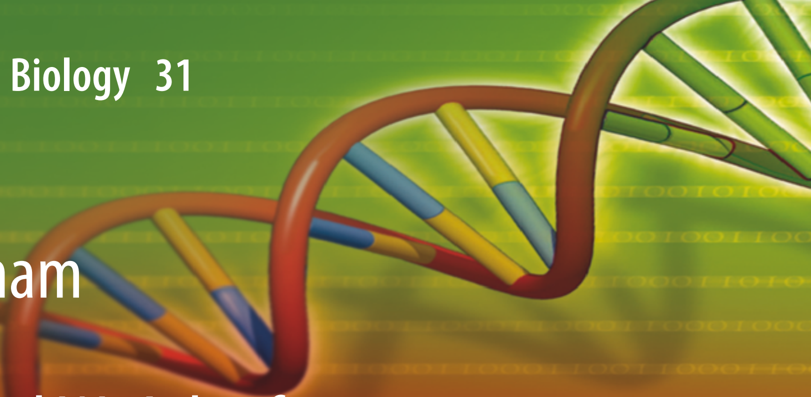
Computational Biology 31



Tuan D. Pham
Hong Yan
Muhammad W. Ashraf
Folke Sjöberg *Editors*

Advances in Artificial Intelligence, Computation, and Data Science

For Medicine and Life Science

iSCB
INTERNATIONAL SOCIETY FOR
COMPUTATIONAL BIOLOGY

Springer

Computational Biology

Volume 31

Endorsed by the *International Society for Computational Biology*, the *Computational Biology* series publishes the very latest, high-quality research devoted to specific issues in computer-assisted analysis of biological data. The main emphasis is on current scientific developments and innovative techniques in computational biology (bioinformatics), bringing to light methods from mathematics, statistics and computer science that directly address biological problems currently under investigation.

The series offers publications that present the state-of-the-art regarding the problems in question; show computational biology/bioinformatics methods at work; and finally discuss anticipated demands regarding developments in future methodology. Titles can range from focused monographs, to undergraduate and graduate textbooks, and professional text/reference works.

More information about this series at http://www.springer.com/series/5769

Tuan D. Pham · Hong Yan ·
Muhammad W. Ashraf · Folke Sjöberg
Editors

Advances in Artificial Intelligence, Computation, and Data Science

For Medicine and Life Science

Springer

Editors
Tuan D. Pham
Center for Artificial Intelligence
Prince Mohammad Bin Fahd University
Khobar, Saudi Arabia

Hong Yan
Department of Electrical Engineering
City University of Hong Kong
Kowloon, Hong Kong

Muhammad W. Ashraf
College of Sciences and Human Studies
Prince Mohammad Bin Fahd University
Khobar, Saudi Arabia

Folke Sjöberg
Department of Biomedical and Clinical Sciences
Linköping University
Linköping, Sweden

ISSN 1568-2684 ISSN 2662-2432 (electronic)
Computational Biology
ISBN 978-3-030-69950-5 ISBN 978-3-030-69951-2 (eBook)
https://doi.org/10.1007/978-3-030-69951-2

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

Preface

Advanced methods recently developed in artificial intelligence (AI), computation, and data science have contributed to many major breakthroughs in medicine and life sciences. The integration of new medical knowledge and smart computer algorithms is making significant progress toward personalized medicine, precision medicine, and personalized healthcare. An example is the use of AI, computation, and data science for intelligent natural language processing that recognizes patients' answers to medical doctors' questions then extracts and stores critical information into electronic health records for constructing structured medical reports [1], and electronic health records can be utilized for clinical predictive modeling using deep learning [2]. It was reported [3] that recurrent neural networks were able to decode speech from neural activity recorded from a sensor implanted in a human brain and transformed the cortical signals into a synthetic voice for clinical assessment and treatment. This technology is helpful for patients who suffer from neurological paralysis, for example, because of stroke, traumatic brain injury, and neurodegenerative diseases.

Current AI, computation, and data science have made significant advances in medical research toward precision and personalized medicine, particularly in oncology, healthcare, and surgery [4–9]. The precision or personalized medicine approach will allow medical doctors to predict more accurately which treatment and prevention strategies for a disease will be effective for which groups of people. It contrasts with the one-size-fits-all approach, in which disease treatment and prevention strategies are developed for the average person, with less consideration for the differences between individuals. Computer-aided surgery can guide surgeons to perform accurately on patients over the areas of interest that even experienced surgeons fail to detect. Figure 1 shows our point of view about the roles of AI, computation, and data science in medicine and life sciences, which can help turn the concept of personalized and precision medicine into reality.

After the recovery from periods of the AI winter, which was coined to be analogous to the idea of a nuclear winter to indicate a time of serious setback for funding and research interest in AI [10], it is witnessed that AI contributes to significant advances in many scientific fields and making societal impacts. This is mainly due to the birth of deep learning methods [11, 12]. Medicine and life science are among areas of research

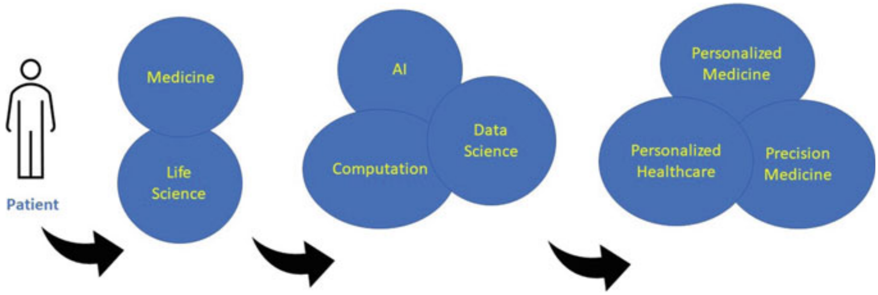


Fig. 1 Point of view about how personalized medicine, precision medicine, and personalized healthcare can be achieved with the role of AI, computation, and data science in medicine and life sciences

that AI is considered to lead the foremost successful applications [13–15], and on top of these areas is the role of AI in cancer study. Over the past decade, AI has been useful for solving problems in basic and clinical cancer research, where solutions to such problems were thought to be impossible due to the limitation of human-based analysis of large volumes of complex biomedical data. A major contribution of AI to cancer is its power to provide fast and accurate diagnostics [16], for example, where AI could assist a medical specialist to detect a cancerous brain tumor correctly in a child as glioblastoma instead of medulloblastoma that was first thought by a neuropathologist. As a result, correct treatment could be recommended to the patient to prevent adverse effects in addition to the failure to kill the cancer.

Another significant contribution of AI to clinical cancer research is prediction and prognosis [17]. Prediction of survival of patients with brain tumors based on tissue and genomic data with state-of-the-art AI known as convolutional neural networks was reported in [18]. This study used the AI method to learn the visual patterns and molecular biomarkers associated with the patient outcomes. As a result, the AI method outperformed the prognosis carried out by conventional clinical standards for brain-tumor classification. Furthermore, the use of pretrained deep neural networks was able to discover the power of a protein biomarker known as DNp73 in rectal cancer [19]. The networks were trained and validated with immunohistochemistry (IHC) imaging data of DNp73 expression on biopsy and surgically resected tumor samples collected from a cohort of 143 rectal cancer patients from the Swedish rectal cancer trial who were either randomly selected for preoperative radiotherapy (pRT) or without pRT. While conventional pathology analysis of the DNp73 expression failed to provide the correlation with the survival rate, the AI-based approach achieved very high accuracy rates for the 5-year prediction and prognosis of the rectal cancer patients either with or without pRT. Figure 2 shows the manual pathology-based quantification of DNp73 expression on IHC imaging, which meets with difficulty in finding the correlation between the protein biomarker and survival information. In cancer therapy, AI is thought to have the potential in many aspects such as drug discovery, development, and validation [20]. Finding the optimal drug for cancer treatment is an expensive and time-consuming process. The augmentation of cancer

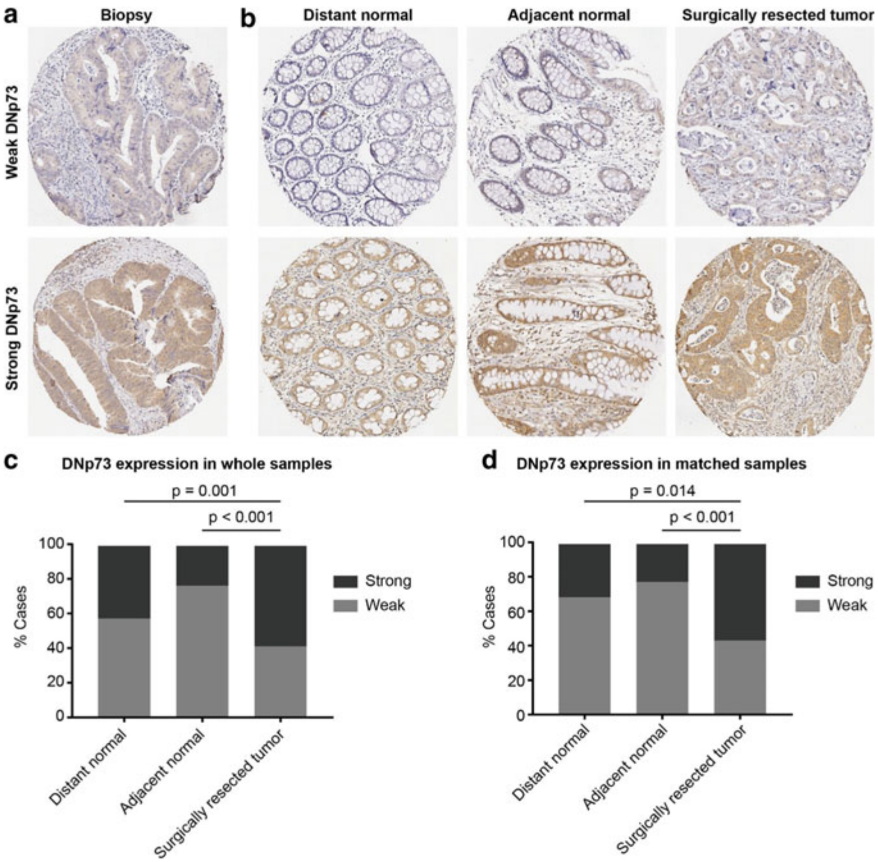


Fig. 2 DNp73 expression by immunohistochemistry staining in tumor samples from rectal cancer patients. A representative immunohistochemistry image of DNp73 expression in biopsies (a) and surgically resected samples, including distant normal mucosa, adjacent normal mucosa, and surgically resected tumor (b); DNp73 expression in distant normal mucosa, adjacent normal mucosa, and surgical tumor obtained from whole samples (c), and matched samples (d). Whole samples indicated all surgically resected samples. Matched samples included surgically resected samples (including distant normal, adjacent normal and primary tumor samples) from the same patient (reused from open access article published under the terms of the Creative Commons Attribution License [19])

therapy with advanced machine intelligence methods can reduce both cost and time significantly and increase accuracy in many solutions involving biomarker discovery that can rigorously match a patient's own data to clinical trials to achieve personalized cancer treatment [20].

Regarding successful applications of computational methods in medicine and life sciences, one of the most noticeable development is tensor computing or also known as tensor decompositions or tensor factorization [21, 22], which is a generalization of vectors and matrices and is considered as a multidimensional array. For example, a vector is a one-dimensional or first-order tensor and a matrix is a two-dimensional or second-order tensor.

Tensor decomposition-based features allow an effective and efficient way for machine learning to perform accurate pattern classification. Tensor-decomposition methods have been applied to the reconstruction of the pathways of a cellular system from genome-scale signals [23], integrative analysis of massive weighted biological networks [24], analysis of physiological signals [25, 26], and extraction of burn areas in color images [27]. Furthermore, tensor computing has been realized as a computational tool for achieving precision medicine in updating medical knowledge [28] and addressing heart failure [29]. In addition, methods of tensor decompositions have been suggested to be useful for machine learning [30, 31], and compressing the dense weight matrix of the fully-connected layers of deep neural networks to allow much faster machine training time [32].

Data science is an emerging multi-disciplinary research area that combines mathematics, statistics, and data-processing methods to represent and extract meaningful information from big data sources to be used for downstream analysis. Because of its multiple disciplines, AI and computation are complementary to data science. In fact, advanced data-science tools have been realized to have the potential to discover meaningful information from big data, which can be utilized to bring into innovation in clinical practice, ranging from personalized therapy, timely drug design, to population screening and electronic health records [33].

Advances in AI, computation, and data science together with enhanced computer hardware for rapid mathematical calculations and cloud computing, which can process big biomedical data efficiently and timely, are promising to bring benefits to three aspects of medicine: clinicians, health systems, and patients. For clinicians, accurate medical information for diagnosis, prognosis, and treatment can be obtained; for health systems, time and space enclosing clinical activities, environments, technologies, people, and organizations engaged in promoting and providing health care can be improved; and for patients, self-assessment of their own clinical data to promote own health can be carried out [13].

Given many advantages offered by AI, computation, and data science to medicine and life science, there are several technical and implementation challenges to overcome. For example, some of the current implementation issues include privacy protection, security, and transparency [13]. Regarding technical challenges, an issue in using AI methods such as deep learning is the problem of data imbalance, where the class distributions are highly imbalanced. Due to imbalanced data, classifiers tend to result in low predictive accuracy for the minority class. This is particularly true for applying AI to medicine [34, 35]. Medical datasets are often not balanced in the class labels because of limited samples collected from patients and the cost for acquiring annotated data. Current research focuses on using novel data transformation and statistical simulations to address this issue.

This edited monograph consists of 15 chapters addressing applications and development of AI, Computation, and Data Science in four areas: Bioinformatics, Medical Image Analysis, Physiology, and Innovation in Medicine and Health.

In Bioinformatics, the work entitled "Intelligent Learning and Verification of Biological Networks" by Helen Richards, Yunge Wang, Tong Si, Hao Zhang, and Haijun Gong addresses the combination of machine-learning and model-checking methods for verifying complex biological networks from omics data, which can provide insights into the pathogenesis of cancers. The chapter entitled "Differential Expression Analysis of RNA-Seq Data and Co-expression Networks" by Sana Javed describes mathematical methods for the analysis of RNA-seq data and coexpression networks with an application to studying lung cancer cell lines. Marta B. Lopes and Susana Vinga contribute a chapter entitled "Learning Biomedical Networks: Toward Data-Informed Clinical Decision and Therapy", which shows network science is a promising tool for contributing to the reality of precision medicine by implementing network analysis of clinical and molecular data for targeted therapies. Basel Abu-Jamous and Asoke K. Nandi presents simultaneous-clustering and deep-learning methods for analyzing multiple gene expression datasets in their contributed chapter entitled "Simultaneous Clustering of Multiple Gene Expression Datasets for Pattern Discovery", in which three cases using real high-throughput biological datasets were studied to illustrate the usefulness of the presented computational methods. Another chapter entitled "Artificial Intelligence for Drug Development" by Muhammad Waqar Ashraf gives a short review on the role of AI and machine learning in the field of chemistry for new drug development using big data for pre-clinical and post-marketing observation. Several interesting theoretical models for studying and monitoring spatial ecology and insect species distribution are presented in the chapter entitled "Mathematical Bases for 2D Insect Trap Counts Modelling" by Danish A. Ahmed, Joseph D. Bailey, Sergei V. Petrovskii, and Michael B. Bonsall, which is certainly of interest to ecologists, entomologists, and those performing field experiments.

