Only primary alignments were



**Fig. 2.** Re-factorization scheme. The two reads  $r_a$  and  $r_b$  are depicted as horizontal bars aligned according to the common leftmost  $k$ -finger  $f_l$  whose supporting strings are depicted as black boxes.  $\omega_a^l$  and  $\omega_b^l$  are the length offsets of  $f_l$  and the vertical bars on  $r_b$  (which has the largest length offset for  $f_l$ ) are the edges between consecutive factorization segments (only edges in the portion aligned with  $r_a$  are shown). The leftmost edge falling in  $r_a$  determines the starting point of the suffix  $w$  of  $r_a$  to re-factorize (highlighted in tiled grey), whose fingerprint will be compared with the fingerprint of  $r_b$  corresponding to the portion highlighted in solid grey.

retained. We then extracted in FASTA format reads overlapping region 32M–34M of chromosome 21. The final dataset was composed by 6141 reads with average length 11124bp (min 2349bp, max 23263bp, median 10417bp) for a total of 68316199 bases. The error rate inferred from the alignment was  $6.22 \times 10^{-3}$ . Original sequence files are available at [http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data\\_collections/HGSVC2/working/20190925\\_PUR\\_PacBio\\_HiFi/](http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data_collections/HGSVC2/working/20190925_PUR_PacBio_HiFi/) (run IDs: r54329U\_190528 – r54329U\_190906).

The first dataset will be referred as **error-free-ds** while the second one as **hifi-ds**. Recall that each input read to **KFinger** is accompanied by its reverse and complement, so that the size of the two datasets is 20000 for **error-free-ds** and 12282 for **hifi-ds**. We have used a double-stranded factorization algorithm built over the basic CFL.ICFL factorization algorithm with threshold parameter 30 (that is, factors of CFL factorization longer than 30 are submitted to ICFL factorization), by splitting each read into segments of length  $\mathcal{X} = 300$ bp in order to limit the factor lengths. The common regions between reads were computed for both datasets before and after re-factorization; then, the overlaps from common regions covering at least a percentage  $P = 80\%$  (coverage percentage of the putative overlap), were obtained. For finding the candidate pairs, we used a  $k$ -finger size set to 7 ( $k = 7$ ) and a minimum supporting length set to 40; 6 is the minimum number of unique shared  $k$ -fingers required for a candidate pair. Moreover, before re-factorization, we required  $k' = 3$  and a length tolerance set to 0 for **error-free-ds** and  $k' = 2$  and length tolerance set to 15 for **hifi-ds**. After re-factorization, we used  $k = 5$  and a minimum supporting length set to 10. We maintained the above values for parameter  $k'$  for the two datasets and the length tolerance set to 0 for **error-free-ds**, while setting to 20 the length tolerance for **hifi-ds**. We have compared, in terms of accuracy,

KFinger with `Minimap2` [14] by retaining only the common regions produced by `Minimap2` having a minimum coverage percentage  $P = 80\%$  with respect to the putative overlap and computing the overlaps on such common regions. Observe that records involving the same read have been discarded both for `KFinger` and `Minimap2`. For each common region and each overlap obtained with `Minimap2` and `KFinger`, we computed an error rate as the ratio of the edit distance, between the two read substrings involved in a common region or an overlap, and the smaller substring length. Tables 1 and 2 report the results for datasets `error-free-ds` and `hifi-ds`, respectively. Both tables report the results on common regions and overlaps produced by `KFinger` before (rows **K**) and after re-factorization (rows **KR**) and on common regions and overlaps obtained from `Minimap2` (rows **M**). In the following, we refer to a common region or an overlap with the generic term *record*. The first three columns “#0”, “# $\leq 3.0$ ” and “# $> 3.0$ ” give the number of records having an error rate equal to 0, greater than 0 but at most 3.0% and over 3.0%, respectively. The last three columns `MinL`, `MaxL` and `AvgL` report the minimum, maximum and average length of the read substrings involved in the record. Overall, `Minimap2` finds more records than `KFinger`. Over `error-free-ds` `Minimap2` outputs a total of 1286932 common regions, 179757 out of them are alternative overlaps between reads, against the 533584 produced by `KFinger`. Observe that a given pair of reads may be involved in more than one record; only `Minimap2` produces alternatives, whereas `KFinger` gives (by choice) just one common region/overlap for two given reads. Then, over `hifi-ds`, `Minimap2` finds a total of 530529 common regions (108655 out of them are alternatives) against the 211230 produced by `KFinger`. Moreover, `Minimap2` finds a total of 502377 overlaps (4455 out of them are alternatives) over `error-free-ds` and a total of 132160 overlaps (461 out of them are alternatives) over `hifi-ds`. `KFinger` produces 433947 overlaps over `error-free-ds` and 147029 overlaps over `hifi-ds` before re-factorization, whereas it produces 465121 overlaps over `error-free-ds` and 173057 overlaps over `hifi-ds` after re-factorization.

Column “# $> 3.0$ ” reports in parentheses the percentage of records (having an error rate over 3.0%) with respect to the total number of obtained records. We consider this value as a proxy of the false positive rate of the prediction. Observe that this percentage is rather low for `KFinger` both for common regions and overlaps, whereas for `Minimap2` it is low only for the overlaps since the parameter  $P = 80\%$  contributes to filter out the common regions produced by `Minimap2` not leading to a good read overlap. Moreover, the re-factorization has mainly determined an improvement in terms of detected overlaps, since for dataset `error-free-ds` 31079 extra overlaps (with an error rate  $\leq 3.0$ ) were detected after the re-factorization with respect to the experiment before re-factorization. Similarly, the re-factorization has produced 25486 extra overlaps for dataset `hifi-ds`. These results suggest that `KFinger` does not compete with `Minimap2` in terms of sensitivity but it is likely to be more specific in terms of common regions. Indeed, `Minimap2` is more tolerant with respect to the sequencing errors and therefore finds more common regions than `KFinger`. On the other

**Table 1.** Experimental results for **error-free-ds**. The rows tagged as **K** and **KR** refer to the common regions/overlaps produced by **KFinger** before and after re-factorization, whereas the row tagged as **M** refers to common regions/overlaps obtained from **Minimap2**.

	#0	# $\leq$ 3.0	# $>$ 3.0	MinL	MaxL	AvgL
Common regions						
K	473053	7643	52888 (10%)	40	37383	4339
KR	474107	7103	52374 (10%)	39	37414	4550
M	498479	29577	758876(59%)	100	37441	2246
Overlaps ( $P = 80\%$ )						
K	433884	8	55 (0.1%)	95	37441	5622
KR	464958	13	150 (0.1%)	95	37441	5364
M	496289	2142	3946(1%)	100	37441	5055

hand, **KFinger** gives fewer common regions and seems to be more precise. To test this hypothesis, and, in particular, to evaluate sensitivity, we compared the predicted common regions with the overlaps computed by mapping reads to the reference genome. We mapped the two datasets to region 32M–34M of human chromosome 21 using **Minimap2** and we kept only reads aligning for at least 95% of their length. From these alignments we devised the set of overlaps such that the length of the overlap was at least 80% of the length of the genomic region spanned by the two reads. We define this set as the set of “alignment-based” overlaps. Please note that we do not expect that the set of alignment-based overlaps coincides with the set of predicted overlaps since (*i*) some overlaps were discarded because of their length and since (*ii*) there exists common regions between reads that do not actually overlap on the genome. For each alignment-based overlap, we checked if there exists a predicted common region that intersects the overlap for at least 50% of their span. If it exists, we considered the alignment-based overlap as found. The dataset **error-free-ds** contains 9273 alignment-based overlaps. As expected, **Minimap2** found all of them, while **KFinger** missed 5 of them before re-factorization and 3 of them after re-factorization. The dataset **hifi-ds** contains 16207 alignment-based overlaps. **Minimap2** was not able to find 2 of them, while **KFinger** missed 1743 of them before re-factorization and 753 of them after re-factorization. These results support the hypothesis that **Minimap2** is more sensitive and more tolerant than **KFinger**, but, on the other hand, it is also less specific, since **Minimap2** reports twice as much common regions as **KFinger**. In terms of time efficiency, we measured the whole time for computing the candidate pairs and the common regions. These two steps are indeed the intensive part of the method. Moreover, the time is given before re-factorization, since the current implementation of the read factorization is not optimal. On a single thread, **KFinger** took 12min and 5s for dataset **error-free-ds** and 4min and 2s for dataset **hifi-ds**. Despite being highly optimized, **Minimap2**

**Table 2.** Experimental results for **hifi-ds**. The rows tagged as **K** and **KR** refer to the common regions/overlaps produced by **KFinger** before and after re-factorization, whereas the row tagged as **M** refers to common regions/overlaps obtained from **Minimap2**.

	#0	# $\leq$ 3.0	# $>$ 3.0	MinL	MaxL	AvgL
Common regions						
K	10309	184461	16460 (8%)	40	17853	4392
KR	6976	187396	16858 (8%)	39	18063	4933
M	9449	217870	303210(57%)	100	18811	2531
Overlaps ( $P = 80\%$ )						
K	2583	143916	530 (0.3%)	97	18169	6036
KR	3275	168710	1072 (0.1%)	97	18169	5880
M	2646	109115	20399(15%)	103	19664	6623

took 4 min and 42 s for dataset **error-free-ds** and 2 min and 16 s for dataset **hifi-ds**.

## 5 Conclusions and Future Developments

We have proposed a method for detecting overlaps in a set of long reads by using a compact numerical representation (fingerprint) based on Lyndon factorization. The method has been implemented in the Python prototype **KFinger** which has been tested over a set of error-free simulated reads and a PacBio HIFI dataset. The experimental results encourage to think that **KFinger** may be a suitable and specific method for finding shared regions between pairs of reads, taking advantage of the compact numeric representation of the reads. In the immediate we plan to improve **KFinger** in terms of time efficiency by improving the implementation of (1) the factorization algorithms used for producing the input fingerprints and (2) of the steps two and three producing the common regions, improvement needed in terms of a more efficient programming language such as C++ and the use of more efficient data structures. In terms of accuracy we plan to investigate the impact of the different factorization algorithms in order to face the typical issues related to long reads: sequencing errors and repetitive regions.

## References







1. Lyndon, R.C.: On Burnside’s problem. *Trans. Am. Math. Soc.* **77**(2), 202–215 (1954)
2. Berstel, J., Perrin, D.: The origins of combinatorics on words. *Eur. J. Comb.* **28**(3), 996–1022 (2007)



3. Delgrange, O., Rivals, E.: Star: an algorithm to search for tandem approximate repeats. *Bioinformatics* **20**(16), 2812–2820 (2004)
4. Mantaci, S., Restivo, A., Rosone, G., Sciortino, M.: An extension of the Burrows-Wheeler transform. *Theoret. Comput. Sci.* **387**(3), 298–312 (2007)
5. Bonizzoni, P., De Felice, C., Zaccagnino, R., Zizza, R.: Lyndon words versus inverse lyndon words: queries on suffixes and bordered words. In: Loporati, A., Martín-Vide, C., Shapira, D., Zandron, C. (eds.) *LATA 2020*. LNCS, vol. 12038, pp. 385–396. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-40608-0\\_27](https://doi.org/10.1007/978-3-030-40608-0_27)
6. Chen, K.T., Fox, R.H., Lyndon, R.C.: Free differential calculus, IV. the quotient groups of the lower central series. *Ann. Math.* **68**(1), 81–95 (1958)
7. Pevzner, P.A., Tang, H., Waterman, M.S.: An Eulerian path approach to DNA fragment assembly. In: *Proceedings of the National Academy of Sciences*, vol. 98, no. 17, pp. 9748–9753. National Academy of Sciences (2001)
8. Berlin, K., Koren, S., Chin, C.-S., Drake, J.P., Landolin, J.M., Phillippy, A.M.: Assembling large genomes with single-molecule sequencing and locality-sensitive hashing. *Nat. Biotechnol.* **33**(6), 623–630 (2015)
9. Loman, N.J., Quick, J., Simpson, J.T.: A complete bacterial genome assembled de novo using only nanopore sequencing data. *Nat. Methods* **12**(8), 733–735 (2015)
10. Jain, C., Dilthey, A., Koren, S., Aluru, S., Phillippy, A.M.: A fast approximate algorithm for mapping long reads to large reference databases. *J. Comput. Biol.* **25**(7), 766–779 (2018)
11. Broder, A.: On the resemblance and containment of documents. In: *Proceedings. Compression and Complexity of SEQUENCES*, pp. 21–29. IEEE Computer Society (1997)
12. Pierce, N.T., Irber, L., Reiter, T., Brooks, P., Brown, C.T.: Large-scale sequence comparisons with sourmash. *F1000Research* **8**, 1006 (2019)
13. Koren, S., Walenz, B.P., Berlin, K., Miller, J.R., Bergman, N.H., Phillippy, A.M.: Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. *Genome Res.* **27**(5), 722–736 (2017)
14. Li, H.: MiniMap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34**(18), 3094–3100 (2018)
15. Bonizzoni, P., et al.: Can we replace reads by numeric signatures? Lyndon fingerprints as representations of sequencing reads for machine learning. In: Martín-Vide, C., Vega-Rodríguez, M.A., Wheeler, T. (eds.) *AlCoB 2021*. LNCS, vol. 12715, pp. 16–28. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-74432-8\\_2](https://doi.org/10.1007/978-3-030-74432-8_2)
16. Giroto, S., Pizzi, C., Comin, M.: MetaProb: accurate metagenomic reads binning based on probabilistic sequence signatures. *Bioinformatics* **32**(17), i567–i575 (2016)
17. Li, Y., Han, R., Bi, C., Li, M., Wang, S., Gao, X.: DeepSimulator: a deep simulator for Nanopore sequencing. *Bioinformatics* **34**(17), 2899–2908 (2018)
18. Bonizzoni, P., De Felice, C., Zaccagnino, R., Zizza, R.: Inverse Lyndon words and inverse Lyndon factorizations of words. *Adv. Appl. Math.* **101**, 281–319 (2018)
19. Bonizzoni, P., De Felice, C., Zaccagnino, R., Zizza, R.: On the longest common prefix of suffixes in an inverse Lyndon factorization and other properties. *Theoret. Comput. Sci.* **862**, 24–41 (2021)



# Can We Detect T Cell Receptors from Long-Read RNA-Seq Data?

Justyna Mika<sup>1</sup>  , Serge M. Candéias<sup>2</sup> , Christophe Badie<sup>3</sup> ,  
and Joanna Polanska<sup>1</sup>  

<sup>1</sup> Department of Data Science and Engineering, Silesian University of Technology, Gliwice, Poland

{justyna.mika, joanna.polanska}@polsl.pl

<sup>2</sup> Laboratoire de Chimie Et Biologie Des Métaux, Université Grenoble Alpes, CNRS, CEA, 38000 Grenoble, France

<sup>3</sup> Cancer Mechanisms and Biomarkers Group, Radiation Effects Department, Radiation Chemical & Environmental Hazards, UK Health Security Agency, Didcot, UK

**Abstract.** T cells play an essential role in defense of the organism against pathogens and cancer. Efficient protection requires a vast repertoire of immune receptors, which is created by the V(D)J recombination process. There are multiple algorithms designed for the annotation of recombined T cell receptor (TR) sequences from traditional (short-read) RNA-Seq, however, none is adjusted for the long-read data. Here we intend to examine whether existing methods for TR sequences annotation using traditional RNA-Seq can be utilized for long-read sequencing data. ImReP, TRUST4, CATT and MiXCR algorithms were applied to data obtained by nanopore technology (PromethION). Adjustment of parameters was performed. The biggest number of CDR3 sequences was detected by the TRUST4 algorithm (20,599 unique TR sequences out of 73,904,478 total reads), representing 25% of the expected number of sequences. The distribution of annotated V and J genes was the same for MiXCR and TRUST4 algorithms and may be used to analyze the repertoire of V/J gene used in rearranged TR genes. Due to the high sequencing error rate of the analyzed sample (median read quality  $Q = 6.9$ ), TR clonotype analysis is not suggested, and additional error correction steps are recommended for such analyses.