In Medical Image Analysis, the chapter entitled "Artificial Intelligence in Dermatology: A Case Study for Facial Skin Diseases" contributed by Rola El-Saleh, Hazem Zein, Samer Chantaf, and Amine Nait-ali presents deep learning of images and computer software for identifying common human facial skin diseases. Another chapter entitled "Medical Imaging Based Diagnosis Through Machine Learning and Data Analysis" by Jianjia Zhang, Yan Wang, Chen Zu, Biting Yu, Lei Wang, and Luping Zhou discusses conventional and state-of-the-art machine learning approaches for medical image analysis, including brain tumor segmentation, mental disease classification, and modeling of brain networks. Vinayakumar Ravi, Harini Narasimhan, and Tuan D. Pham contribute a chapter entitled "EfficientNet-Based Convolutional Neural Networks for Tuberculosis Classification", which studied 26 pretrained convolutional neural networks for detecting tuberculosis (TB). The deep-learning networks were tested using a large public database of TB X-rays, and the results suggest EfficientNet models are promising for early diagnosis of TB. Another contribution entitled "AI in the Detection and Analysis of Colorectal Lesions Using Colonoscopy" by Zhe Guo, Xin Zhu, Daiki Nemoto, and Kazunori Togashi describes the use of colonoscopy for the screening, diagnosis, and therapy of colorectal cancer, and proposes an AI-based concept for assisting endoscopists in performing their task. Another work on the application of AI to medical imaging equipment is addressed in the chapter entitled "Deep Learning-Driven Models for Endoscopic Image Analysis" by Xiao Jia, Xiaohan Xing, Yixuan Yuan, and Max Q.-H. Meng.

In Physiology, the chapter entitled "A Dynamic Evaluation Mechanism of Human Upper Limb Muscle Forces" by Qing Tao, Zhaobo Li, Quanbao Lai, Shoudong Wang, Lili Liu, and Jinsheng Kang investigates the mechanisms of the human upper limb, where a Hill-type model was used for predicting muscle forces and can be useful for developing robotic upper limb rehabilitation systems. Another chapter entitled "Resting-State EEG Sex Classification Using Selected Brain Connectivity Representation" by Jean Li, Jeremiah D. Deng, Divya Adhia, and Dirk De Ridder presents classifier ensembles and feature analysis methods for sex difference classification using electroencephalography (EEG) signals.

Regarding Innovation in Medicine and Health, Giovanni Briganti contributes a chapter entitled "Augmented Medicine: Changing Clinical Practice with Artificial Intelligence", in which an innovation in medical technologies known as augmented medicine are addressed with applications to clinical practice, and discussed for potential future directions. Finally, the chapter entitled "Environmental Assessment Based on Health Information Using Artificial Intelligence" by Jannik Fleßner, Johannes Hurka, and Melina Frenken shows how novel communication, computational, and AI methods can contribute to healthcare by means of effective assessment of patient-specific environmental risk factors.

In summary, AI has become pervasive in many areas of research and applications. While computation can significantly reduce mental efforts for complex problem solving, effective computer algorithms allow continuous improvement of AI tools to handle complexity in both time and memory requirements for machine learning with large datasets. Data science is an evolving scientific discipline that strives for overcoming the hindrance of analytics with big data encountered by traditional skills that are too limited to enable the discovery of useful intelligence for leveraging research outcomes. Solutions to many problems in medicine and life sciences, which cannot be provided by conventional approaches, are urgently needed for the society. This edited book attempts to report recent advances in the complementary domains of AI, computation, and data science with applications in medicine, health, and life sciences. Benefits to the readers are manifold as researchers from similar or different fields can be aware of advanced developments and novel applications that can be useful for either immediate implementations or future scientific pursuit.

Khobar, Saudi Arabia Tuan D. Pham
Kowloon, Hong Kong Hong Yan
Khobar, Saudi Arabia Muhammad W. Ashraf
Linköping, Sweden Folke Sjöberg

References

1. Freiherr G (2020) How artificial intelligence might personalize healthcare, Siemens Healthineers, https://www.siemens-healthineers.com/news/mso-column-ai-personalizing-healthcare.html
2. Miotto R, Li L, Kidd BA, Dudley JT (2016) Deep patient: an unsupervised representation to predict the future of patients from the electronic health records. Sci Rep 6:26094
3. Anumanchipalli GK, Chartier J, Chang EF (2019) Speech synthesis from neural decoding of spoken sentences. Nat 568:493–498
4. Frohlich H, et al. (2018) From hype to reality: data science enabling personalized medicine. BMC Med 16:150
5. Bibault JE, Giraud P, Burgun A (2016) Big data and machine learning in radiation oncology: State of the art and future prospects. Cancer Lett 382:110–117
6. Shrager J, Tenenbaum JM (2014) Rapid learning for precision oncology. Nat Rev Clin Oncol 11:109–118
7. Ding MQ, et al. (2018) Precision oncology beyond targeted therapy: Combining omics data with machine learning matches the majority of cancer cells to effective therapeutics. Mol Cancer Res 16:269–278
8. Ahmed Z, Mohamed K, Zeeshan S, Dong X (2020) Artificial intelligence with multi-functional machine learning platform development for better healthcare and precision medicine. Database 2020: baaa010
9. Zhou XY, Guo Y, Shen M, Yang GZ (2020) Application of artificial intelligence in surgery. Front Med 14:417–430
10. AI Expert Newsletter (2013) W is for Winter, Wayback Machine, archived 9 November 2013
11. Bengio Y, Courville A, Vincent P (2013) Representation learning: A review and new perspectives. IEEE Trans Pattern Anal Mach Int 35:1798–1828
12. Schmidhuber J (2015) Deep Learning in Neural Networks: An overview. Neural Netw 61:85–117
13. Topol EJ (2019) High-performance medicine: The convergence of human and artificial intelligence. Nat Med 25:44–56
14. Webb S (2018) Deep learning for biology. Nat 554:555–557
15. Science News Staff (2017) AI is changing how we do science. Get a glimpse. Science, https://www.sciencemag.org/news/2017/07/ai-changing-how-we-do-science-get-glimpse
16. Savage N (2020) How AI is improving cancer diagnostics. Nat 579:S14–S16
17. Huang S, Yang J, Fong S, Zhao Q (2020) Artificial intelligence in cancer diagnosis and prognosis: Opportunities and challenges. Cancer Lett 471:61–71
18. Mobadersany P, et al. (2018) Predicting cancer outcomes from histology and genomics using convolutional networks. Proc Nat Acad Sci 115:E2970–E2979
19. Pham TD, Fan C, Zhang H, Sun XF (2020), Artificial intelligence-based 5-year survival prediction and prognosis of DNp73 expression in rectal cancer patients. Clin Transl Med 10:e159
20. Ho D (2020) Artificial intelligence in cancer therapy. Sci 367:982–983

21 Kolda TG, Bader BW (2009) Tensor decompositions and applications. SIAM Rev 51:455–500

22. A. Cichocki, et al. (2015) Tensor decompositions for signal processing applications: From two-way to multiway component analysis. IEEE Signal Process Mag 32:145–163
23. Alter O, Golub GH (2005) Reconstructing the pathways of a cellular system from genome-scale signals by using matrix and tensor computations. Proc Natl Acad Sci USA 102:17559–17564
24. Li W, Liu CC, Zhang T, et al. (2011) Integrative analysis of many weighted co-expression networks using tensor computation. PLoS Comput Biol 7:e1001106
25. Cong F, et al. (2015) Tensor decomposition of EEG signals: A brief review. J Neurosci Methods 248:59–69

26. T.D. Pham, H. Yan (2018) Tensor decomposition of gait dynamics in Parkinson's disease. IEEE Trans Biomed Eng 65:1820–1827
27. Cirillo MD, Mirdell R, Sjoberg F, Pham TD (2019) Tensor decomposition for color image segmentation of burn wounds. Scientific Reports 9:329
28. Luo Y, Wang F, Szolovits P (2017) Tensor factorization toward precision medicine. Brief Bioinform 18:511–514
29. Luo Y, Ahmad FS, Shah SJ (2017) Tensor factorization for precision medicine in heart failure with preserved ejection fraction. J Cardiovasc Transl Res 10:305–312
30. Sidiropoulos ND, et al. (2017) Tensor decomposition for signal processing and machine learning. IEEE Trans Signal Process 65:3551–3582
31. Ji Y, Wang Q, Li X, Liu J (2019) A Survey on tensor techniques and applications in machine learning. IEEE Access 7:162950–162990
32. Novikov A, Podoprikhin D, Osokin A, and Vetrov D (2015) Tensorizing neural networks. Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS'15, Montreal, Canada) 1:442–450
33. Tim H, et al. (2019) From big data to precision medicine. Front Med 6:34
34. Larrazabal AJ, et al. (2020) Gender imbalance in medical imaging datasets produces biased classifiers for computer aided diagnosis, Proc Natl Acad Sci USA 117:12592–12594
35. Naseriparsa M, et al. (2020) RSMOTE: Improving classification performance over imbalanced medical datasets. Health Inf Sci Syst 8:22

The original version of the book was revised: Book volume no "31" has been inserted in the book. The correction to the book is available at https://doi.org/10.1007/978-3-030-69951-2_16

Contents

Part I Bioinformatics

Part II Medical Image Analysis

Part III Physiology

Part IV Innovation in Medicine and Health

Part I
Bioinformatics

Chapter 1
Intelligent Learning and Verification of Biological Networks

Helen Richards, Yunge Wang, Tong Si, Hao Zhang, and Haijun Gong

Abstract Machine learning and model checking are two types of intelligent computing techniques that have been widely used to study different complicated systems nowadays. It is well-known that the cellular functions and biological processes are strictly regulated by different biological networks, for example, signaling pathways and gene regulatory networks. The pathogenesis of cancers is associated with the dysfunctions of some regulatory networks or signaling pathways. A comprehensive understanding of the biological networks could identify cellular signatures and uncover hidden pathological mechanisms, and help develop targeted therapies for cancers and other diseases. In order to correctly reconstruct biological networks, statisticians and computer scientists have been motivated to develop many intelligent methods, but it is still a challenging task due to the complexity of the biological system and the curse of dimensionality of the high-dimensional biological data. In this work, we will review different machine learning algorithms and formal verification (model checking) techniques that have been proposed and applied in our previous work and discuss how to integrate these computational methods together to intelligently infer and verify complex biological networks from biological data. The advantages and disadvantages of these methods are also discussed in this work.

1.1 Introduction

Nowadays, targeted therapy has become an important and effective treatment method for many types of cancers, which uses specific chemical compounds to target some mutated genes and proteins implicated in tumorigenesis. To develop targeted therapies, we need to identify important genetic mutations and understand how these

H. Richards
Biomedical Engineering Department, Saint Louis University, St. Louis, MO, USA

Y. Wang · T. Si · H. Zhang · H. Gong (✉)
Department of Mathematics and Statistics, Saint Louis University,
220 N Grand Blvd, St. Louis, MO, USA
e-mail: haijun.gong@slu.edu

T. D. Pham et al. (eds.), *Advances in Artificial Intelligence, Computation, and Data Science*, Computational Biology 31,
https://doi.org/10.1007/978-3-030-69951-2_1

mutated genes and proteins influence the gene regulatory networks and signaling pathways that are involved in the cancer cell growth. So, understanding of the biological network is one of the key steps to investigate the cellular system and pathogenesis of different diseases. Reconstruction of biological networks involves two major steps: inference and verification, that is, how to intelligently and correctly infer optimal networks, and how to intelligently and automatically verify or falsify the inferred networks. Statisticians and computer scientists have proposed different statistical learning and model checking techniques to implement network inference and verification in the past years.

Different learning methods have been developed to infer the biological networks from high-dimensional omics data generated by the modern genomic technologies. The traditional deterministic methods are based on the ordinary differential equations [3], which is not convenient to handle the noisy high-dimensional data. Most network learning methods are based on the probabilistic graphical models (PGM) [16, 17, 31], for example, the Bayesian network and dynamic Bayesian networks methods. In the PGM, each node represents a variable which could be a gene or protein, and the edge connecting two nodes indicates a possible causal or conditional dependency relationship (which could be activation or inhibition or association). The simplest PGM is a Bayesian network (BN) model, but the BN model can not handle the positive or negative feedback loops which are important processes in the genetic networks and signaling pathways. In most omics datasets, the number of observations or measurements is significantly fewer than the number of genes or proteins, some LASSO-based methods, for example, graphical LASSO [15, 36], have been developed to estimate inverse covariance matrix, which can infer some undirected networks.

To reconstruct a network which contains causality information and allows the feedback loops, different Dynamic Bayesian network (DBN) [17, 31, 32, 37]-based models, which assume a first-order Markov chain, have been proposed to learn directed networks. We are more interested in the regulatory networks which contain more information than the correlation and causality graphs. Most network learning methods can not identify the “activation” and “inhibition” relationship between different nodes in the regulatory networks. Our previous work [22, 35] introduced a signed integer weight by modifying the influence score proposed in [49] to identify the activation/inhibition relationship for each edge on the regulatory networks.

Several biomedical studies [4, 11, 29, 34] found that, in some biological systems, the network structure is time-varying at different stages. For example, the network structure of naive/effector T cells is different from the senescent T cells [11] due to some mutations during tumorigenesis. Recently, time-varying network inference methods were proposed [1, 18, 27, 40, 44], including dynamic vector autoregressive model [18], heterogeneous DBN model [27], L1-regularized logistic regression model [1], and dynamic linear model [44]. However, these methods can only be used to learn time-varying causality networks or undirected graphs, instead of regulatory networks.

There are several steps that are challenging in the reconstruction of time-varying regulatory network than the stationary network inference. The first step before the network structure learning is to identify when the network structure starts to make changes. This is a change-points detection problem in statistics. The change-point detection is an open problem in high-dimensional time-series statistical data analysis, it is especially difficult if both the number and locations of change-points are unknown. The second challenge is how to learn the optimal time-varying networks at different stages. Our recent work [43] indicates that if we could identify the change-points, we still can use the dynamic Bayesian network model to learn the optimal networks at different stages. Another challenge is how to intelligently and automatically check whether or not the inferred networks are consistent with existing experiments or known databases. It is not realistic to manually verify complex networks which are composed of thousands of genes, proteins, and interactions. Our previous work [19, 20, 22, 24, 25] had introduced and applied different model checking techniques, including statistical model checking, symbolic model checking and probabilistic model checking, to formally analyze the signaling pathway models and gene regulatory networks by checking some desired temporal logic formulas abstracted from the experiment and known database.

Our recent work [43] proposed a preliminary integrative approach to reconstruct time-varying regulatory networks. The reconstruction of regulatory networks from time-series high-dimensional data involves several key steps: identification of change-points, network structure learning and searching at different stages, learn the activation/inhibition relationship for each edge, and verification of inferred networks. In this chapter, we will review several machine learning algorithms and model checking techniques that we have proposed and applied in our previous biological network studies [19, 20, 22, 24, 25, 35, 43] and discuss their limitations in the network inference and verification, and propose a way to integrate these techniques together for the regulatory network reconstruction.