**Keywords:** TCR detection · Oxford Nanopore Sequencing · Long reads

## 1 Introduction

### 1.1 VDJ Recombination

B and T lymphocytes play an essential role in the organism's defense against pathogens and cancer. This defense is mediated by clonally distributed immunoglobulins (Ig) and T cell receptors (TR). Efficient protection requires a vast repertoire of different Igs and TRs, which is assured by the V(D)J recombination process required for their expression. It consists of the selection and assembly of one of each variable (V), diversity (D), and

joining (J) gene segments into an exon coding for the complementarity determining regions of TRs. The most variable is the complementarity determining region 3 (CDR3) at the junction of V, D and J genes. V, D and J gene ends are heavily processed by random nucleotide deletion and/or addition before ligation. Due to the random nature of these mechanisms created exons may be in-frame, out-of-frame or contain a stop codon. The last two cannot create a functional receptor or immunoglobulin but may be found circulating in the blood.

## 1.2 Immune Repertoire Identification–State of the Art

The ability to characterize the immune repertoire on a large scale and identify rearranged TRs was improved thanks to the advances in high-throughput sequencing. The most used approach is Ig/TR sequencing, which requires previous amplification of DNA material covering V/J genes region, including the CDR3 region. It allows obtaining nucleotide sequences of all possible rearranged Ig/TR genes focusing on the CDR3 region or full variable region.

Recently, the attention of scientists has been attracted by the detection of Ig and TR sequences from bulk RNA sequencing data, which is currently produced on a massive scale. In RNA-Seq experiments, only reads mapped to the reference genome are considered to study the transcriptomic profile of a donor. Unmapped reads, which might include sequences of rearranged V/J genes, are filtered out from the further analysis. Some algorithms have been developed for recycling these waste reads to detect TR or Ig sequences. This approach allows the identification of a donor's immune profile without the need to spend time and money on another experiment. Traditional RNA-Seq data is characterized by a low sequencing error rate, thanks to which the accuracy of identified immune receptor sequences is very high.

Some of the available algorithms are applicable only for B cell receptor identification (V'DJer [1], BASIC [2], BALDR [3], IgBLAST [4]), others can only identify T cell receptors (TraCeR [5], RTCR [6]), and many can be utilized for both types of cells (ImReP [7], TRUST4 [8], IMSEQ [9], MiXCR [10], VDJPuzzle2 [11], CATT [12]). Some of these algorithms are designed to work with targeted RNA-Seq only (IMSEQ, LymAnalyzer [13], TCRklass [14], RTCR). They assume the presence of V/J genes in every read, and thus are not applicable for detecting immune receptors from bulk RNA-Seq. VDJPuzzle and TraCeR are designed to work with scRNA-Seq data only.

## 1.3 CDR3 Detection from Long-Read Data

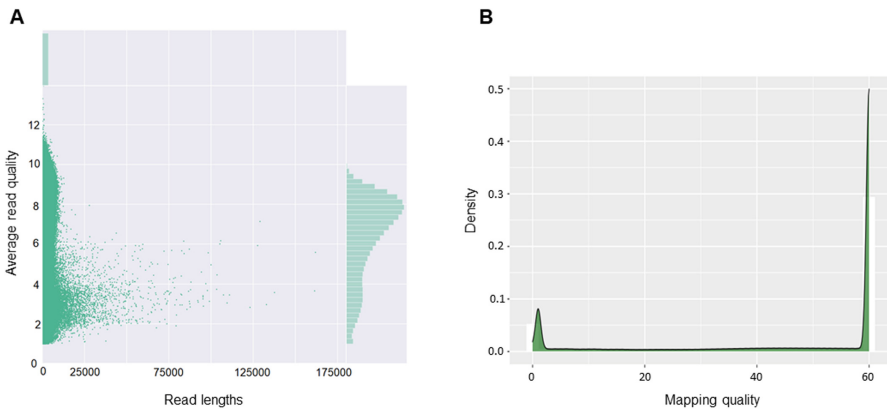
Long-read sequencing is a relatively new technology developed by Oxford Nanopore Technology (ONT) and Pacific Biosciences (PacBio). They create reads on the scale of kilobase pairs, however, the error rate is high due to the difficulty in identifying the DNA bases from the complex electrical signals [15]. Types of errors introduced to the sequences produced by ONT are more or less evenly distributed between insertions, deletions, or substitutions [16, 17]. This high error rate is problematic in detecting highly variable CDR3 region, for which no template exists. On the other hand, ONT sequencing is under current development due to the low cost, high throughput, and portability. We may expect it to become more accurate and popular in the near future. What is more,

long reads are able to cover the full rearranged TR region (consisting of  $\sim 500$ bp [6]), which is an advantage to short-read data covering only parts of TRs.

Currently, there are no algorithms specifically designed to identify immune repertoire from bulk long-read RNA sequencing. This study aims to check if methods for the identification of immune receptor sequences from non-targeted RNA-Seq short reads can be utilized for long-read data. We focus only on the CDR3 region of the  $\beta$  chain T cell receptors. We tested four algorithms (ImReP [7], TRUST4 [8], CATT [12], and MiXCR [10]) on a file obtained with PromethION (ONT). In addition, we checked if optimization of parameters may improve the work of algorithms on long reads.

## 2 Material

A collection of data from Cruz-Garcia et al. [18] was utilized. Poly A + RNA isolated from whole blood samples obtained from 3 donors pooled together (30 ml of blood in total) was sequenced in 2 flow cells of PromethION. The fastq file consisted of 73,904,478 raw reads with a median length of reads equal to 666 bp. The median sequencing quality of a read equaled to  $Q = 6.9$  in Phred score, which means that the median accuracy of sequencing is around 80% (every 1 out of 5 nucleotides is potentially falsely detected). Figure 1A shows a correlation between read length and average read quality, performed by the NanoPack algorithm [19].



**Fig. 1.** A—Scatter plot showing relations between read length and average read quality with corresponding histograms on sides performed by NanoPack, B—Histogram and density plot of mapping quality.

## 3 Methods

### 3.1 Mapping to Reference Genome

Minimap2 [20] was used to align reads to the GRChv38 human genome using default parameters. Samtools [21] was used for BAM file sorting. 54,218,305 reads were mapped

to the reference genome with high mapping quality (Fig. 1B). The remaining 19,686,173 reads were not mapped to the genome, serving as the primary source of sequences with variable CDR3 region. Rsamtools [22] was used to extract reads mapped to the TCR $\beta$  C1 and C2 gene regions (constant regions of T cell receptors).

### 3.2 Algorithms for TR $\beta$ Detection from Sequencing Data

Four algorithms: ImReP, TRUST4, CATT and MiXCR were applied to check if they can be used for the identification of CDR3 sequences from long-read, bulk RNA-Seq data. Table 1 summarizes the algorithms.

**Table 1.** Summary of algorithms for CDR3 detection from traditional RNA-Seq

	ImReP	TRUST4	CATT	MiXCR
Overview of the method	Alignment-free detection of reads containing full-length CDR3 $\rightarrow$ match reads containing partial CDR3 $\rightarrow$ correction of PCR and sequencing errors using CAST algorithm	Candidate reads extraction based on significant overlap criterion $\rightarrow$ de novo assembly using greedy seed extension approach $\rightarrow$ annotation and extension of partial alignments	Detection and assembly of CDR3 using de Bruijn graph $\rightarrow$ pattern match $\rightarrow$ data-driven error correction $\rightarrow$ annotation and confidence evaluation using Bayes classifier	Read alignment using Smith-Waterman/Needleman-Wunsh algorithms $\rightarrow$ partial alignment assembly and CDR3 extension $\rightarrow$ assembly of clonotypes and full receptor sequences
Requires previous mapping to the genome	Yes	No, but available	No, but available	No
Chains available	IGH, IGK, IGL, TRA, TRB, TRD, TRG	IGH, IGK, IGL, TRA, TRB, TRD, TRG	IGH, TRA, TRB	IGH, IGK, IGL, TRA, TRB, TRD, TRG
Complementarity determining regions available	CDR3	CDR1, CDR2, CDR3	CDR1, CDR2, CDR3	CDR1, CDR2, CDR3
Adjustable assembly parameters	Yes	No	No	Yes
Multithreaded processing	No	Yes	Yes	Yes
Error correction	Yes	Yes	Yes	Yes
Analysis of partial alignments	Yes	Yes	Yes	Yes
Outputs germline CDR3 sequence	No	Yes	Yes	Yes
Outputs out-of-frame CDR3 sequences	No	Yes	No	Yes
Reference page	<a href="https://github.com/Mangul-Lab-USC/imrep">https://github.com/Mangul-Lab-USC/imrep</a>	<a href="https://github.com/liu-lab-dfei/TRUST4">https://github.com/liu-lab-dfei/TRUST4</a>	<a href="https://github.com/GuoBio/infoLab/CATT">https://github.com/GuoBio/infoLab/CATT</a>	<a href="https://mixcr.readthedocs.io/en/master/index.html#">https://mixcr.readthedocs.io/en/master/index.html#</a>

All algorithms define TR CDR3 as an amino-acid sequence starting with cysteine (C) and ending with conserved phenylalanine (F) (i.e., ending with FGxG motif), as

proposed by the IMGT database [23]. All algorithms are publicly available, working as standalone programs. ImReP and MiXCR allow the user to change alignment and assembly parameters to optimize the work of the tool. CATT and TRUST4 provide predefined options to work on single-cell and 10x Genomics data, in addition to traditional bulk RNA-Seq data.

ImReP consists of two stages. First, every read is translated into an amino-acid sequence. Second, CDR3 is inferred from reads overlapping V and J gene segments simultaneously. Full CDR3 sequences are determined using adjustable parameters of minimum overlap and maximum mismatch between read and reference V/J genes. In the second stage of the algorithm, the overlap between reads matching only V or J gene segments is determined using a suffix tree. If the defined threshold is exceeded, the reads are concatenated to create a full CDR3. Last, CAST clustering is used to correct PCR and sequencing errors. As a result, unique clonotypes with their frequencies are determined.

TRUST4 first performs candidate TR read extraction. For reads not mapped to V, J, or C genes, significant overlap between read and reference region is determined using k-mer hits and chaining procedure (default,  $k = 9$ ). Next, *de novo* assembly of candidate reads into immune receptor region is performed. TRUST4 builds an index for all k-mers in the existing contigs and applies the seed-extension paradigm to identify alignments. Overlap between read and contig is defined as a block of at least 31-bp exact matches, while unaligned bases are outside the contig. After contig extension, TRUST4 sorts the reads by their frequency, using k-mer frequency rule. Additional parameters are predefined for the case of paired-end and barcoded data. The last step consists of annotating V, J and C genes, with correction of sequencing errors and extension of partial CDR3 sequences, if applicable.

CATT is designed especially for small-sized data with short-read lengths. The algorithm consists of four main steps, detection and assembly of CDR3 sequences, motif pattern match, error correction and gene annotation. First, reads are mapped to V/J reference genes by BWA algorithm. Both fully and partially mapped reads are considered, where the latter are used to construct potential CDR3 sequences using k-mer chaining procedure. Next, all CDR3 sequences with open reading frame and without stop codon are selected. Using an in-house data-driven procedure correction of PCR and sequencing errors is performed. Finally, annotation of V, D and J genes is performed using Bayes classifier.

MiXCR first applies an in-house procedure for read mapping to reference TR region using k-mer chaining algorithm. The parameters are set in default to handle traditional short-read data (seed length = 5). Once the best reference candidate (or few candidates) is chosen, the alignments are built using the classical Needleman-Wunsch and modified Smith-Waterman algorithms. MiXCR also allows assembling partially overlapping reads into full CDR3 contig, imputing germline sequences for good quality. After the alignment step, the assembling of clonotypes is performed. The assembler algorithm starts with extracting gene features from aligned reads and performs mapping and clustering of good quality reads. PCR and sequencing errors are corrected during these steps and low-quality reads are rescued. Multiple parameters can be controlled at every step of MiXCR algorithm.

All calculations were run on Ubuntu 20.04 server with 24x CPU and 128Gb RAM. TRUST4 and CATT algorithms were applied on raw reads (fastq file).

### 3.3 CDR3 Sequences Comparison

We compared the nucleotide sequence of identified CDR3. The distribution of annotated V and J genes obtained by the different algorithms was compared using Jensen-Shannon Divergence (JSD) metric [24]. It ranges from 0 to 1, reflecting identical and totally different distributions respectively.

## 4 Results

### 4.1 Reads Mapped to TR $\beta$ C1 and C2 Gene Region

To estimate the expected number of rearranged TR $\beta$  gene sequences in the sample and therefore the maximal number of CDR3 regions possible, we measured the number of reads mapped to the constant region of T cell receptors. In the case of beta chain, there are two C genes on chromosome 7. There were 41,094 and 42,023 reads overlapping C1 and C2 gene regions respectively in analyzed sample, representing together 0.1% of total reads. Thus, we expect only a small fraction of reads to cover CDR3 region in an analyzed sample.

### 4.2 Impact of Parameters on CDR3 Detection

First, all algorithms were run on default parameters. For ImReP and MiXCR no full CDR3 sequences were detected. TRUST4 resulted in the detection of 20,599 unique TRs, out of which 8,041 were annotated as functional (in-frame). CATT algorithm detected 1,019 unique CDR3 sequences, all functional.

Next, adjustment of parameters was performed for ImReP and MiXCR algorithms to check their impact on CDR3 detection from long reads. However, as TRUST4 and CATT do not allow the user to adjust the alignment parameters, no action was performed for the two programs.

ImReP allows users to adjust parameters of minimum overlap and maximum mismatch between read and reference sequence. Every candidate read is divided into three parts: prefix (potentially overlapping with the suffix of V gene), CDR3 region and suffix (potentially overlapping with the prefix of J gene). The parameters of matches and mismatches can be set separately for the outside and inside of the potential CDR3 region. The default values are set to 4/2 amino acids overlapping/mismatching the reference sequence and the outside of CDR3 region (prefix and suffix of the read); and 1/2 amino acids overlapping/mismatching the reference sequence and the inside of CDR3 region (right part of V gene after cysteine and left part of J gene before phenylalanine). From quality control of our data, we expect that 1 in 5 nucleotides is incorrectly detected. Thus, we increased the allowed number of mismatching amino acids outside CDR3 region to 3. This resulted in the detection of one full CDR3 sequence. Additional change of remaining parameters did not impact the final number of detected CDR3 sequences,

however, it did affect the number of reads overlapping either V or J gene region. Table 2 presents the results obtained with the best selection of parameters (5/2 matches and 3/3 mismatches for outside/inside of CDR3, respectively). ImReP also allows adjustment of minimum overlap between two reads assigned to either V or J genes to concatenate them and create a full CDR3. However, this parameter did not impact the number of detected CDR3 sequences.

MiXCR provides a unique pipeline for non-targeted genomic data obtained by traditional RNA-Seq. However, the default parameters are too strict for the low-quality of the long-read data. MiXCR algorithm allows users to adjust multiple parameters on all analysis steps, including read alignment, assembly of clonotypes and partial alignments, and gene annotation. First, we decreased the minimum alignment score required for further read processing (as we expect more mismatches in the case of ONT sequences in reference to traditional RNA-Seq). This action did increase the number of analyzed sequences but did not change the final number of full CDR3s. Worth noticing that sequences may be extracted after the alignment step for detailed inspection. Next, we increased the length of seed used in the aligner to 9, for we are working with long reads. This highly decreased the time taken to complete the analysis (about 8x faster work when compared to default seed length = 5) but, again, did not impact the final number of CDR3 sequences. The next step included assembly and extension of partial alignments. Again, changing the parameters into less strict or not performing assembly of partial alignments did not impact the final number of detected CDR3 sequences. Worth noticing that MiXCR does not allow mismatches within the overlap of partially aligned sequences. Therefore, it is hard to recover TR sequences from the sample with low sequencing quality. Finally, assembly of clonotypes using alignments obtained in previous steps was performed. Lowering the sequencing quality threshold of nucleotides to 7 (which is the median sequencing quality of our data) allowed the detection of 10 CDR3 sequences. The change of the remaining parameters did not affect the result.

Table 2 shows the results of CDR3 identification for the four algorithms using the best selection of parameters. Because ImReP does not allow parallelization, we measured the running time of all algorithms using one thread only. MiXCR and TRUST4 were the fastest algorithms, where the latter had a minor average memory consumption. ImReP was also very time efficient (due to the alignment-free procedure). However, it required the most extensive memory reservoir. CATT was running the longest on one thread. Worth noticing that the parallelization of algorithms strongly shortened the working time of CATT, TRUST4 and MiXCR algorithms.