1.2 Statistical Learning of Regulatory Networks

During the pathogenesis of some diseases or stimulation under some conditions, some genetic or protein mutations might change the structure of genetic networks or sequence of signal transduction. For example, the mutations of tumor-suppressor proteins P53 and Rb, and oncoproteins RAS and MDM2 could change the signal transduction in the P53-MDM2, Rb-E2F, and RAS networks. Figure 1.1 illustrates simple regulatory networks, which are composed of three-nodes, with time-varying network structures. These structure changes include some newborn edges connecting two nodes, removal of some old edges, changes of activation/inhibition relationship. To reconstruct a regulatory network, we should first check the network structure is time-invariant or time-varying over time from the high-dimensional time-series data, which is a change-points detection problem in statistics.

To identify the change-points, which describe when the network structure starts to make changes, from the high-dimensional time-series data is a difficult problem in statistics. This is especially challenging if both the number and locations

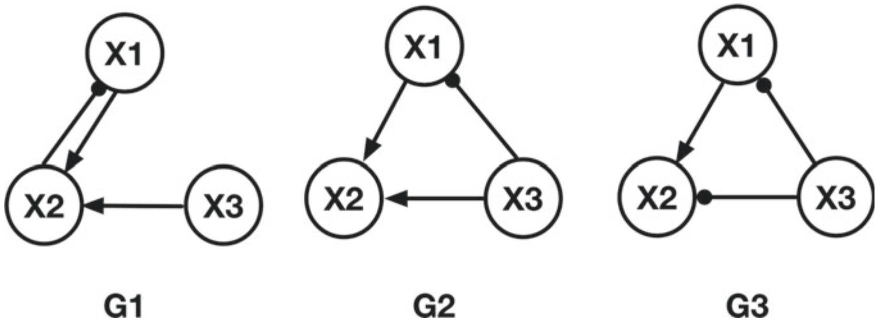


Fig. 1.1 Illustration of time-varying regulatory networks at three different stages. The arrow represents activation and the filled ball arrow represents inhibition relationship

of change-points are jointly unknown. Some unsupervised and supervised learning-based change-points detection methods [13, 14, 26, 30, 39, 41] have been proposed, including the density-ratio test [30], Bayesian inference approaches [13, 41], product partition model [5], reversible jump Markov Chain Monte Carlo sampling algorithm [26], logistic regression [14], and hidden Markov model [39]. If the data is low-dimensional, most of these methods work well. To handle the high-dimensional omics data, most existing methods have their limitations.

1.2.1 INSPECT Change-Points Identification

To choose a change-points identification method that can handle the high-dimensional omics data is the first important step in the network reconstruction, it is challenging if both the number and locations of change-points are unknown. In our recent work [43], we adopted a change-point estimation method called INSPECT proposed by [42] and applied it to identify the change-points from the high-dimensional microarray data. Our studies found this method could be used to identify multiple change-points from high-dimensional time-series data if we can choose some correct values for a small number of parameters.

The INSPECT (informative sparse projection for estimation of change-points) algorithm is based on the ADMM (alternating direction method of multipliers) algorithm to perform sparse singular value decomposition on the CUSUM (cumulative sum) transformation matrix [10, 42] and find sparse leading left singular vectors for the change-points identification. The details are given in [42], for completeness, we summarize some key points of this algorithm in this chapter. Given some high-dimensional time-series microarray data described by a matrix $X = (X_1, \ldots, X_n)^T \in R^{n \times p}$, which consists of p genes measured at n different time points, in which, X_t $(1 \leq t \leq n)$ describes the expression level of all the p genes observed at time t. X_t follows some unknown distribution with a time-changing mean vector μ_t at different stages. Assuming there are ν change-points denoted by $\{c_1, \dots, c_\nu\}$ in the data X, where $1 \le c_1 < c_2 < \cdots < c_\nu \le n-1$, then, there are $\nu + 1$ stages described by $\{s_1, \dots, s_{\nu+1}\}$, where $s_i = (c_i, c_{i+1})$. At any stage, the network structure is assumed to be time-invariant, then, the mean vectors $\mu^{(i)}$ is a constant at its stage s_i.

Algorithm 1: Change-points Estimation INSPECT Pseudocode [42]

```
Input: High-dimensional time-series data D; Parameters (regularization, threshold)
1. Perform CUSUM transformation T, where
   $[T_{n,p}(X)]_{t,i} = \sqrt{\frac{n}{t(n-t)}}\left(\frac{t}{n}\sum_{j=1}^{n} X_{j,i} - \sum_{j=1}^{t} X_{j,i}\right)$.
2. Find $k$-sparse leading left singular vectors $\hat{v}_k$ of CUSUM using ADMM algorithm: solve
   convex optimization problem: $\hat{v}_k = argmax||T^T v||_2$, with the constraint
   $||\mu^{(i)} - \mu^{(i-1)}||_0 \le k$, where $i = 1, 2, \dots, \nu$, and $k \in \{1, \dots, p\}$.
3. Locate change-points by wild Binary segmentation.
Output: Number and locations of change-points
```

Algorithm 1 summarizes the procedure of INSPECT method [42] to estimate the number and locations of multiple change-points from high-dimensional time-series data. The first step is to perform the CUSUM (cumulative sum) transformation [10, 42] matrix $T_{n,p} : R^{n \times p} \to R^{(n-1)\times p}$, which is defined in the Algorithm 1, to obtain the optimal projection direction that are closely aligned with the changes of mean vectors between the stages s_i and s_{i+1}. The second step is to find the k-sparse leading left singular vectors $\hat{v}_k$ [42, 46] of the CUSUM transformation matrix T, which is equivalent to solving a convex optimization problem. Most parts of the network structure should be conserved, so, there should not be too many edge-changes during stage transition. A sparsity is introduced using the constraint $||\mu^{(i)} - \mu^{(i-1)}||_0 \le k$ to constrain the number of edge-changes. Finally, locate change-points by wild Binary segmentation. More details with proof are in Wang et al.'s work [42].

This work only discussed the INSPECT change-points detection algorithm that we have applied in our previous work, the interested readers could also refer to [5, 13, 14, 26, 39, 41] for different change-points detection methods for comparison.

1.2.2 Network Structure Learning and Searching

Given the time-series dataset D which is expressed by an $n \times p$ matrix showing the expression levels of p genes measured at n different time points. The change-points detection algorithm is implemented first, if there is no change-point, then, the network can be modeled by a stationary dynamic Bayesian network (DBN); if there are at least one change-points, then, the network structure is time-varying, we should infer different DBN models at different stages, that is, time-varying DBNs.

1.2.2.1 Time-Invariant DBN Learning

Dynamic Bayesian Network (DBN) has been reviewed in many previous studies. We will briefly review this method. DBN model is a pair (G, Θ), where $G = (V, E)$ represents a directed graph, in which V represents a set of random variables or nodes, E is a set of edges, and $\Theta = P(X|Par(X))$ is a set of conditional probability distributions of the nodes $X \in V$ given its parental nodes denoted by $Par(X)$. Figure 1.2 illustrates how to use a DBN to model a simple three-nodes regulatory network (left) given in Fig. 1.1. In the time-invariant DBN model, the network structure is assumed stationary, and the state of each node measured at time $t + 1$ is dependent on the states of its parental nodes and itself measured at time t only. For example in Fig. 1.2, the node X_2's value at time t_2 is dependent on its parental nodes X_1 and X_3's values and itself at t_1.

The directed time-invariant DBN can be encoded by a joint distribution [31] over all the random variables $V = (\mathbf{X_1}, \mathbf{X_2}, \ldots, \mathbf{X_n})$:

$$\begin{aligned} P(V) = P(\mathbf{X_1}, \mathbf{X_2}, \ldots, \mathbf{X_n}) &= \textstyle\prod_{\mathbf{X} \in V} P(\mathbf{X}|Par(\mathbf{X})) \\ &= P(\mathbf{X_1})P(\mathbf{X_2}|\mathbf{X_1}) \times \ldots \times P(\mathbf{X_n}|\mathbf{X_{n-1}}), \end{aligned}$$

where, $P(\mathbf{X_t}|\mathbf{X_{t-1}}) = P(X_{t1}|Par(X_{t1})) \times \cdots \times P(X_{tp}|Par(X_{tp}))$, $Par(X_{tj})$ represents the gene j's parents' level measured at time $t - 1$.

Given the data $\mathbf{D}$, a scoring function is needed to evaluate the goodness of the network. Different scoring metrics, including the Bayesian Dirichlet equivalence (BDe) metric [28], BIC/AIC [2], Chow-Liu tree learning algorithm [8], have been applied to learn the structure of the dynamic Bayesian network. Bayesian Dirichlet equivalence (BDe) metric [28] is one of the most widely used methods to learn the network structures. Learning an optimal directed network (G^*, Θ) is equivalent to

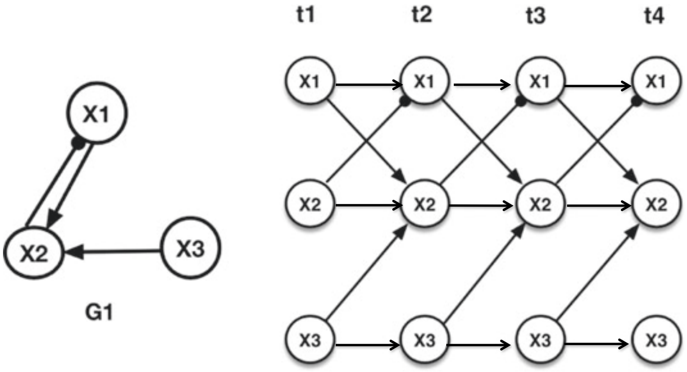


Fig. 1.2 Illustration of a three-nodes regulatory network (left) and a stationary dynamic Bayesian network model (right) whose structure is invariant with time. The state of each node measured at time $t + 1$ is dependent on the states of its parental nodes and itself measured at time t only

maximizing the posterior distribution of the network G, that is to solve the following optimization problem:

$$\begin{aligned} G^* = \text{argmax}_G\, P(G|\mathbf{D}) = \text{argmax}_G\, P(G,\mathbf{D}) = \text{argmax}_G\, P(\mathbf{D}|G)P(G), \\ P(\mathbf{D}|G) = \int P(\mathbf{D}|G,\Theta)P(\Theta|G)d\Theta, \\ \mathbf{D}|\Theta \sim Multinomial(\Theta), \\ \Theta|G \sim Dirichlet(\alpha). \end{aligned}$$

where, $P(G)$ is the prior of the network G, which can be chosen in different ways, e.g, the minimal description length (MDL) was used in [17]. $P(\mathbf{D}|G)$ is the likelihood function if the parameter vector Θ is continuous. The structure parameters Θ's prior follows Dirichlet distribution with a hyperparameter vector α given a network G; while the $\mathbf{D}$ is a multinomial sample dependent on the parameters Θ; and Θ are assumed to be globally and locally independent. Later, some heuristic searching algorithms (e.g., greedy searching and simulated annealing which was used in the Banjo) are applied to find the optimal networks. Algorithm 2 summarizes the procedure to infer an optimal stationary dynamic Bayesian network model. It is noteworthy to mention that, the BDe metric has assumed the following two assumptions are valid: two directed acyclic networks G_1, G_2 are equivalent if they encode the same joint probability distribution; and, if the network G_1 is equivalent to G_2, the distribution function of Θ will be same in both networks.

Algorithm 2: BDe-based Stationary Network Structure Learning

```
  Input: High-dimensional time-series data D
  for each network G do
      Optimal network structure learning;
          $\mathbf{D}|\Theta \sim Multinomial(\Theta)$;
          Dirichlet prior distribution for $\Theta|G$;
          Estimate Bayesian Dirichlet equivalence (BDe) metric;
  end
  Optimal network searching:
    Sort networks according to BDe scores;
    Search optimal network by simulated annealing;
Output: Optimal regulatory networks
```

Another popular method is to maximize the BIC score which is written as $BIC(G) = logP(G|\mathbf{D}) - f(N)|G|$, where $f(N)$ is a non-negative penalization function, and $|G|$ denotes the network complexity. Thus, finding an optimal network requires trading off fit and complexity of network. It is also called minimal description length (MDL) score. If $f(N) = 1$, we get the Akaike Information

Criterion (AIC) scoring function [2]. Many network structure learning algorithms have been proposed in the past twenty years, the interested readers could refer to [2] and other references for details.

1.2.2.2 Time-Varying DBN Learning

Figure 1.3 illustrates time-varying regulatory networks and how to use the dynamic Bayesian networks to model these regulatory networks with different structures at different stages. The regulatory network experiences structure changes from stage G1 to G2 to G3, including the removal of edges (X1 activates X2), birth of new edges (X3 inhibits X1), and change of activation/inhibition relationship (X3 inhibits X2) at three different stages. If there is at least one change-point, the first-order Markov property will not be valid in the time-varying DBN because of the new network structure. Our recent work [43] proposed that, in the time-varying DBN model, the first-order Markov property should be updated as: the state of any node measured at time $t + 1$ is dependent on the states of itself and its parental nodes measured

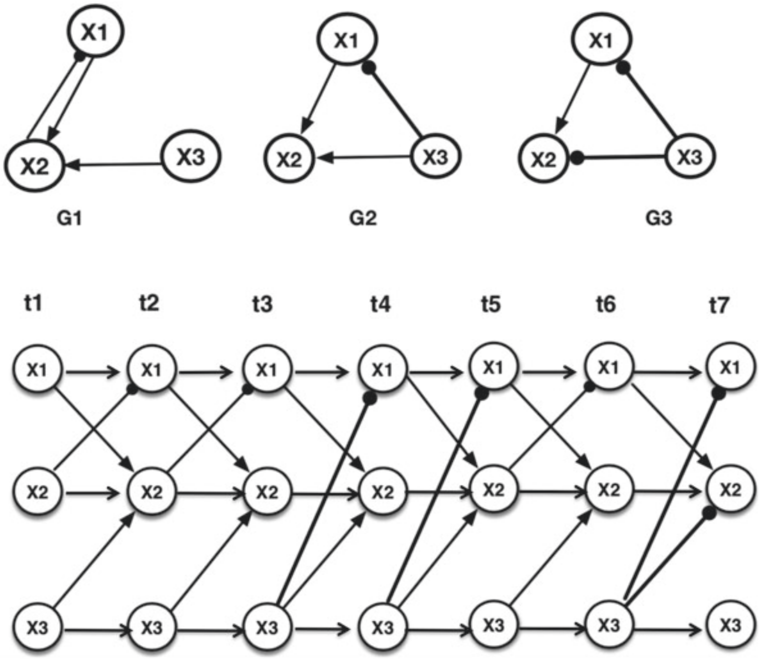


Fig. 1.3 Illustration of time-varying regulatory network (up), and time-varying dynamic Bayesian network model (bottom). The regulatory network experiences structure changes from stage G1 to G2 to G3, including the removal of edges (X1 activates X2), birth of new edges (X3 inhibits X1), and change of activation/inhibition relationship (X3 inhibits X2) at three different stages

at time t and also its current network structure or stage. So the structure parameter Θ which is described by the conditional probability distribution can be written as $\Theta = P(X|Par(X), S)$, where $X \in V$, and S represents the stage.