**Table 2.** Results of CDR3 detection from long reads. Parameters for ImReP and MiXCR were adjusted to obtain the best results. \*Number of sequences detected after alignment step of MiXCR algorithm.

	ImReP	TRUST4	CATT	MiXCR
Number of unique full CDR3s	1	20,599	1,019	10
Number of functional CDR3s	1	8,041	1,019	5
Number of partial CDR3s–V/J genes	209,923 / 1,128	Not reported	Not reported	3,459* / 3,459*
Running time of an algorithm on 1 thread	184m 33s	129m 39s	3,654m 50s	128m 34s
Average memory consumption	18.0Gb	1.8Gb	3.3Gb	6.2Gb

### 4.3 Analysis of Detected CDR3s

ImReP returned one CDR3 sequence, which consisted of only four amino acids (CASF). Any of the remaining algorithms did not identify this sequence. Interestingly number of sequences with annotated only V or only J gene regions was much higher (209,923 and 1,128 of V and J genes, respectively). The number of sequences with identified V gene highly exceeds the number of sequences with annotated J genes.

MiXCR detected 10 CDR3 sequences, out of which half were functional. For all the sequences, both V and J genes were annotated. Interestingly, there were 3,459 candidate sequences returned by the algorithm after the alignment step. All these sequences were initially annotated with both V and J gene, some of them had multiple possible V or J genes annotated. It must be noted that the D gene was sometimes erroneously identified as an Ig D genes (about 35% of all D gene annotated sequences). 1,198 sequences were inferred to be in-frame. Due to the low quality, sequences did not pass further steps of an algorithm.

CATT returns only functional sequences, as those without an open reading frame or with stop codon are filtered out. 1,019 sequences were detected; however, only 20 had V and J genes annotated simultaneously. Most sequences were annotated with J genes (757 reads), and about one-third of sequences had V gene annotated (282 reads).

TRUST4 detected the biggest number of CDR3 sequences among all analyzed algorithms. Almost all sequences were annotated with V and J genes at the same time, and only 124 sequences did not have either gene classified. Most of the sequences were identified in only one copy, the biggest number of copies for a unique CDR3 sequence equaled 32. The majority of sequences were out-of-frame or contained a stop codon.

The number of detected CDR3 sequences is lower than expected for all algorithms. Based on an analysis of reads mapped to TR $\beta$  C1 and C2 gene regions, we expected about 80 thousand CDR3 sequences in an analyzed sample. TRUST4 provided the best result; however, it covers only 25% of the expected number of sequences.

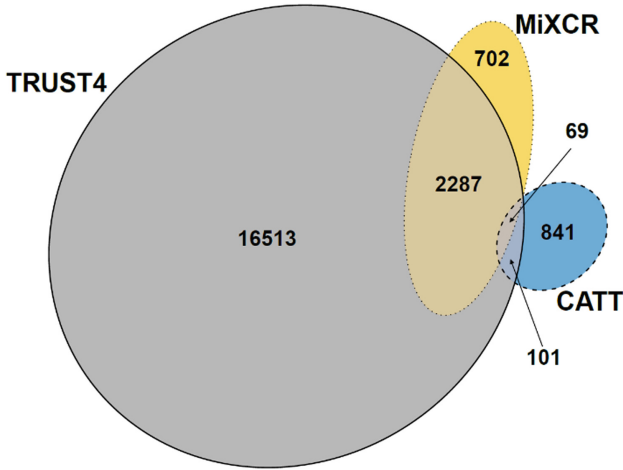
Next, we compared the CDR3 sequences obtained by CATT, TRUST4 and the first step alignments of MiXCR. For every algorithm, we found some duplicated CDR3 sequences (Table 3). For TRUST4 and MiXCR about 10% of CDR3 sequences did not start with cysteine, which is required for the correct conformation of TR. After filtration of these sequences, we were left with the following numbers of unique CDR3 sequences for CATT: 1,011, MiXCR: 3,058, and TRUST4: 18,970. Figure 2 shows the number of common CDR3 sequences between the algorithms. There were only 69 clonotypes identical among all algorithms. Most of the sequences detected by MiXCR were in common with TRUST4, whereas the majority of CATT and TRUST4 sequences were unique for the algorithms. For the 69 common clonotypes, annotation of V and J gene names was identical for MiXCR and TRUST4 algorithms. CATT did not identify most of the J genes and proposed different annotations of the V gene for eight sequences. Furthermore, an inspection of CDR3 sequences reported by CATT showed that most of them include the insertion of thymine nucleotides (reaching up to 45 T nucleotides stretches) at the end of the CDR3 sequence. These nucleotides are not present in J genes and most probably come from sequencing errors.

**Table 3.** Number of CDR3 sequences detected by TRUST4, CATT and first step alignments from MiXCR.

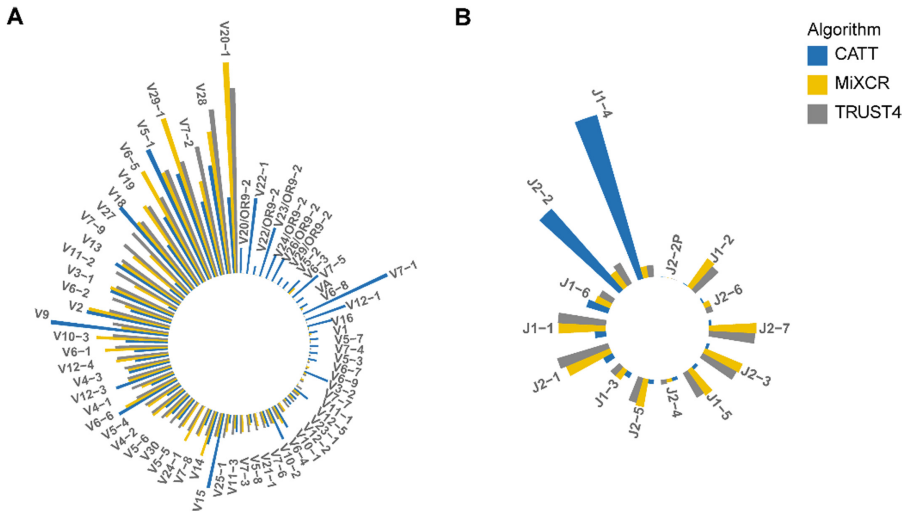
	TRUST4	CATT	MiXCR
All CDR3 sequences	20,599	1,019	3,459
CDR3 duplicates	13	8	27
Unique CDR3 sequences	20,586	1,011	3,432
Sequences starting with TGT codon	13,871	657	2,249
Sequences starting with TGC codon	5,099	354	809
<b>Sequences starting with Cysteine</b>	<b>18,970</b>	<b>1,011</b>	<b>3,058</b>
% of sequences starting with Cysteine	92.15%	100.00%	89.10%

Prompted by the high similarity in common V gene distribution between the algorithms, we investigated V and J gene frequencies for all the CDR3 sequences starting with cysteine. Distribution of V and J genes is almost identical between MiXCR and TRUST4 (Fig. 3;  $JSD.V = 0.01$  and  $JSD.J = 0.00$ ). CATT reports quite different V gene composition ( $JSD.V = 0.17$  for MiXCR and  $0.18$  for TRUST4) and very different J gene composition ( $JSD.J = 0.49$  for MiXCR and  $0.47$  for TRUST4).