In the time-varying DBN learning, there will be different optimal network structures denoted by $G^* = (G_1^*, \dots, G_{\nu+1}^*)$ at different stages $S = (S_1, \dots, S_{\nu+1})$ if there exist ν change-points $(c_1, \dots, c_\nu)$. To learn the optimal time-varying network structures is to maximize the joint posterior distribution of all the network structures which are stage dependent, the optimization problem is express as

$$\begin{aligned}(G_1, \dots, G_{\nu+1})^* &= \text{argmax}_{(G_1, G_2, \dots, G_{\nu+1})} P((G_1, \dots, G_{\nu+1})|D, S) \\ &= \text{argmax}_{G=(G_1, G_2, \dots, G_{\nu+1})} P(D|S)P(G|S)\end{aligned}$$

To directly solve the above optimization problem is very difficult, in our recent work [43], we first applied the INSPECT algorithm to identify the number and locations of change-points. That is, we know how many stages and when the structure starts to make changes. Then, we used the stationary DBN model to learn the network structures at different stages individually. That is, we assume these network structures are independent at different stages, so the above optimization problem can be simplified as

$$\begin{aligned}(G_1, \dots, G_{\nu+1})^* &= \text{argmax}_{(G_1, G_2, \dots, G_{\nu+1})} P((G_1, \dots, G_{\nu+1})|(D_1, \dots, D_{\nu+1}), (S_1, \dots, S_{\nu+1})) \\ &= \text{argmax}_{G=(G_1, G_2, \dots, G_{\nu+1})} P(G_1|D_1, S_1) \dots P(G_{\nu+1}|D_{\nu+1}, S_{\nu+1})\end{aligned}$$

where, D_i represents the observation data at stage S_i. Algorithm 3 summarizes the pseudocode of the time-varying network structure learning.

Algorithm 3: Time-varying Network Structure Learning

```
Input: High-dimensional time-series data $D$;
1st, Identify Number and locations of change-points;
2nd, Divide the data $D$ according to stages;
for data $D_t$ of the stage $S_t$ do
    for each network $G_t$ do
        Stationary DBN network structure learning at stage $S_t$;
    end
    Optimal network searching;
end
Output: Optimal time-varying regulatory networks
```

Most existing network reconstruction methods based on the dynamic Bayesian network could only infer a causality or correlation graph, not a regulatory network. The regulatory network should contain the activation and inhibition information to describe the regulatory relationship. Next, we will discuss how to identify the regulatory relationship on the inferred stationary and time-varying networks.

1.2.3 Regulatory Relationship Identification

Figures 1.2 and 1.3 illustrate time-invariant and time-varying regulatory networks. On the graph, an arrow is used to represent the activation event, while the filled ball arrow is used to represent the inhibition event. In our previous work [35], we introduced and estimated the signed integer weights, which modified the influence score proposed in [49], to identify the activation and inhibition relationship and interaction strength. If it is a time-invariant regulatory network, the identified activation/inhibition relationship on each edge will not change over time. A positive weight corresponds to an activation event, while a negative value corresponds to an inhibition event between two nodes. The Bayesian network inference with Java objects [49] estimated the influence score according to the cumulative distribution function which is written as

$$G_{ijk}(t) = \sum_{m=0}^{k} \omega_{ijm}(t) = \sum_{m=0}^{k} P(X_{ti} = m | Par(X_{ti}) = j). \tag{1.1}$$

$G_{ijk}(t)$ calculates the probability that gene X_{ti}'s level is no more than k given its parent gene takes a value of j. For the gene X_{ti} at time t, if ω_{ijm} is an increasing function, X_{ti}'s expression level has a high chance to be upgraded given that its parent's level increases. Then, the interaction between X_{ti} and its parent will be voted as an activation event by a predefined voting machine [49] based on the values of $G_{ijk}(t)$; else, it will be voted as an inhibition event. Our previous work [35] converted the influence score to be a signed integer weight, which can not only describe the regulatory relationship and interaction strength, but also update the state transfer function in the symbolic model checking.

In the time-varying networks, the structure changes could influence the sign and magnitude of interaction strength. Our recent work [43] extended the Eq. 1.1 to allow the integer weights to change at different stages in response to the network structure changes. Now the quantity $G_{ijk}(t, S_t)$ in Eq. 1.2 measures the probability that, at time t in the stage S_t, a node X_i will take a value $m \in \{0, 1, \ldots, k\}$ given that its parent nodes $Par(X_i)$ taking a value of j at stage S_t. If ω_{ijm} is an increasing (decreasing) function of its parent node's value, then, this interaction will be voted as an activation (inhibition) event at a specific stage S_t based on the values of $G_{ijk}(t, S_t)$. Finally, the influence score will be converted into a signed integer weights at different stages.

$$G_{ijk}(t, S_t) = \sum_{m=0}^{k} \omega_{ijm}(t, S_t) = \sum_{m=0}^{k} P(X_{ti} = m | Par(X_{ti}) = j, S_t). \tag{1.2}$$

The estimation of stage-dependent integer weight using the Eq. 1.2 is not easy compared with Eq. 1.1 if the change-points or stages are unknown. To simplify the problem, if we can identify the change-points in the first step using the change-points detection algorithm, we can estimate the integer weights of the networks at different stages individually using the Eq. 1.1. This simplification is based on the assumption that the network structures are independent at different stages. Algorithm 4 summarizes the procedure for the time-varying regulatory network structure learning and integer weight estimation.

Algorithm 4: Signed Integer Weight Estimation in Time-varying Networks

```
Input: High-dimensional time-series data D;
1st, Identify Number and locations of change-points;
2nd, Divide the data $D$ according to stages;
for data $D_t$ of the stage $S_t$ do
    for each network $G_t$ do
        Optimal stationary DBN network structure learning;
        Signed integer weight estimation;
        for each edge $(X_i, Par(X_i))$ do
            Compute $\omega_{ijm}(t, S_t) = P(X_{ti} = m | Par(X_{ti}) = j, S_t)$;
            Compute $G_{ijk}(t, S_t) = \sum_{m=0}^{k} \omega_{ijm}(t, S_t)$;
            Calculate Influence Score based on $G_{ijk}(t, S_t)$;
            Convert the influence score into signed integer weights;
        end
        Optimal network searching;
    end
end
Output 2: Optimal time-varying regulatory networks;
          Signed integer weights at different stages.
```

Gene regulatory networks or signaling pathways models are normally complex and composed of thousands of genes or hundreds of proteins and interactions, but with a small number of observations, especially in the time-series data. All the network inference and change-points detection algorithms are dependent on some parameters. The inferred “optimal” network could be different if we change some parameters’ values. After the networks are inferred, how to validate these “optimal” networks? Without validation, we can not trust the simulation results or predictions made by the inferred networks. Next section, we will introduce two major formal verification methods, symbolic model checker SMV and probabilistic model checker PRISM that have been applied in our previous studies, and discuss how to use them to formally verify the inferred networks.

1.3 Formal Analysis of Regulatory Networks

Model Checking [9] is a formal verification technique for the finite state systems modeled by a Kripke Structure, which is depicted by $M = (S, s_0, R, L)$, where S is a finite set of states with the initial state $s_0 \in S$, R is a transition relation between states, and L is a labeling function. During model checking, the model M will first be converted into a state transition system, then, model checker will automatically and exhaustively search the state transition system to verify or falsify the desired property, which is expressed as a temporal logic formula ψ, starting from the initial state s_0. Model Checking problem [9] is expressed as $\{s \in S | M, s \models \psi\}$, which means, the model M satisfies ψ. Different model checking tools, for example, BLAST, Prism, SPIN, NuSMV, et al., have been developed to verify the design of hardware and software systems. The model checking technique was also introduced to study the cyber-physical system [7, 38] and biological systems [19, 20, 23]. Our previous studies [19, 20, 22, 24, 25] have introduced and applied different model checking techniques, including statistical model checker, symbolic model checker SMV, and probabilistic model checker PRISM, to formally analyze signaling pathway models and gene regulatory networks. This technique has been very successful in hardware systems verification, according to our studies, it is very promising that it could be a powerful tool in biological network verification.

Though we had discussed the temporal logic formulas in our previous studies [19, 20, 22, 24, 25], we will revisit some key formulas and semantics in this section again for completeness.

1.3.1 Temporal Logic Formula

Temporal logic formulas ψ can be divided into two subtypes, Linear Temporal Logic (LTL) and Computation Tree Logic (CTL). The LTL or CTL formula ψ is composed of an atomic proposition AP, Boolean variables, Boolean connectives which include $\vee$ (or), $\wedge$ (and), $\neg$ (not), and $\rightarrow$ (implication), but in SMV or PRISM code, we use the following symbols | (or), & (and), ! (not) to encode the Boolean connectives. In the LTL formula, we also need the temporal operators **X, F, G, U** to describe some property on a path, in which, $\mathbf{X}p$ means p holds in the next state of the path; $\mathbf{F}p$—p holds at some state in the Future (eventually) on the path; $\mathbf{G}p$—p holds Globally (always) at every state on the path; $p\mathbf{U}q$—p holds Until q holds on the path. A CTL formula describes properties of computation trees [9], which have branches corresponding to all possible paths from the root (that is, initial state of the system). In the CTL formula, the path quantifier **A, E** must precede the LTL operators **X, F, G, U**, so there are eight CTL operators: AX, EX, AG, EG, AF, EF, AU, EU that are frequently used in our studies.

Given the state formulas ψ and path formulas ϕ, the syntax of the CTL logic is expressed as [9]

$$\psi ::= AP \mid \psi_1 \vee \psi_2 \mid \neg\psi \mid \mathbf{E}\phi \mid \mathbf{A}\phi$$
$$\phi ::= \mathbf{X}\psi \mid \mathbf{F}\psi \mid \mathbf{G}\psi \mid \psi_1\mathbf{U}\psi_2.$$

The path $\pi = s_0, s_1, \ldots$ represents an infinite sequence of states, where $s_i \in S$, and for every $i \geq 0$, $(s_i, s_{i+1}) \in R$ represents a transition of the system, and π^i denotes the suffix of π starting at s_i. The semantics of a CTL formula is summarized below [9]

$$\begin{array}{ll}
M, s \models p & \text{iff } \ p \in L(s); \\
M, s \models \neg\psi & \text{iff } \ M, s \models \psi \text{ does not hold;} \\
M, s \models \psi_1 \vee \psi_2 & \text{iff } \ M, s \models \psi_1 \text{ or } M, s \models \psi_2; \\
M, \pi \models \mathbf{X}\psi & \text{iff } \ M, \pi^1 \models \psi; \\
M, \pi \models \psi_1\mathbf{U}\psi_2 & \text{iff } \ \text{there exists } k \geq 0 \text{ such that, } M, \pi^k \models \psi_2 \\
 & \text{and for all } 0 \leq j < k, \ \ M, \pi^j \models \psi_1; \\
M, s \models \mathbf{E}\phi & \text{iff } \ \text{there exists a path } \pi \text{ from } s \text{such that } M, \pi \models \phi; \\
M, s \models \mathbf{A}\phi & \text{iff for every path } \pi \text{ from } s, \ \ M, \pi \models \phi.
\end{array}$$

The readers could refer to [9] for more details about the LTL and CTL semantics and formulas.

1.3.2 Symbolic Model Checking

The symbolic model verifier (SMV) encoded by the ordered binary decision diagram (OBDD) [6] is a powerful model checking technique that has been successfully used for the verification of CPU or digital circuits designs. Our previous work have applied SMV to verify different regulatory networks [35, 43] and signaling pathway models [20, 23]. Algorithm 5 describes the pseudocode of SMV program given a network and some experimental observations. Each program starts with a "MODULE MAIN", and all the variables (describing the genes/proteins) are defined by "VAR" and initialized by "init" under the keyword "ASSIGN". The state transition for any node for the next state is updated by a state transfer function which is dependent on the signed integer weights estimated by the Algorithm 4, the value of the current state and corresponding parental nodes' values.

After the network is encoded, finally, the keyword "SPEC" will be used to encode some desired CTL formulas which are abstracted from the experimental observations. SMV can exhaustively search the state transition system M and check whether $M \models \psi$ is true or not. If the model M satisfies the property ψ, SMV will output "True", else, "False" with a counter-example will be given.

The LTL and CTL formula discussed in the Sects. 1.3.1–1.3.2 can only describe properties of an infinite sequence of states. That is, there is no time bound. However, if M describes a stochastic model, in most simulation studies, we have to stop the simulation to check some properties, so there is a time bound. For example, we can use the software BioNetGen [25] to model the stochastic biochemical reactions, which

Algorithm 5: Symbolic Model Verification of Regulatory Networks

```
Part 1: SMV Code Structures
Input 1: Inferred networks $M$;
         Signed integer weights;
         Temporal logic formula $\psi$
for each network, do
    MODULE MAIN: Starter of SMV code;
    VAR: Declare variables;
    ASSIGN & init: Initialize variables and assign values;
    next: Update state by the transfer functions;
end
Output 1: SMV Code for each network
Part 2: SMV Verification of CTL Formula
Input 2  : SMV Code from Part 1;
         Observations/Experimental results
  1. Design desired CTL formula $\psi$ from experiment;
  2. SPEC: Specify the CTL formula $\psi$;
  3. Attach CTL formula to the end of SMV code;
  4. Run SMV Model Checker to verify $M \models \psi$.
Output 2: Network $M$ satisfies $\psi$: True or False
```

will generate some traces or executions. The temporal logic formula describing these traces are time-bounded. Next, we will introduce the time-bounded linear temporal logic (BLTL).

1.3.3 Time-Bounded Linear Temporal Logic (BLTL)

Similar to the LTL formula, a time-bounded LTL (BLTL) formula is constructed by the atomic propositions (AP) using boolean connectives and bounded temporal operators. The syntax of the logic is expressed as

$$\phi ::= AP \mid \phi_1 \vee \phi_2 \mid \phi_1 \wedge \phi_2 \mid \neg\phi_1 \mid \phi_1 \mathbf{U}^t \phi_2.$$

There are three frequently used time-bounded operators: **F**, **G**, and **U**, which are defined as: $\mathbf{F}^t\phi$ or $\mathbf{F}(\leq t)[\phi]$ means ϕ holds true within time t; $\mathbf{G}^t\phi$ or $\mathbf{G}(\leq t)[\phi]$ means ϕ holds true globally up to time t; the time-bounded until operator $\phi_1\mathbf{U}^t\phi_2$ or $\phi_1\mathbf{U}(\leq t)\ \phi_2$ means, *within* time t, ϕ_1 will hold until ϕ_2 becomes true. The basic BLTL operators $\mathbf{F}^t$, $\mathbf{G}^t$, $\mathbf{U}^t$ can also be combined together to construct a composite operator to verify some complicated properties. In our work, we have used the formula $F^{t_1}G^{t_2}[\phi]$ to describe the property that ϕ holds true within time t_1 and will be globally true up to time t_2. The semantics of time-bounded LTL is defined with respect to the trace of a stochastic system. If σ^k denotes the trace starting at the step k, then, $\sigma^k \models \phi$ means that, the track σ^k satisfies the bounded LTL formula ϕ. The semantics of time-bounded LTL is written as

- $\sigma^k \models AP$ if and only if AP holds true in s_k;
- $\sigma^k \models \phi_1 \vee \phi_2$ if and only if $\sigma^k \models \phi_1$ or $\sigma^k \models \phi_2$;
- $\sigma^k \models \phi_1 \wedge \phi_2$ if and only if $\sigma^k \models \phi_1$ and $\sigma^k \models \phi_2$;
- $\sigma^k \models \neg\phi_1$ if and only if $\sigma^k \models \phi_1$ does not hold;
- $\sigma^k \models \phi_1 \mathbf{U}^t \phi_2$ if and only if there exists $i \in N$ such that, for each $0 \leq j < i$, $\sigma^{k+j} \models \phi_1$, and if $\sum_{0 \leq l < i} t_{k+l} \leq t$, then $\sigma^{k+i} \models \phi_2$.