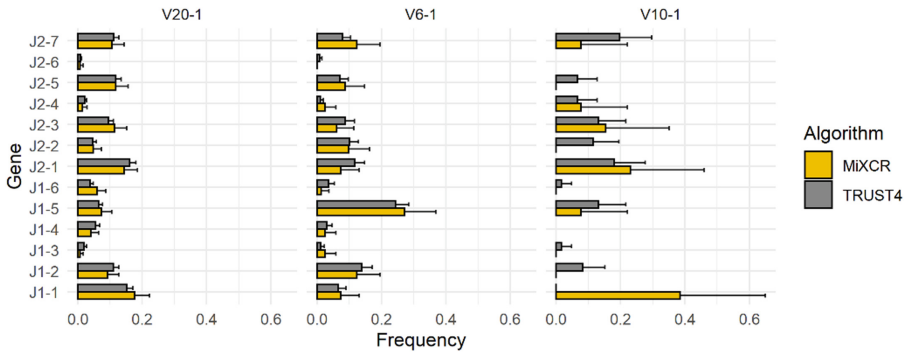
To further investigate high similarity of results obtained by TRUST4 and MiXCR algorithms, we compared the J gene repertoire used by selected high, medium, and low-frequency V genes, namely V20-1, V6-1 and V10-1. As expected, we observed that the more frequent the V gene, the more similar was the J gene distribution (Fig. 4,  $JSD.high = 0.01$ ,  $JSD.medium = 0.02$ ,  $JSD.low = 0.37$ ). For the low-frequency V10-1 gene, only 14 CDR3 sequences were reported by MiXCR, causing the observed bias in J gene distribution.



**Fig. 2.** Common unique nucleotide CDR3 sequences detected by the algorithms. Sequences after first alignment step of MiXCR were considered. ImReP results were not included in the figure. Figure created with <http://eulerr.co/> [25].



**Fig. 3.** V and J gene frequencies reported by the algorithms. **A**–V gene frequencies. **B**–J gene frequencies.



**Fig. 4.** J gene distribution used by high-frequency V20–1, medium-frequency V6–1 and low-frequency V10–1 genes reported by MiXCR and TRUST4 algorithms.

## 5 Discussion

Nanopore sequencing is a high throughput and low-cost technology providing reads of kilobase pairs length. However, due to the difficulty in identifying DNA bases from complex electrical signals [15] it is also characterized by a high error rate. Data used in the following analysis had a sequencing quality of 80%, meaning that one in five nucleotides is most probably incorrectly sequenced. This high error rate affects the identification of highly variable CDR3 regions from rearranged Ig and TR genes. For that, it is worth to consider applying error correction methods, like [15] or [26], as a preprocessing step.

This work aimed to apply existing algorithms for CDR3 detection from bulk RNA-Seq on data with long reads. The following algorithms were checked: ImReP, TRUST4, CATT and MiXCR. They use different methods for TR $\beta$  sequence identification, some require previous alignment (ImReP), and others apply an in-house procedure for read mapping (TRUST4, CATT and MiXCR). Furthermore, ImReP and MiXCR allow users for custom parameter adjustment, whereas CATT and TRUST4 provide ready pipelines for different types of input data.

ImReP algorithm performs identification of V/J genes based on amino-acid sequences, without previous read alignment to reference region. This approach ensures a high speed of the algorithm, however, it requires reads with high sequencing quality. In the case of data with 80% sequencing accuracy, about 60% of amino acids might be incorrectly identified due to the falsely detected nucleotide. This affects the identification of full CDR3s based on reads overlapping both V and J genes and reads partially overlapping CDR3 region. In the latter case, finding the exact overlap between the two candidate sequences is a huge challenge.

The remaining algorithms first apply read alignment to the TR $\beta$  region using nucleotide sequences before the identification of CDR3 region is performed. This step allows for more efficient extraction of candidate reads, confirmed by a much higher number of final CDR3s reported by TRUST4, CATT or MiXCR (considering first step alignments).

CATT reported 1,019 final CDR3 sequences; however, only 20 had V and J genes assigned. What is more, it only returns in-frame sequences, which do not represent the

whole TR repertoire. Having cDNA data, we expect at most 10% of sequences to be out of frame [27], so restriction to only functional sequences causes a lack of information. In addition, sequences reported by CATT include excessive stretches of T nucleotides, resulting in amino acid sequence with tandem phenylalanine. This most probably is a sequencing error, and such sequences should be discarded. Finally, as reported by the authors, CATT is specially designed for data with short read length and small size, which are both the opposite of Nanopore Sequencing. Because the parameters are not adjustable for the user, long-read data cannot therefore be used with CATT.

MiXCR algorithm is the most adjustable one, allowing for optimizing multiple parameters at every step of the procedure. Lowering the quality score of alignments and increasing the seed length speeded up the algorithm and allowed for the detection of 10 full CDR3s. However, an inspection of first step alignments showed additional 3,447 CDR3 sequences with annotated both V and J genes. Detailed analysis of these sequences showed that they could be utilized to analyze the V and J gene repertoire used in rearranged TR $\beta$  genes.

TRUST4 resulted in the biggest number of detected CDR3 sequences, even though it does not allow users to adjust parameters. The number of identified CDR3 sequences represented only 25% of the expected number of sequences; however, it still was six times more efficient than MiXCR first step alignments.

For TRUST4 results and MiXCR first step alignments, the functional status of sequences was mostly out-of-frame, which is not expected in the case of cDNA samples [27]. Here, the proportion of functional sequences should highly exceed the fraction of nonfunctional ones. This high number of out-of-frame sequences might be caused by a high rate of sequencing errors, which either introduce indels shifting the reading frame or introduce substitutions resulting in a stop codon. Applying a custom post-processing algorithm to correct the frameshifts and duplicates in the data might be worth considering. Also, clustering of highly similar sequences might be a way to go.

What is more, TRUST4 and MiXCR can be utilized for the detection of V and J genes distributions. As the vast majority of sequences identified by these algorithms were annotated with both V and J genes, we can use this information to analyze V and J gene diversity in a given sample and to compare it between samples. Furthermore, the information about the distribution of V genes might give insights into the evolution of TR repertoire, thus obtaining this knowledge from bulk RNA-Seq long reads is very useful. This work shows that TRUST4 and MiXCR can be used to provide information about the evolution of the TR repertoire from bulk RNA-Seq long reads, precious information in pathophysiological conditions.

**Acknowledgment.** This work was funded by the European Social Fund grant POWR.03.02.00–00-I029 and by the Silesian University of Technology grant for Support and Development of Research Potential.

## References

1. Mose, L.E., Selitsky, S.R., Bixby, L.M., et al.: Assembly-based inference of B-cell receptor repertoires from short read RNA sequencing data with V'DJer. *Bioinformatics* **32**(24), 3729–3734 (2016). <https://doi.org/10.1093/bioinformatics/btw526>