The interested readers could refer to [21, 25] for more details about the BLTL semantics and formulas.

1.3.4 Probabilistic Model Checker PRISM

Most cellular signaling pathways or genetic network stochastic simulation models are continuous-time stochastic processes. PRISM is a popular probabilistic model checker, which can automatically and formally model and verify three types of probabilistic models, including the discrete-time Markov chains, Markov decision processes, and continuous-time Markov chains (CTMCs) models, and the PRISM model file is given an extension .sm.

Algorithm 6: PRISM Model Checking Pseudocode

```
Part1 : ctmc // Continuous-Time Markov Chains Model: model.sm
Input: Probabilistic $M$;
          Bounded temporal logic formula $\psi$
          const double c = 0.1; // Declare constant c
          const int N = 100; // Declare constant N
module NAME //Starter of a module NAME;
Gene1: [0..N] init N; // Initialize variable and assign values;
[] predicate $\rightarrow$ rates: updates // State update form;
[] geneA >0 $\rightarrow$ c*geneA: geneA' = geneA - 1 // Update geneA's state with a rate c*geneA
endmodule
Part 2: PRISM Verification: property.csl
const double T = 80; define a constant time T = 80 seconds;
const double p = 0.9; define a constant probability p = 90%;
P>=p [$F$ <=T (geneA >= 50)]; check the property is true/false;
P=? [F<=T (geneA >= 50)]; estimate the probability;
Output: True/False, or Estimate the probability that $\psi$ is true
```

Algorithm 6 summarizes the procedure to write PRISM code to verify a CTMC stochastic model. In our stochastic simulation models of cellular networks, the PRISM code starts with the keyword **ctmc**. We first should define some constants using the keyword **const**, for example, the reaction rates. All the PRISM programs should contain at least one module using the keyword "**module ... endmodule**", which contains all the variables and updates. The variables should be declared and initialized first. For example, we can define the possibles for the geneA using the code: "geneA: [0..N] **init** N;", which means, geneA takes an integer value ranging from 0 to N, and its initial value is specified (as N) with the keyword **init**.

PRISM can implement both synchronous and asynchronous update of modules, so it can model most biochemical reactions occurring asynchronously in the cell. The state update of each variable in the module are decided by the "predicate" and "updates" taking the form: [] **predicate** $\rightarrow$ **rates** : **updates**. That is, for each variable X, if the $predicate$ is true, then, the states of the corresponding variable X in the module will be updated "asynchronously" (using an "empty" square bracket []). The variable X's value will be updated to a new value X' according to the "$updates$" rules in the form of $X' = f(X)$ with a rate of "$rates$". The "$rates$" should be proportional to some known parameters (e.g., reaction rates) and the number of molecules that each variable represents.

In the continuous-time Markov chain models, we will use the continuous stochastic logic (CSL), which is a property specification language for CTMC process, to specify the temporal logic properties. The PRISM can formally verify two types of properties. The first type of property is to verify or falsify the formula $P_{\geq p}$ $[\phi]$, which is an assertion, the answer could be true or false. PRISM applies the Wald's sequential probability ratio test (SPRT) to check a given CSL formula on-the-fly when the simulation traces can give an answer ("True", "False") at a given confidence level p. For example in the Algorithm 6, the property "$P_{>=p}$ $[\mathbf{F}_{<=t}$ $(geneA \geq 50)]$" means, within t seconds, the probability that "the number of geneA molecules in the cell will be no less than 50" is at least p. This verification procedure is similar to the hypothesis testing-based statistical model checking method [25, 47, 48]. The second type of CSL formula that PRISM can analyze is to use the confidence interval (CI) method to estimate a numerical value of a bounded property $P_{=?}[\phi]$, that is to estimate the probability that the formula could be true. For example, "$P_{=?}$ $[\mathbf{F}_{<=t}$ $(geneA \geq 50)]$" estimates the probability that the number of geneA molecules will be no less than 50 within t seconds. PRISM could run the simulation to check the property on the fly and estimate the probability.

Now, we have introduced change-points detection algorithm, stationary and time-varying dynamic Bayesian network structure learning algorithm, signed integer weights estimation algorithm, symbolic model checking technique SMV and probabilistic model verification technique PRISM. Next, we will discuss how to integrate all these methods in a unified framework to reconstruct regulatory networks.

1.4 Integrative Data Analysis

The Drosophila microarray data [4], which measures 4028 genes' expression levels of Drosophila at four stages: embryonic, larval, pupal periods, and adulthood with 67 time points, has been used to study the regulatory networks that are involved in the muscle development. Some algorithms [33, 40, 41] have been developed to reconstruct undirected networks associated with the muscle development. Since the number of time points is significantly smaller than the number of genes, some dimensional reduction methods have been used to select a small number of genes to study. In this section, we will discuss the procedure how to integrate these methods together to analyze the Drosophila data.

We have known the real change-points of the Drosophila microarray data [4] are at the location/index (31, 41, 59). In Fig. 1.4, we applied the stationary DBN to reconstruct a time-invariant network which is composed of 11 genes from all the 67 time-point observations. Some studies have indicated that the network structure should be different in the Drosophila's life cycle. Below, we will reconstruct the time-varying networks at different stages for comparison.

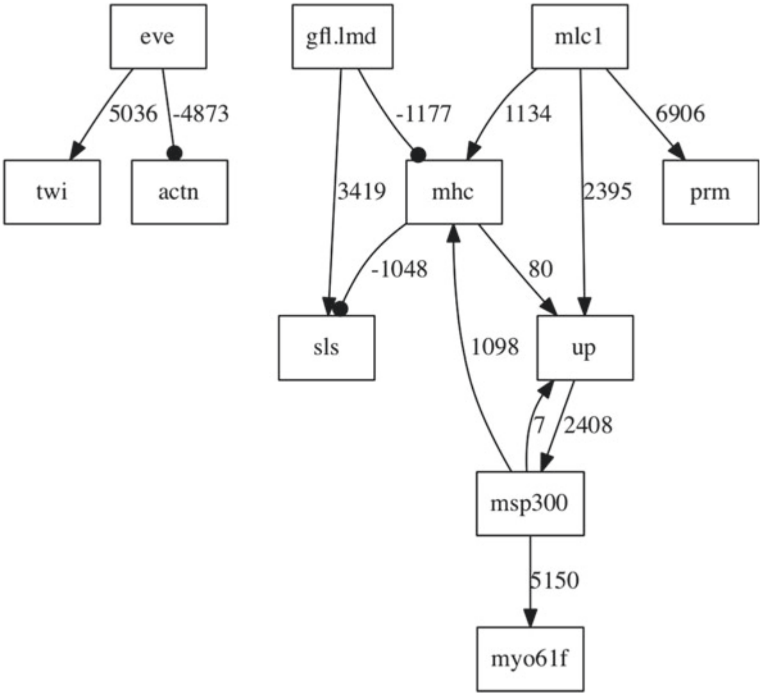


Fig. 1.4 Reconstruction of an optimal regulatory network using stationary DBN from all the 67 time-point observations in the Drosophila's life cycle

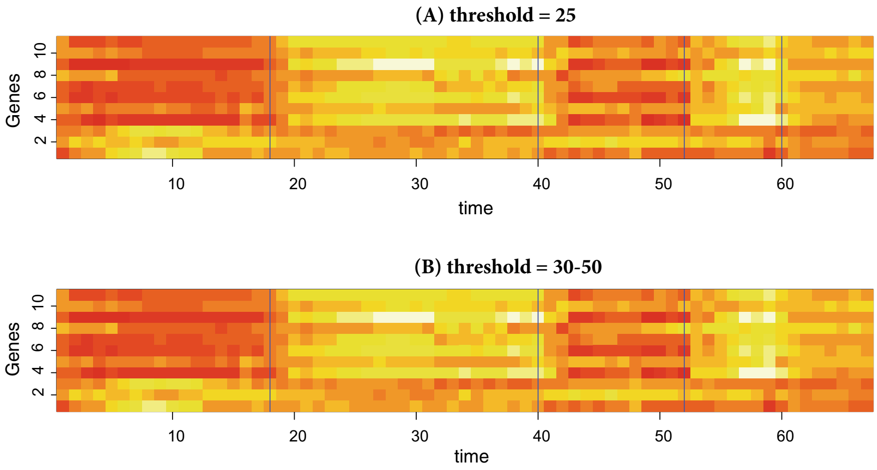


Fig. 1.5 Estimation of change-points from 11 genes, where the blue vertical lines represent the change-point lines with different threshold values (**A**: threshold = 25; **B**: threshold = [30, 50])

We will apply the INSPECT algorithm to infer the change-points assuming both the number and location of change-points are unknown. Most change-points detection algorithms are sensitive to some parameters when analyzing the high-dimensional data. INSPECT is sensitive to at least one parameter, for example, the parameter "threshold", which is used to test whether an identified change-point is a true changepoint.

We first identified the change-points from the low-dimensional dataset which has only eleven genes that are involved in the wing muscle development identified in the previous studies [12, 50]. Figure 1.5 plots the inferred change-points on the heatmap of 11 genes, the blue vertical lines represent the change-points. Figure 1.5A has 4 change-points (18, 40, 52, 60) with a threshold value 25, while Fig. 1.5B has 3 change-points (18, 40, 52) with a threshold value ranging within [30,50]. Our results indicate that the parameter's value influences the number of change-points only, it does not change the locations. Then, we test the performance of the INSPECT algorithm on high-dimensional data. We randomly sampled 400 genes from the 4028 genes and repeat 100 times and count the frequencies for the identified change-points. Figure 1.6 plots some histograms of estimated change-point locations using different threshold values. The results are similar to the Fig. 1.5, a smaller threshold value will lead to more change-points, while a larger threshold value will infer fewer change-points. To find a threshold value that is universal to any size of high-dimensional data is not realistic, our studies found when you change the data size, the previous threshold value will not work anymore.

After the change-points are identified, the next step is to apply dynamic Bayesian network inference method and signed integer weight estimation algorithm to infer optimal time-varying regulatory networks of Drosophila's muscle development.

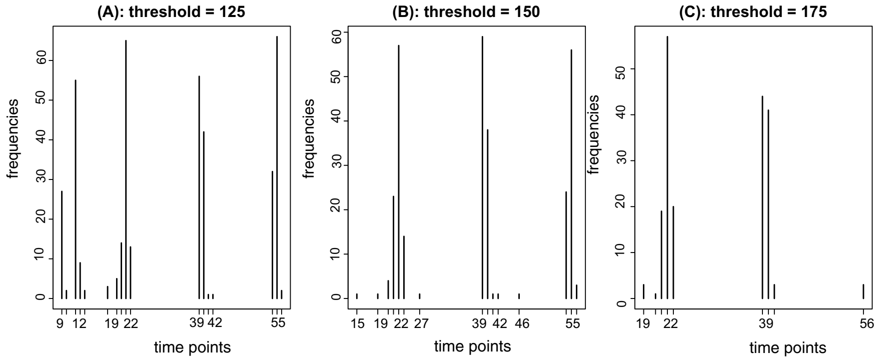


Fig. 1.6 Histograms of estimated change-point locations using different threshold values. A smaller threshold value will lead to more change-points (**A**), while a larger threshold value will infer fewer change-points (**C**)

Since the change-points are already known, the data can be splitted into four different subsets for the network reconstruction at different stages individually.

Figure 1.7 illustrates four optimal regulatory networks during the Drosophila's life cycle from the embryonic (A), larval (B), pupal (C) to adulthood (D). The solid lines with arrows represent activation, while the circle-head arrows represent inhibition. The integers on the directed edges are signed integer weights, which describe the interaction strength between two nodes and regulatory relationship. Apparently, these four optimal regulatory networks undergo systematic rewiring, that is, they are not invariant in the Drosophila's life cycle, but most regulatory relationships are still conserved. Figure 1.7 shows that, in the embryonic stage, the gene msp300 is an upstream gene, it can continuously activate several downstream genes, including the mlc1, up, eve, and myo61f to promote the embryonic development. But when the cell enters the larval, pupal, and adulthood stage, msp300's activities will be regulated by some of its previous downstream genes, for example, it could be inhibited by the mlc1 and but activated by the mhc gene. This result could explain the previous experimental discovery [45] that msp300 regulates the actin-dependent nuclear anchorage. Our previous work [43] first identified msp300 as a hub gene in the muscle development and explained the experimental discoveries.

This network reconstruction method is very sensitive to some parameters due to the small number of measurements, and this method could infer more than one optimal network at the same stage using the same data. Figure 1.8 illustrates another inferred four top-scoring non-identical networks at different stages. Comparison of Figs. 1.7 and 1.8 shows several differences in these optimal network structures. Our technique could output any number of top-scoring optimal regulatory networks. These networks are all "statistically" optimal, it does not mean that they are biologically correct. The next step is to verify or falsify the inferred network and choose the best network that is consistent with the experiments and used for further data analysis and simulation. Next, we will discuss how to apply SMV and PRISM model checker for the formal

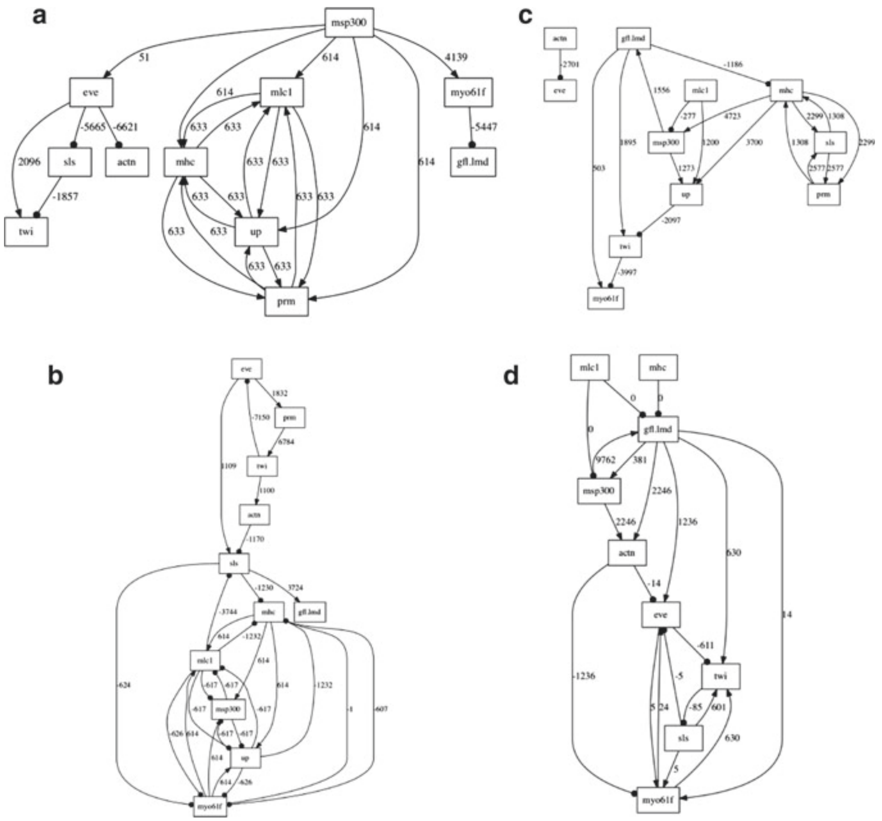


Fig. 1.7 Optimal regulatory networks during the Drosophila's life cycle from the embryonic (**A**), larval (**B**), pupal (**C**) to adulthood (**D**)

analysis of inferred networks. The below examples are used for demonstration only, the interested readers could refer to our previous work [21, 22, 35, 43] for more details.