2. Canzar, S., Neu, K.E., Tang, Q., Wilson, P.C., Khan, A.A.: BASIC: BCR assembly from single cells. *Bioinformatics* **33**(3), 425–427 (2017). <https://doi.org/10.1093/bioinformatics/btw631>
3. Upadhyay, A.A., Kauffman, R.C., Wolabaugh, A.N., et al.: BALDR: A computational pipeline for paired heavy and light chain immunoglobulin reconstruction in single-cell RNA-seq data. *Genome Med.* **10**, 20 (2018). <https://doi.org/10.1186/s13073-018-0528-3>
4. Ye, J., Ma, N., Madden, T.L., Ostell, J.M.: IgBLAST: An immunoglobulin variable domain sequence analysis tool. *Nucleic Acids Res* **41**(Web Server issue), W34–W40 (2013)
5. Stubbington, M.J.T., Lönnberg, T., Proserpio, V., et al.: T cell fate and clonality inference from single-cell transcriptomes. *Nat. Methods* **13**(4), 329–332 (2016). <https://doi.org/10.1038/nmeth.3800>
6. Gerritsen, B., Pandit, A., Andeweg, A.C., de Boer, R.J.: RTCR: A pipeline for complete and accurate recovery of T cell repertoires from high throughput sequencing data. *Bioinformatics (Oxford, England)* **32**(20), 3098–3106 (2016). <https://doi.org/10.1093/bioinformatics/btw339>
7. Mandric, I., Rotman, J., Yang, H.T., et al.: Profiling immunoglobulin repertoires across multiple human tissues using RNA sequencing. *Nat. Commun.* **11**, 3126 (2020). <https://doi.org/10.1038/s41467-020-16857-7>
8. Song, L., Cohen, D., Ouyang, Z., et al.: TRUST4: Immune repertoire reconstruction from bulk and single-cell RNA-seq data. *Nat. Methods* **18**, 627–630 (2021). <https://doi.org/10.1038/s41592-021-01142-2>
9. Kuchenbecker, L., et al.: IMSEQ—a fast and error aware approach to immunogenetic sequence analysis. *Bioinformatics* **31**(18), 2963–2971 (2015). <https://doi.org/10.1093/bioinformatics/btv309>
10. Bolotin, D., Poslavsky, S., Mitrophanov, I., et al.: MiXCR: Software for comprehensive adaptive immunity profiling. *Nat. Methods* **12**, 380–381 (2015). <https://doi.org/10.1038/nmeth.3364>
11. Rizzetto, S., Koppstein, D.N.P., Samir, J., et al.: B-cell receptor reconstruction from single-cell RNA-seq with VDJpuzzle. *Bioinformatics* **34**(16), 2846–2847 (2018). <https://doi.org/10.1093/bioinformatics/bty203>
12. Chen, S.-Y., Liu, C.-J., Zhang, Q., Guo, A.-Y.: An ultra-sensitive T-cell receptor detection method for TCR-Seq and RNA-Seq data. *Bioinformatics* **36**(15), 4255–4262 (2020). <https://doi.org/10.1093/bioinformatics/btaa432>
13. Yu, Y., Ceredig, R., Seoighe, C.: LymAnalyzer: A tool for comprehensive analysis of next generation sequencing data of T cell receptors and immunoglobulins. *Nucleic Acids Res.* **44**(4), e31 (2016). <https://doi.org/10.1093/nar/gkv1016>. Epub 2015 Oct 7. PMID: 26446988; PMCID: PMC4770197
14. Yang, X., et al.: TCRklass: A new K-string-based algorithm for human and mouse TCR repertoire characterization. *J. Immunol.* **194**(1), 446–454 (2015). <https://doi.org/10.4049/jimmunol.1400711>
15. Wang, L., Qu, L., Yang, L., Wang, Y., Zhu, H.: NanoReviser: An error-correction tool for nanopore sequencing based on a deep learning algorithm. *Front. Genet.* **12**(11), 900 (2020). <https://doi.org/10.3389/fgene.2020.00900>
16. Sahlin, K., Medvedev, P.: Error correction enables use of Oxford nanopore technology for reference-free transcriptome analysis. *Nat. Commun.* **12**, 2 (2021). <https://doi.org/10.1038/s41467-020-20340-8>
17. Dohm, J.C., Peters, P., Stralis-Pavese, N., Himmelbauer, H.: Benchmarking of long-read correction methods. *NAR Genomics Bioinformatics* **2**(2), lqaa037 (2020). <https://doi.org/10.1093/nargab/lqaa037>
18. Cruz-Garcia, L., et al.: Generation of a transcriptional radiation exposure signature in human blood using long-read nanopore sequencing. *Radiat. Res.* **193**(2), 143–154 (2020). <https://doi.org/10.1667/RR15476.1>

19. de Coster, W., D'Hert, S., Schultz, D.T., Cruts, M., van Broeckhoven, C.: NanoPack: Visualizing and processing long-read sequencing data. *Bioinformatics* **34**(15), 2666–2669 (2018). <https://doi.org/10.1093/bioinformatics/bty149>
20. Li, H.: Minimap2: Pairwise alignment for nucleotide sequences. *Bioinformatics* **34**(18), 3094–3100 (2018). <https://doi.org/10.1093/bioinformatics/bty191>
21. Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R.: The Sequence alignment/map (SAM) format and SAMtools. *Bioinformatics* **25**, 2078–2079 (2009). 1000 Genome Project Data Processing Subgroup (2009)
22. Morgan, M., Pagès, H., Obenchain, V., Hayden, N.: Rsamtools: Binary alignment (BAM), FASTA, variant call (BCF), and tabix file import (2021). R package version 2.8.0, <https://bioconductor.org/packages/Rsamtools>
23. Lefranc, M.-P.: IMGT, the international ImMunoGeneTics information system. *Cold Spring Harb. Protoc.* **2011**(6), pp. pdb-top115, 2011 Jun 1. DOI:<https://doi.org/10.1101/pdb.top115>. <http://www.imgt.org/FAQ/#question15>
24. Lin, J.: Divergence measures based on the shannon entropy. *IEEE Trans. Inf. Theory* **33**(1), 145–151 (1991)
25. Larsson, J.: Eulerr: Area-Proportional Euler and Venn Diagrams with Ellipses (2020). R package version 6.1.0, <https://cran.r-project.org/package=eulerr>
26. Vaser, R., Sović, I., Nagarajan, N., Šikić, M.: Fast and accurate de novo genome assembly from long uncorrected reads. *Genome Res.* **27**(5), 737–746 (2017). <https://doi.org/10.1101/gr.214270.116>. Epub 2017 Jan 18
27. Li, S., Wilkinson, M.F.: Nonsense surveillance in lymphocytes? *Immun.* **8**(2), 135–141 (1998). [https://doi.org/10.1016/s1074-7613\(00\)80466-5](https://doi.org/10.1016/s1074-7613(00)80466-5)

## Author Index

- Abramov, Vladislav I-205  
Aguilera, Concepción María I-42, II-359  
Alcalá, Rafael II-359  
Alcalá-Fdez, Jesús I-42, II-359  
Alé, Anibal I-343  
Al-enezi, Mamdouh S. I-184  
Alija-Pérez, José-Manuel I-417  
Alió, Jorge I-108, I-119  
Álvarez, Rubén I-417  
Amato, Federica I-381  
Anderson, Paul II-249  
Anguita-Ruiz, Augusto I-42, II-359  
Aridhi, Sabeur II-153
- Badie, Christophe II-450  
Bakalova, Snezhana M. I-216  
Bandyopadhyay, Oishila I-364  
Bardini, Roberta II-18, II-179  
Barraso, Marina I-343  
Barton, Vojtech II-288  
Benali, Anass I-343  
Benes, Jakub II-319  
Benso, Alfredo I-395  
Bentourkia, M'hamed I-3, I-184, I-309  
Berciano-Guerrero, Miguel-Ángel I-319  
Bernal, Carolina I-343  
Blahuta, Jiri I-283  
Bohniková, Alžbeta II-220  
Bonizzoni, Paola II-436  
Bugáňová, Alžbeta I-28  
Bustos-Aibar, Mireia I-42, II-359
- Calderón-Ramírez, Saúl II-375  
Cámara-Sánchez, Sofía II-359  
Candéias, Serge M. II-450  
Carballido, Jessica A. II-90  
Carrera, Laura I-343  
Castillo-Secilla, Daniel I-404  
Cavas, Francisco I-108, I-119  
Cecchini, Rocío L. II-90  
Celi, Simona II-208  
Chatzikyrkou, Konstantina I-135  
Chavez-Monjaras, Sandra M. I-75  
Choudhury, Alokeparna I-364
- Christin, Ann I-343  
Cicolin, Alessandro I-381  
Cimrák, Ivan I-28, II-220  
Costanzo, Manuel II-103  
Cruz-Albarran, Irving A. I-75  
Cvetković, Izabel I-453
- Damigos, Gerasimos I-135  
Dario León Bueno de Camargo, Erick I-154  
Davidson, Jean II-249  
de Souza, Leozítor Floro I-241  
Depuydt, Lore II-419  
Devignes, Marie-Dominique II-153  
Di Carlo, Stefano II-18, II-179  
Díaz-Jiménez, David I-297  
Díez-González, Javier I-417  
Dobrovolny, Michal II-319  
Dóczy, Tamás I-269  
Domínguez, Enrique II-375  
Doud, Andrew II-249  
du Plessis-Burger, Nelita II-399  
Durawa, Agata I-357  
Dziadziuszkó, Katarzyna I-357
- Echenne, Bernard I-309  
Efroni, Sol II-346  
El Hage, Rawad I-90  
Esmatjes, Enric I-343  
Espinilla, Macarena I-297  
Exarchos, T. II-83
- Farmer, Marie I-309  
Feng, James J. II-220  
Ferrero-Guillén, Rubén I-417  
Feu, Silvia I-343  
Fostier, Jan II-419  
Fuentes-Fino, Ricardo Javier II-375  
Fulop, Tamas I-184
- García-Nieto, José I-227, I-319  
García-Sánchez, Carlos II-103  
Garzón, Ester M. II-234  
Gasparotti, Emanuele II-208  
Georgieva, Milena I-216