Before the application of model checkers for network verification, we need to build a model first to describe the system. During the SMV network verification, we prefer to build a discrete value model with fewer parameters than other types of models, that is, each variable or gene can only take discrete values. For example, we can assume that each gene or variable can take three possible values $\{-1, 0, 1\}$, which represent down-regulated, normal, and up-regulated, and the initial state is randomly assigned a value of either 0 or -1. Note, we can also assume that the variable can take n possible values $\{1, 2, \ldots, n\}$ if needed. Then, we use state transfer functions to update the state of each variable at different stages. For example, the SMV code for the state update of the gene "twi" in the adulthood stage is not only dependent

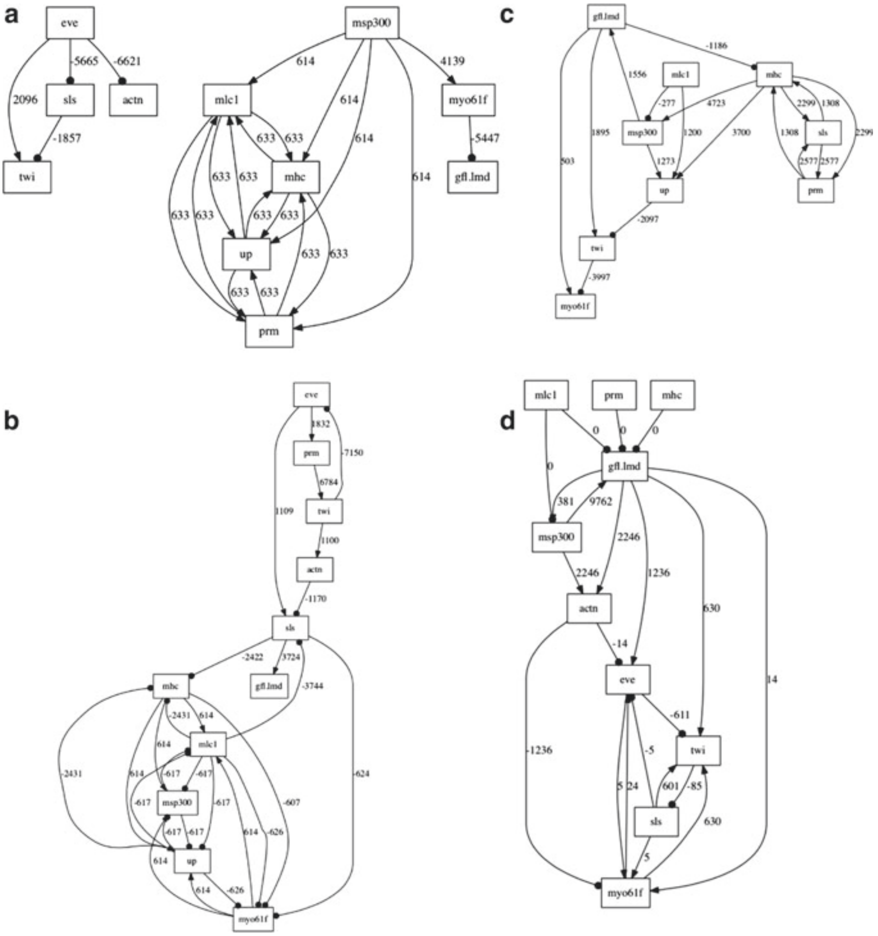


Fig. 1.8 Another four top-scoring regulatory networks during the Drosophila's life cycle

on the values of the parental genes myo61f, sls, glf.lmd, and eve, but also dependent on the signed integer weights:

```
next(twi) :=
          case
               630*myo61f + 601*sls + 630*gfl.lmd - 611*eve > 0  : 1;
               630*myo61f + 601*sls + 630*gfl.lmd - 611*eve = 0  : 0;
               630*myo61f + 601*sls + 630*gfl.lmd - 611*eve < 0  :-1;
          esac;
```

For illustration, we designed some putative CTL formulas to show how to translate some experimental results or existing database into temporal logic formulas for the SMV model checker to verify or falsify the inferred networks in Table 1.1.

Table 1.1 Putative CTL formulas for the regulatory networks verification

Property	CTL formula
P1	$AG(msp300 > 0 \rightarrow AX(mlc1 = 1 \,\&\, up = 1 \,\&\, eve = 1))$
P2	$AG(msp300 > 0 \rightarrow AF(twi < 0 \,\&\, sls < 0 \,\&\, prm < 0))$
P3	$AG(msp300 > 0 \rightarrow AF(twi > 0 \,\&\, sls > 0 \,\&\, prm > 0))$
P4	$EG((mlc1 = 1 \mid msp300 = 1 \mid gfl.lmd = 1) \rightarrow EF(twi \leq 0 \,\&\, up \geq 0))$
P5	$AG((mlc1 = 1 \mid msp300 = 1 \mid gfl.lmd = 1) \rightarrow AF(twi \leq 0 \,\&\, up \geq 0))$
P6	$AG(mhc{=}1 \rightarrow AF(prm = 1 \,\&\, mlc1 = 1 \,\&\, up = 1))$
P7	$AG((mhc = 1 \rightarrow msp300 = 1) \rightarrow EF(msp300 = 1 \,\&\, mhc \leq 0))$
P8	$AG((mhc = 1 \rightarrow AF(prm = 1)) \,\&\, (prm = 1 \rightarrow AF(sls = 1))$ $\&\, (sls = 1 \rightarrow AF(mhc \geq 0)))$
P9	$AG((sls = 1\ ) \rightarrow AF(mhc = 1 \,\&\, mhc \leq 0))$
P10	$AG((msp300 = 1 \rightarrow AF(mlc1 = \text{-}1)) \,\&\, (mlc1 = 1 \rightarrow AF(myo61f = \text{-}1))$ $\&\, (myo61f = 1 \rightarrow AF(msp = 1)))$
P11	$AG((sls = 1 \rightarrow AF(myo61f = 1)) \,\&\, (myo61f = 1 \rightarrow AF(eve = \text{-}1))$ $\&\, (eve = 1 \rightarrow AF(twi = \text{-}\ 1)) \,\&\, (twi = 1 \rightarrow AF(sls = \text{-}\ 1)))$

Our recent work [43] has verified or falsified several temporal logic formulas related to the time-varying regulatory network of Drosophila. We list some similar CTL formulas in Table 1.1 to explain how to construct temporal logic formulas and their meanings in biology, but we will not provide the verification results in this chapter. The interested readers could refer to [43] for similar examples and computer code. The formulas P1-P3 describe the properties related to the gene msp300. These formulas have the operators **AX** and **AF**. P1 means, the activation of msp300 will activate the genes mlc1, up, and eve if they are msp300's downstream genes. This type of formula could intelligently identify many downstream genes at the same time in the large network. P2-P3 are used to check whether or not there exists a path on which msp300's overexpression will finally inhibit (P2) or activate (P3) the genes twi, sls, and prm's expression levels. Similar formulas like the P1-P3 can be designed to check some hub genes and their downstream genes' behaviors. Formulas P4-P5 describe the properties of (twi, up)'s parental genes mlc1, msp300 and gfl.lmd, for some (P4) or all (P5) paths, it is globally true that either mlc1 or msp300 or gfl.lmd's overexpression will finally inhibit twi's activity but promote the gene up's expression. The formula P4 with the "EF" operator is weaker than P5 with the "AF" operator in verification. P6 checks whether or not the mhc's overexpression will finally promote the genes prm, mlc, and up's activity. Property 7 describes a negative feedback loop between mhc and msp300. P8-P11 describes a sequence of reaction events, which can be easily expressed using the CTL formula, then verified or falsified by the SMV model checker. These properties are not easy to be simulated using traditional computational methods (e.g., stochastic simulation or differential equations). Using the traditional simulation methods, you have to estimate many unknown parameters if the computational models are complicated which is very challenging.

Our SMV verification technique can avoid using some unknown parameters. In our previous work, we have applied this technique to identify some key biomarkers and processes that can initiate the cell's apoptosis or proliferation. If we can translate the KEGG database into CTL formulas, the SMV model checker could automatically verify/falsify whether or not the inferred networks are consistent with the KEGG database, instead of manually checking with the database. If the formula is satisfied, SMV will output "True", else it will output "False" with a counter-example. The networks satisfying all or most properties will be biologically correct or reasonable, which can be used for further analysis or modeling and predictions.

Finally, we will obtain one or more than one inferred optimal networks that are verified to be consistent with most existing experiments by the SMV model checker. Sometimes, given some small networks, we can use stochastic continuous models to describe the inferred optimal networks. The probabilistic model checker PRISM, based on sequential probability ratio test and confidence interval estimation methods, can formally analyze some quantitative properties of the system described by the bounded temporal logic formulas.

Table 1.2 designed some putative BLTL formulas for the PRISM analysis if we could build a stochastic continuous model. Formula P1 means that, within time t, the number of "prm" molecules will be at least 100 with a probability of at least 0.9. Formula P2 checks whether the number of "msp300" molecules will be continuously greater than 50 from time 10 to t. PRISM can check whether these formulas are true or not with different values of time t. PRISM can also estimate the probability that the number of "prm" molecules will be at least 100 within time t, which is expressed as P3. P4 will estimate the probability that the msp300's level will be above 50 all the time during the time interval [10, 30]. However, the continuous models contain many unknown parameters than the discrete value models, including the rates of reaction, synthesis, binding or degradation. Most of these parameters are not easy to be estimated from the experiments. The formulas in Table 1.2 are used for demonstration only since we do not know the parameters in the inferred regulatory network. We usually use PRISM to formally analyze some small networks with known parameters. Our previous work [21] has applied PRISM to formally analyze a stochastic signaling pathway model, the interested readers could refer to [21] for computer code and more details.

Table 1.2 Two types of putative BLTL formulas used for the PRISM model verification. These formulas are used for demonstration only

	BLTL formula
Assertion P1	$P_{\geq 0.9}(\phi_1) = P_{\geq 0.9}\ [F_{\leq t}\ (prm \geq 100)]$
Assertion P2	$P_{\geq 0.9}(\phi_2) = P_{\geq 0.9}\ \{G_{[10,t]}\ (msp300 > 50)\}$
Estimation P3	$P_{=?}(\phi_3) = P_{=?}\ [F_{\leq t}(prm \geq 100)]$
Estimation P4	$P_{=?}(\phi_4) = P_{=?}\ \{G_{[10,30]}\ (msp300 > 50)\}$

1.5 Discussions

In this work, we discussed how to integrate the machine learning and model checking methods to reconstruct regulatory network from the high-dimensional data. In the integrative approach, the change-points detection algorithm is first applied to identify the change-points which describe when the system moves to a new stage; then, the dataset is splitted into subsets for individual analysis according to the stages. The stationary dynamic Bayesian network method is applied to learn the optimal network structure at different stages; in the meantime, a signed integer weight estimation algorithm is used to learn the regulatory relationship (activation/inhibition). There could be several inferred optimal networks using the current network inference methods, then SMV model checker will be applied to formally verify the inferred time-varying networks by checking some temporal logic formulas which are abstracted from experiment or KEGG. The optimal and biologically correct network should be consistent with existing experiments and known databases. Given a continuous stochastic model, PRISM was introduced for further analysis of the networks by checking some time-bounded temporal logic formulas. However, PRISM is only applicable to some small networks with all parameters already known to us because it is not realistic to estimate many model parameters in large networks.

This work and our recent study [43] found that the change-points detection algorithm is sensitive to the parameter's values and the data size in the high-dimensional time-series data analysis. It is not possible to find a universal parameter value for the change-points estimation according to our analysis. So, the change-points detection is still one of the most challenging problems in network reconstruction. During the time-varying network structure learning, we assume the network structures are independent at different stages, so we can split the data according to the change-points and infer the networks and estimate the integer weights at different stages individually. However, the networks might not be independent at different stages. In the long run, we need to develop a real time-varying network structure learning algorithm that does not assume the stage independence, and propose novel change-points estimation algorithms to overcome the shortcomings of previous methods in the time-varying network reconstruction.

Acknowledgements This work was partially supported by the NIH-NIGMS grant 1R15GM129696-01A1.

References

1. Ahmed A, Xing E (2009) Recovering time varying networks of dependencies in social and biological studies. Proc Natl Acad Sci 106:11878–11883
2. Akaike H (1974) A new look at the statistical model identification. IEEE Trans Autom Control 19
3. Akutsu T, Miyano S, Kuhara S (2000) Inferring qualitative relations in genetic networks and metabolic pathways. Bioinformatics 16:727–734

4. Arbeitman M, Furlong E, Imam F, Johnson E et al (2002) Gene expression during the life cycle of drosophila melanogaster. Science 297:2270–5
5. Barry D, Hartigan J (1992) Product partition models for change point problems. Ann Stat 20(1):260–279
6. Bryant R (1986) Graph-based algorithms for boolean function manipulation. IEEE Tran Comput 35(8):677–691
7. Ceccarelli M, Cerulo L, Santone A (2014) De novo reconstruction of gene regulatory networks from time series data, an approach based on formal methods. Methods 69(3):298–305
8. Chow C, Liu C (1968) Approximating discrete probability distributions with dependence trees. IEEE Trans Info Theory 14
9. Clarke EM, Grumberg O, Peled DA (1999) Model checking. MIT Press
10. Darling D, Erdos P (1956) A limit theorem for the maximum of normalized sums of independent random variables. Duke Math J 23:143–155
11. Doering T et al (2012) Network analysis reveals centrally connected genes and pathways involved in cd8+ t cell exhaustion versus memory. Immunity 37
12. Dondelinger F, Lebre S, Husmeier D (2013) Non-homogeneous dynamic bayesian networks with bayesian regularization for inferring gene regulatory networks with gradually time-varying structure. Mach Learn 90
13. Fearnhead P (2006) Exact and efficient Bayesian inference for multiple change point problems. Stat Comput 16(2):203–213
14. Feuz K, Cook D, Rosasco C, Robertson K, Schmitter-Edgecombe M (2014) Automated detection of activity transitions for prompting. IEEE Trans Human-Mach Syst 45(5):1–11
15. Friedman J, Hastie T, Tibshirani R (2007) Sparse inverse covariance estimation with the graphical lasso. Biostatistics, pp 1–10
16. Friedman N, Linial M, Nachman I, Pe'er D (2000) Using bn to analyze expression data. J Comput Biol 7:601–620
17. Friedman N, Murphy K, Russell S (1998) Learning the structure of dynamic probabilistic networks. In: Proceedings of the 14th conference on the uncertainty in artificial intelligence
18. Fujita A, Sato J et al (2007) Time varying modeling of gene expression regulatory networks using the wavelet dynamic vector autoregressive method. Bioinformatics 23(13):1623–1630
19. Gong H (2013) Analysis of intercellular signal transduction in the tumor microenvironment. BMC Syst Biol 7:S5
20. Gong H, Feng L (2014) Computational analysis of the roles of er-golgi network in the cell cycle. BMC Syst Biol 8:S4
21. Gong H, Feng L (2014) Probabilistic verification of er stress-induced signaling pathways. In: Proceedings of IEEE international conference on bioinformatics and biomedicine
22. Gong H, Klinger J, Damazyn K, Li X, Huang S (2015) A novel procedure for statistical inference and verification of gene regulatory subnetwork. BMC Bioinform V16:S7
23. Gong H, Wang Q, Zuliani P, Clarke E (2011) Formal analysis for logical models of pancreatic cancer. In: 50th IEEE conference on decision and control and European control conference
24. Gong H, Zuliani P, Komuravelli A, Faeder J, Clarke E (2012) Computational modeling and verification of signaling pathways in cancer. Proceedings of algebraic and numeric biology, LNCS 6479
25. Gong H, Zuliani P, Komuravelli A, Faeder JR, Clarke EM (2010) Analysis and verification of the HMGB1 signaling pathway. BMC Bioinform 11(7)
26. Green P (1995) Reversible jump markov chain monte carlo computation and bayesian model determination. Biometrika 82:711–732
27. Grzegorczyk M, Husmeier D (2009) Nonstationary continuous dynamic bayesian networks. Adv Neural Inf Process Syst (NIPS) 22:682–690
28. Heckerman D, Geiger D, Chickering D (1995) Learning bayesian networks: the combination of knowledge and statistical data. Mach Learn 20(3)
29. Horwitz B (2003) The elusive concept of brain connectivity. NeuroImage 19:466–470
30. Kawahara Y, Sugiyama M (2009) Sequential change-point detection based on direct density-ratio estimation. In: SIAM international conference on data mining, pp 389–400