- Georgieva, Olga II-389  
 Giannantoni, Leonardo II-179  
 Gimenez, Marga I-343  
 Glodek, Anna II-119  
 Gómez, Carmelo I-108, I-119  
 Gonçalves, Douglas S. II-142  
 Gorrab, Siwar I-171  
 Grønli, Tor-Morten I-256  
 Guillén, Alberto I-404  
 Gutiérrez-Mondragón, Mario A. II-275  
 Guyeux, Christophe II-300
- Hadziahmetovic, Armin I-429  
 Hawblitzel, Grif II-249  
 Hejmel, László I-269  
 Hernandez, Teresa I-343  
 Hernandez-Vasquez, Marco A. II-375  
 Herrera, Luis Javier I-404, I-442  
 Herrero, Laura I-42  
 Hora, Sheena II-333  
 Hurtado, Sandro I-227, I-319
- Iaione, Fábio I-241  
 Ivanov, Ivan II-57
- Jakusovszky, Ava II-249  
 Jančigová, Iveta II-220  
 Janszky, József I-269  
 Jat, Avnish Singh I-256  
 Jelitto-Gorska, Malgorzata I-357  
 Jennane, Rachid I-90  
 Joppich, Markus I-429  
 Jovančević, Ana I-453  
 Ju, Shih Ting I-241
- Kaneti, Jose I-216  
 Khalil, Abdelouahed I-184  
 Khawaja, Anthony I-90  
 Kim, Paul II-249  
 König, Caroline II-275  
 Konur, Savas II-193  
 Koumenti, Argyro I-135  
 Kovalčíková Ďuračiková, Kristína I-28  
 Kratochvil, Miroslav II-353  
 Krejcar, Ondrej II-319  
 Kressman, McClain II-249
- Lahiri, Chandrajit II-413  
 Lakshman Kumar, Harsha II-249  
 Laky, Norbert I-269
- Lamghari, Youssef I-3  
 Larionova, Irina I-205  
 Lavrincik, Jan I-283  
 Ledoux, Julie II-133, II-138  
 Livne, Dani II-346  
 López-Rubio, Ezequiel II-375  
 López-Ruiz, José Luis I-297  
 Loubopoulos, Athanasios I-135  
 Lu, Huizhong I-3  
 Lupión, Marcos I-330  
 Luppi Silva, Olavo I-154  
 Lytaev, Sergey I-143
- Maliha, Elie I-90  
 Marchese, Pietro II-208  
 Marczyk, Michal II-33, II-71  
 Marín, Sara I-343  
 Mariotti, Alessandro II-208  
 Marques dos Santos, José Diogo II-260  
 Marques dos Santos, José Paulo II-260  
 Martín, Ruben I-343  
 Martínez-Gutiérrez, Alberto I-417  
 Martini, Lorenzo II-18  
 Mascaró, Marilina II-90  
 Matzko, Richard Oliver II-193  
 Medina-Quero, Javier I-330  
 Melnik, Roderick I-47, I-59  
 Micsinyei, László I-269  
 Mierla, Laurentiu II-193  
 Mika, Justyna II-450  
 Miličević, Nebojša I-453  
 Mira, Jorge I-119  
 Mittal, Karuna II-413  
 Molina-Cabello, Miguel A. II-375  
 Molnár, Balázs I-269  
 Montoro-Lendínez, Alicia I-297  
 Morales, Juan Carlos I-442  
 Morales-Hernandez, Arelly G. I-75  
 Morales-Hernandez, Luis A. I-75  
 Morgun, Andrey I-205  
 Mourouzis, Jordanis I-135  
 Moustakas, Konstantinos I-135  
 Mrukwa, Anna II-33  
 Mucherino, Antonio II-142
- Naiouf, Marcelo II-103  
 Navas-Delgado, Ismael I-227, I-319  
 Nedorez, Yana II-169  
 Nematzadeh, Hossein I-319  
 Noura, Kaouther I-171

- Oliva, Cristian I-343  
 Olmo, Gabriella I-381  
 Orsi, Gergely I-269  
 Ortega, Emilio I-343  
 Ortigosa, Pilar M. I-330, II-234  
 Ortiz, Sergio I-442  
 Ostellino, Sofia I-395
- Pal, Swadesh I-47  
 Palejev, Dean II-57  
 Pantos, Constantinos I-135  
 Patsiris, S. II-83  
 Pavlik, Lukas I-283  
 Pavlopoulos, Angelos I-135  
 Pawar, Shrikant II-413  
 Pérez-Sánchez, Horacio II-234  
 Perlaki, Gábor I-269  
 Petescia, Alessia II-436  
 Petrini, Iván II-90  
 Pierides, Iro I-16  
 Piñero, David I-108  
 Pinti, Antonio I-90  
 Pirola, Yuri II-436  
 Polanska, Joanna I-357, II-309, II-399,  
 II-450  
 Politano, Gianfranco I-395  
 Polo-Rodríguez, Aurora I-330  
 Ponzoni, Ignacio II-90  
 Pratihar, Sanjoy I-364  
 Prazuch, Wojciech I-357  
 Prieto-Matías, Manuel II-103  
 Puertas-Martín, Savíns II-234
- Qazi, Sahar II-3
- Raza, Khalid II-3  
 Rechichi, Irene I-381  
 Redondo, Juana L. II-234  
 Redondo-Sánchez, Daniel I-404  
 Refrégier, Guislaine II-300  
 Rejab, Fahmi Ben I-171  
 Renders, Luca II-419  
 Repaska, Zuzana I-283  
 Reyes-Farias, Marjorie I-42  
 Rizzi, Raffaella II-436  
 Rodrigues, Edson I-154  
 Rojas, Fernando I-442  
 Rojas, Ignacio I-404, I-442  
 Romero, Enrique I-343  
 Rosales-Hernandez, Andrea I-75
- Rosinés, Josep I-343  
 Rucci, Enzo II-103  
 Rueda, Luis II-333  
 Ruiz-Ojeda, Francisco Javier I-42
- Sadovsky, Michael I-197, I-205, II-169  
 Sáez-Gutiérrez, Francisco L. I-108, I-119  
 Salvetti, Maria Vittoria II-208  
 Samanta, Sourav I-364  
 Sánchez-Infantes, David I-42  
 Sarker, Bishnu II-153  
 Satagopam, Venkata II-353  
 Savino, Alessandro II-18  
 Schneider, Reinhard II-353  
 Schubö, Alexandra I-429  
 Schwarzerova, Jana I-16  
 Sedlar, Karel I-16, II-45  
 Selamat, Ali II-319  
 Sellami, Akrem II-153  
 Senashova, Maria I-197  
 Shakola, Felitsiya II-57  
 Skutkova, Helena II-288  
 Slowik, Hanna II-71  
 Sola, Christophe II-300  
 Soria-Gondek, Andrea I-42  
 Soukup, Tomas I-283  
 Stolyarchuk, Maxim II-138  
 Suwalska, Aleksandra II-399
- Tabbone, Salvatore II-153  
 Tchernanov, Luba II-133, II-138  
 Tényi, Ákos I-269  
 Teterleva, Agnia I-205  
 Thieu, Thi Kim Thoa I-59  
 Thomas, Ella II-249  
 Tobiasz, Joanna II-309  
 Torres-Martos, Álvaro I-42, II-359  
 Toumi, Hechmi I-90  
 Trivedi, Yash II-333  
 Turcotte, Éric I-184
- Valenzuela, Olga I-442  
 van der Spuy, Gian II-399  
 Vasighizaker, Akram II-333  
 Vega, Carlos II-353  
 Velázquez, José S. I-108, I-119  
 Vellido, Alfredo I-343, II-275  
 Verde, Paula I-417  
 Vignali, Emanuele II-208

Viguera-Becerril, Daniela [I-75](#)

Vilá, Irene [I-343](#)

Vinagre, Irene [I-343](#)

Vlamos, P. [II-83](#)

Weckwerth, Wolfram [I-16](#)

Zaccagnino, Rocco [II-436](#)

Zacharaki, Evangelia I. [I-135](#)

Zarranz-Ventura, Javier [I-343](#)

Zerva, Nefeli [I-135](#)

Zimmer, Ralf [I-429](#), [II-45](#)

Zizza, Rosalba [II-436](#)

Zyla, Joanna [II-33](#), [II-71](#)