31. Kim S, Imoto S, Miyano S (2003) Inferring gene networks from time series microarray data using dynamic bayesian networks. Brief Bioinform 4:228–235
32. Kim S, Imoto S, Miyano S (2004) Dynamic bayesian network and nonparametric regression for nonlinear modeling of gene networks from time series gene expression data. BioSystems 75:57–65
33. Lebre S, Becq J, Devaux F, Stumpf M, Lelandais G (2010) Statistical inference of the time-varying structure of gene-regulation networks. BMC Syst Biol 4:130
34. Luscombe N et al (2004) Genomic analysis of regulatory network dynamics reveals large topological changes. Nature 431:308–312
35. Ma Y, Damazyn K, Klinger J, Gong H (2015) Inference and verification of probabilistic graphical models from high-dimensional data. Lect Notes Bioinform 9162
36. Mazumder R, Hastie T (2012) The graphical lasso: new insights and alternatives. Electron J Stat 6:2125
37. Ong I, Glasner J, Page D (2002) Modelling regulatoruypathways in e. coli from time series expression profiles. Bioinformatics 18:S241–S248
38. Parvu O, Gilbert D (2016) A novel method to verify multilevel computational models of biological systems using multiscale spatio-temporal meta model checking. PLOS One
39. Reddy S, Fun M, Burke J, Estrin D, Hansen M, Srivastava M (2010) Using mobile phones to determine transportation modes. ACM Trans Sens Netw 6(2):1–27
40. Robinson J, Hartemink R (2010) Learning non-stationary dynamic bayesian networks. J Mach Learn Res 11:3647–3680
41. Schwaller L, Robin S (2016) Exact bayesian inference for off-line change-point detection in tree-structured graphical models. Stat Comput 27(5)
42. Wang T, Samworth R (2017) High dimensional change point estimation via sparse projection. In: Statistical Methodology
43. Wang Z, Guo Y, Gong H (2019) An integrative analysis of time-varying regulatory networks from high-dimensional data. In: IEEE international conference on big data, pp 3798–3807
44. Yoshida R, Imoto S, Higuchi T (2005) Estimating time-dependent gene networks form time series microarray data by dynamic linear models with markov switching. CSB05, IEEE CSBC
45. You J, Starr D, Wu X, Parkhurst S, Zhuang Y, Xu T, Xu R, Han M (2006) The kash domain protein msp-300 plays an essential role in nuclear anchoring during drosophila oogenesis. Deve Biol 289(2):336–45
46. You Y, Wang T, Samworth R (2015) A useful variant of the davis-kahan theorem for statisticians. Biometrika 102:315–323
47. Younes HLS, Simmons RG (2002) Probabilistic verification of discrete event systems using acceptance sampling. CAV, LNCS 2404:223–235
48. Younes HLS, Simmons RG (2006) Statistical probabilistic model checking with a focus on time-bounded properties. Inf Comput 204(9):1368–1409
49. Yu J, Smith V, Wang P, Hartemink A, Jarvis E (2004) Advances to bayesian network inference for generating causal networks from observational biological data. Bioinformatics 20:3594–3603
50. Zhao W, Serpedin E, Dougherty E (2006) Inferring gene regulatory networks from time series data using the minimum description length principle. Bioinformatics 22

Chapter 2
Differential Expression Analysis of RNA-Seq Data and Co-expression Networks

Sana Javed

Abstract At present, RNA-seq has become the most common and powerful platform in the study of transcriptomes. A major goal of RNA-seq analysis is the identification of genes and molecular pathways which are differentially expressed in two altered situations. Such difference in expression profiles might be linked with changes in biology giving an indication for further intense investigation. Generally, the traditional statistical methods used in the study of differential expression analysis of gene profiles are restricted to individual genes and do not provide any information regarding interactivities of genes contributing to a certain biological system. This need led the scientists to develop new computational methods to identify such interactions of genes. The most common approach used to study gene-set interactivities is gene network inference. Co-expression gene networks are the correlation-based networks which are commonly used to identify the set of genes significantly involved in the occurrence or presence of a particular biological process. This chapter describes a basic procedure of an RNA-seq analysis along with a brief description about the techniques used in the analysis: an illustration on a real data set is also shown. In addition, a basic pipeline is presented to elucidate how to construct a co-expression network and detect modules from the RNA-seq data.

Keywords RNA-seq · Normalization · Co-expression networks · Modules

2.1 Systems Biology

Systems biology is an interdisciplinary field comprising various statistical and mathematical techniques along with modelling of complicated biological systems. This field is concerned to reveal complex phenomenon occurring in biological processes at different levels like cellular, molecular or organism [1]. One major goal of systems biology is to model and deduce emergent properties of different interactive processes occurring in biological systems whose theoretical description might

S. Javed (✉)
Department of Mathematics, COMSATS University Islamabad, Lahore Campus, Pakistan
e-mail: sana.javed17@gmail.com

T. D. Pham et al. (eds.), *Advances in Artificial Intelligence, Computation, and Data Science*, Computational Biology 31,
https://doi.org/10.1007/978-3-030-69951-2_2

be given only through systems biology methodologies [2]. Such approaches usually involve the information about metabolic networks or cell signalling networks [3]. The Human Genome Project has opened a new door for researchers to conduct collaborative research on the biological processes using information provided at gene level [4]. Researchers use gene expression profiles obtained via different platforms like microarray, SOLid or HeliScope for further analysis using computational methods.

2.2 High Throughput Sequencing

Microarray analysis was in use at a wider level by scientists in past decades to identify the gene expression profiles. Microarray is a tool that contains thousands of short DNA elements called probes on a chip which are used in hybridization of a cDNA/cRNA library to see relative profusion of nucleic acid sequence in the library. However, microarray analysis still has some limitations and one of them is the preparation of adequate target DNA samples and DNA fragments spotted slides. To overcome this issue high throughput sequencing (HTS) techniques have been introduced which are cost effective, rapid, enhanced and time saving [5]. HTS technologies an alternative to microarray analysis have brought the revolution in the field of molecular biology [6, 7]. These technologies read a large amount of genes at once and helps in identifying the key regulators involved in a specific process. The most common and economical platform used nowadays to measure gene expression profiles is RNA-seq and its ability comparatively to microarray is very high. RNA-seq analysis has the ability to sequence the entire transcriptome instead of just measuring predefined transcripts as in microarray analysis. A comparison between RNA-seq and microarray analysis is shown in Fig. 2.1.

2.3 RNA-seq Analysis

RNA-seq could provide us the information about the status of gene whether it is on/off, expression levels of genes and the time of their activation or deactivation. Such information assists scientists/researchers to study cell biology at intensive level and evaluate variations which might indicate disease.

A major goal of RNA-seq analysis is the identification of genes and molecular pathways which are differentially expressed in two or more environmental conditions. Such differences in expression profiles might be linked with changes in biology giving a sign for further investigation.

For instance, suppose we have two groups of cells; one group consists of normal cells while the other group includes cancer cells. It is obvious that functioning of cancer cells differs from normal cells. We are intended to identify what genetic mechanism is instigating such difference. During some cellular processes, some of the genes are active through mRNA while some are inactive. HTS provides us the

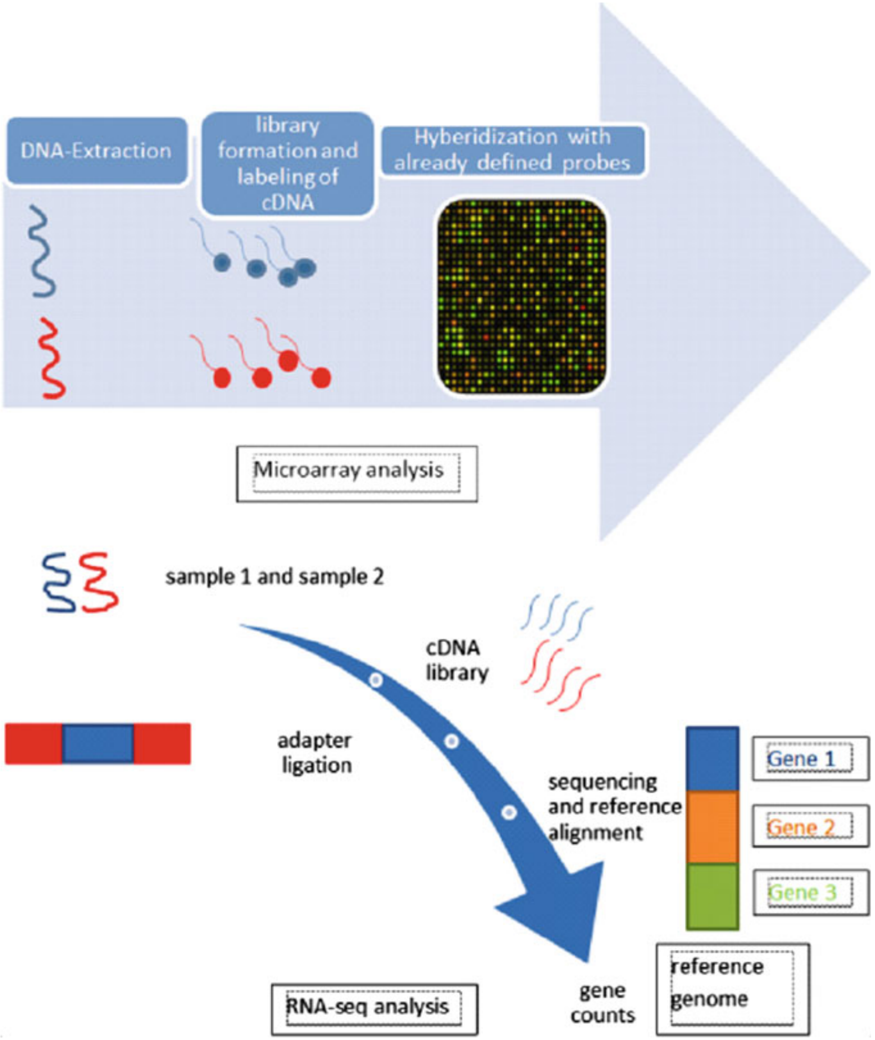


Fig. 2.1 Microarray versus RNA-seq; gene expressions in microarrays are based on the intensity while in RNA-seq they depend on sequencing

information which genes are active, and how much they are transcribed. RNA-seq method determines the sequence of nucleotides in a segment of RNA. RNA-seq analysis is used to measure the gene expressions in normal cells and cancer cells through which we can infer the variations between both groups.

A workflow of RNA-seq analysis includes the following:

- formulating a sequencing library,
- biological and technical variations,

- assessment of variations,
- method for differential expression analysis,
- generalized linear models,
- hypothesis test,
- normalization of data,
- principal component analysis,
- data analysis of gene expression profiles.

2.4 Formulating a Sequencing Library

There are several procedures and platforms to organize a sequence library but Illumina is the one that is most frequently used. A rapid progress in Illumina technology over the past decades has profoundly allocated the pace for the remarkable improvements in upshots and declines in cost. As a result, Illumina technologies lead the HTS market.

In the preparation of a sequencing library, initially we segregate the RNA and separate the RNA into small fragments due to the reason that RNA sequence might be thousands of bases long but machines can sequence only (200–300 bp) fragments. Then RNA fragments are converted into double-stranded DNA. Further, we attach sequencing adaptors which let the sequencing machine to identify the fragments yielding an opportunity to align distinct samples at the same instant as distinct samples employ distinct adaptors. There might be numerous DNA fragments with missing adaptors so PCA amplification is performed to amplify the genes containing adaptors. Further, a quality control is executed to check the library depths and library fragment lengths; they should be of moderate length. The cDNA library is then examined by HTS, constructing small sequences that correspond to either one or both ends of the fragment (see Fig. 2.2).

A location in the reference genome is determined by the genome fragments that match the read fragments. Once we determine the location for a read we can easily identify whether it lies within coordinates of the gene or it is associated with some other interesting feature. Finally, we obtain a gene expression matrix like Table 2.1 after determining the gene counts. Feature counts [8] and HTSeq [9] are the famous tools used to determine the gene counts. The total number of read counts in a particular sample sets up the expression profile or library for that sample.

2.5 Biological and Technical Variations

Technical and biological variations are two kinds of variations which might be encountered during any RNA-seq experiment. Biological variation (BV) comes when experiments are conducted on different samples under same condition while technical variation (TV) occurs due to disparity in measurements of gene counts on the

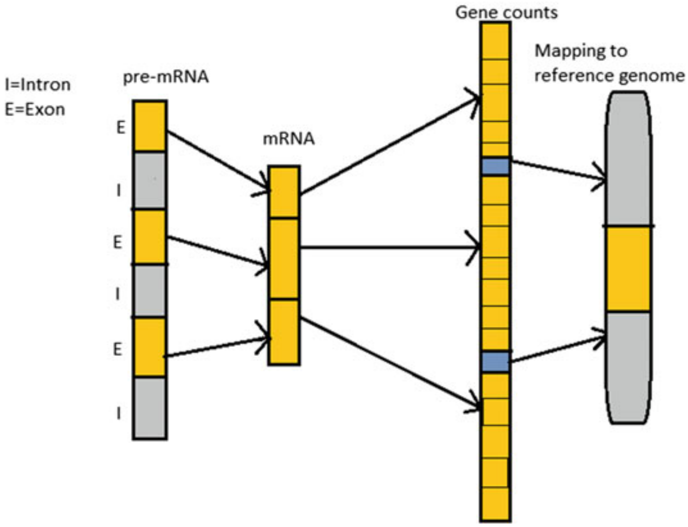


Fig. 2.2 RNA-seq process (1) estimation of gene counts from mRNA (2) alignment of gene counts back to the reference genome

Table 2.1 Gene expression matrix

Gene	Normal cells		Cancer cells	
	Sample 1	Sample 2	Sample 3	Sample 4
Gene 1	12	15	7	34
Gene 2	334	246	500	300
Gene 3	2560	2500	3000	2250

same sample run at different time points independently. So, the coefficient of total variation V is defined as follows [10]:

$$(\text{Coefficient of } V)^2 = (\text{Coefficient of } BV)^2 + (\text{Coefficient of } TV)^2$$

The coefficient of BV is the disparity through which the unspecified real concentration of the gene differs among RNA replicates, and it is not affected even if we increase library depths. On the other hand, the coefficient of TV decreases if we escalate the sequencing depth. It shows that coefficient of biological variation plays an important role in RNA-seq analysis, and it is elementary to consider differential expression analysis with respect to biological variation (BV). As many researchers still find RNA-seq experiments very expensive so it is essential to assess coefficient of BV in a more reliable way with few number of replicates by keeping in view the fact that distinct genes might have distinct levels of BV.

Although we can use the same approach to analyse the RNA-seq data like microarray in which gene counts are standardized and transformed to achieve normality [11–13], for smaller sequencing depths it is incapable to handle small gene counts. An additional prevailing and major difficulty is that count data usually displays a vigorous mean–variance connection that is not appreciated by current normal-based evaluates, providing theoretically poor statistical conclusions. This issue can be handled by considering accurate probability distributions for the gene counts and estimating their exact probabilities which subsequently provide efficient statistical inferences [14–17]. Additional benefit of fitting accurate distribution is that they provide flexibility for unravelling biological variation from technical variation [16, 17].

2.6 Assessment of Variations

The most common frameworks used to estimate the differentially expressed genes patterns in an analysis are the Poisson and the negative binomial (NB) distributions. Although Poisson distribution is more convenient to use comparatively to NB as it is based on only one parameter, it has a shortcoming that it doesn't accumulate the BV in the data [16, 18] which might bring the deviation in concentration of different genes in different RNA samples. Such dissimilarity in abundance of gene counts across distinct samples will show overdispersion of gene counts relative to Poisson model, which puts a constraint that mean and variance of the model are identical.

2.6.1 Poisson's Distribution

Let the number of read counts of gene G in sample k be denoted by z_{Gk} and the total number of genes considered in the experiment be N_G. Assume that z_{Gk} follow a Poisson distribution and the experiment is conducted on the same sample with the recurrent sequencing runs. Let ρ_{Gk} denote the fraction of read counts of gene G in the sample k $\sum_{G=1}^{N_G} \rho_{Gk} = 1$ for each sample k.

$\Rightarrow E(z_{Gk}) = \mu_{Gk} = \rho_{Gk} N_k$; $N_k =$ Total number of read counts in library k

$\Rightarrow \text{var}(z_{Gk}) = E_\rho(\text{var}(z|\rho)) + \text{var}_\rho(E(z|\rho)) = \mu_{Gk} + \varphi_G \mu_{Gk}^2$

$\Rightarrow \frac{\text{var}(z_{Gk})}{\mu_{Gk}^2} = \frac{1}{\mu_{Gk}^2} + \varphi_G \Rightarrow$ Coefficient of $V^2 = \frac{1}{\mu_{Gk}^2} + \varphi_G$,

where $\sqrt{\varphi_G} = \frac{\text{standarddeviation}}{\text{mean}}$ denote the coefficient of BV of ρ_{Gk}, while first term represents the coefficient of TV^2 of ρ_{Gk}. Though $\sqrt{\varphi_G}$ is the coefficient of BV, it also takes into account technical sources like library formation along with BV among samples [10].

2.6.2 Negative Binomial Distribution

In an RNA-seq analysis, biological replicates contain more BV as compared to technical replicates. In case of biological replicates, such issue should be tackled at an intense level otherwise it might results in false-positive rates (FPR) due to miscalculation of sampling error. NB-distribution has two parameters, mean and dispersion, which permits modelling with more general mean–variance association. It provides a precise modelling when we have low gene counts, and also accommodate BV among biological replicates through its dispersion parameter [10]. The NB framework might be expanded with quasi-likelihood (QL) approaches to explain gene-associated inconsistency encountered through biological and technical causes [19].

Let z_{Gk} represents the number of counts of gene G in k-th sample. Then $E(z_{Gk}) = \mu_{Gk}$ and $\text{var}(z_{Gk)} = \sigma_G^2(\mu_{Gk} + \varphi\mu_{Gk}^2)$ where φ and σ_G^2 denote NB and QL dispersion parameters [20]. Disparity in gene counts across different samples can be modelled through any or both of NB/QL dispersion parameters. Here, φ being a global parameter and σ_G^2 a gene-specific parameter reflects their distinct roles in the analysis. From the definitions of φ, it is clear that NB dispersion parameter accounts for the overall variation across all genes and its square root is known as the coefficient of BV [10], whereas σ_G^2 considers only gene-associated inconsistency above and below the overall level. This global parameter φ gives advantage to NB modelling over Poisson as it has aptitude of capturing innate variation in the biological system.

Generally, instead of φ a trended NB dispersion is estimated by fitting a mean-dispersion trend across all genes. This trended dispersion is basically a function $\varphi(B)$, where B represents the overall expression level of gene G [10]. This approach is more flexible and accounts for empirical mean–variance relationships as well.

Mostly we have limited numbers of replicates so it is hard to estimate QL dispersion parameter so an empirical Bayes (EB) technique is implemented by which moderated EB estimates are obtained [19, 21, 22]. This method basically compresses the raw QL dispersion estimates towards a mean-dependent trend to obtain EB estimates.

2.7 Method for Differential Expression Analysis

There are several packages available in R, which have the functions to share information across genes for dispersion evaluation like edgeR, DESeq, BBSeq, DSS, baySeq and ShrinkBayes. edgeR package shrinks dispersion/variance estimates towards a common/local estimate to obtain moderated estimates [10, 16]. DEseq considers the mean expression strength across all samples to identify and rectify very low dispersion estimates [14]. In BBSeq framework, the dispersion is modelled on the mean; the main objective of this method is to lessen the effect of outliers through the mean absolute deviance of dispersion estimations [23]. A Bayesian methodology, DSS, provided in [24] estimates the variance for separate genes that accounts for the diversity of dispersion estimates for distinct genes. baySeq [25] and ShrinkBayes [26] approaches are associated with Bayesian model and primarily relevant with the estimation of posterior probabilities or FDR for differential expressions.

2.8 Generalized Linear Model (GLM)

Consider the read counts for gene G in sample k then the mean value for these counts in a generalized linear framework will be modelled as follows:

$$\log(\mu_{Gk}) = \sum_{l=1}^{n} y_{kl}\alpha_{Gl} + \log N_k;$$

y_{kl}= sample-associated predictor for coefficient l,
α_{Gl}= gene-associated value for coefficient l; $l = 1, 2, \ldots, n$,
N_k= Total number of reads in the library k.

Each of the coefficients y_{kl} and α_{Gl}; $l = 1, \ldots, n$ depicts a feature of the experimental design. The nonzero value of y_{kl} shows its contribution to the expression of sample k while α_{Gl} defines its influence on the expression value of gene G in affected samples. As there might be different resources used in the formation of different samples so the library depths might vary from sample to sample. To overcome this issue an offset term is introduced which is primarily the log-transformed library size and confirms that such disparity in library depths will not bring any specious changes in expressions.

2.9 Hypothesis Test

There are two common choices for gene-wise testing namely likelihood ratio tests and quasi-likelihood (QL) F-test. The likelihood ratio tests are frequently used with GLM framework while QL F-test is preferred when we have a few number of replicates as they provide more robust and authentic results in such scenario. Mostly, a null hypothesis is set that the coefficient or contrast is equivalent to zero against the two-sided alternative that it is nonzero to conduct a comparative analysis of RNA-seq data. For instance, testing the null hypothesis that log fold change (FC) between control and diseased gene expression is exactly zero, i.e. gene is not affected by or playing a significant role in disease.

Table 2.2 Read counts per sample

Genes	Sample 1	Sample 2
Gene 1	500	1000
Gene 2	700	900
Gene 3	300	1100
Total	1500	3000

2.10 Normalization of Data

Normalization of data is usually done before analysis of the gene expression profiles because of the varying library/sample sizes since number of reads assigned to the samples may differ, as shown in Table 2.2. The main factor behind this variation might be the higher concentration of low-quality reads captured on the flow cell in some samples comparatively to others. To tackle this issue and adjust the read counts in each sample, the easiest way is normalization, i.e. to divide read count per gene by the sum of read counts per sample.

There are several procedures of normalization implemented in R or R-Studio. Some of them, implemented in the edgeR library of R, are described below. An initial step before normalization is to remove all the genes having 0 counts across all the samples. So the genes which are not transcribed will have no effect on the calculated scaling factors.

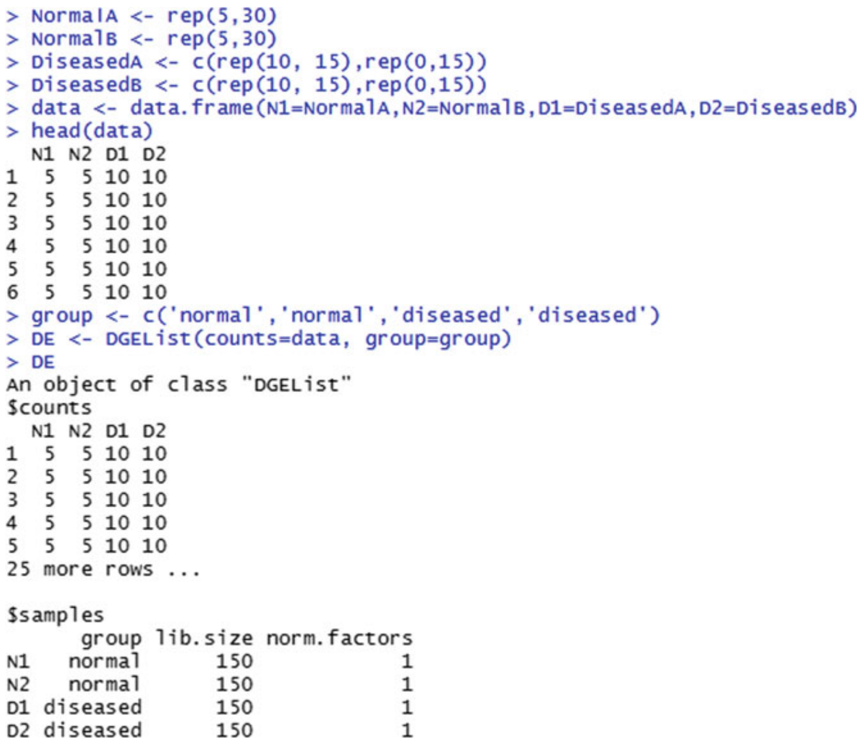
2.11 Trimmed Mean of M-values (TMM)

TMM [27] is the weighted trimmed mean of M-values with the weights from the delta method on Binomial data. In the case of unstated reference sample, the sample with upper quartile nearest to the mean upper quartile will be considered. An application on a test data is shown as following:

Firstly a dataset (Console 2.1) is created consisting of two groups normal and diseased each containing two samples. Further, a DGEList object is generated.

Finally, we have calculated the scaling factors using method TMM in edgeR library of R-Studio and run the DE test; results are shown in Console 2.2.

If we look at the normalization factor of gene 1, then it is clear that first gene is not differentially expressed among the groups because 5/0.7071068 ~ 10/1.4142136 ~ 7.07 and the test implies that only 15 genes are DE.



```
> NormalA <- rep(5,30)
> NormalB <- rep(5,30)
> DiseasedA <- c(rep(10, 15),rep(0,15))
> DiseasedB <- c(rep(10, 15),rep(0,15))
> data <- data.frame(N1=NormalA,N2=NormalB,D1=DiseasedA,D2=DiseasedB)
> head(data)
  N1 N2 D1 D2
1  5  5 10 10
2  5  5 10 10
3  5  5 10 10
4  5  5 10 10
5  5  5 10 10
6  5  5 10 10
> group <- c('normal','normal','diseased','diseased')
> DE <- DGEList(counts=data, group=group)
> DE
An object of class "DGEList"
$counts
  N1 N2 D1 D2
1  5  5 10 10
2  5  5 10 10
3  5  5 10 10
4  5  5 10 10
5  5  5 10 10
25 more rows ...

$samples
      group lib.size norm.factors
N1   normal      150            1
N2   normal      150            1
D1 diseased      150            1
D2 diseased      150            1
```

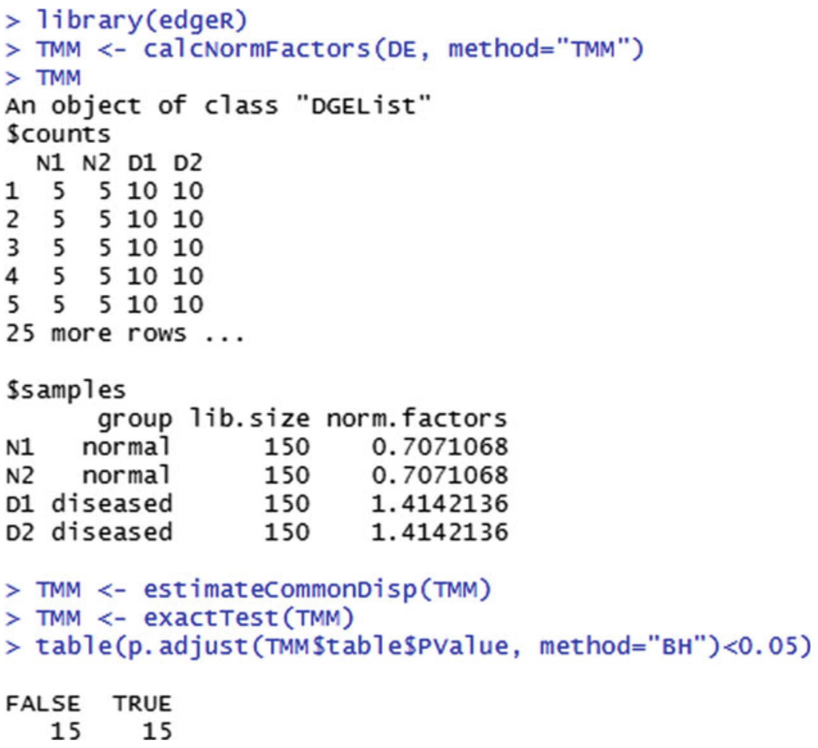
Console 2.1 Dataset along with DGEList object

2.12 Relative Log Expression (RLE)

RLE is the scaling factor technique introduced by Anders and Huber [14]. A median sample is computed from the geometric mean of all samples and the median ratio of each sample to the median one is used as the scaling factor. Console 2.3 shows the normalization factors and test results obtained via RLE method applied on the data set in Console 2.1.

2.13 Upper-Quartile Normalization

This technique, developed by Bullard et al. [28], calculates the scaling factors from the 75% quantile of the expression values for each sample. The results obtained through upper-quartile normalization method on the same data set used above are shown in Console 2.4.

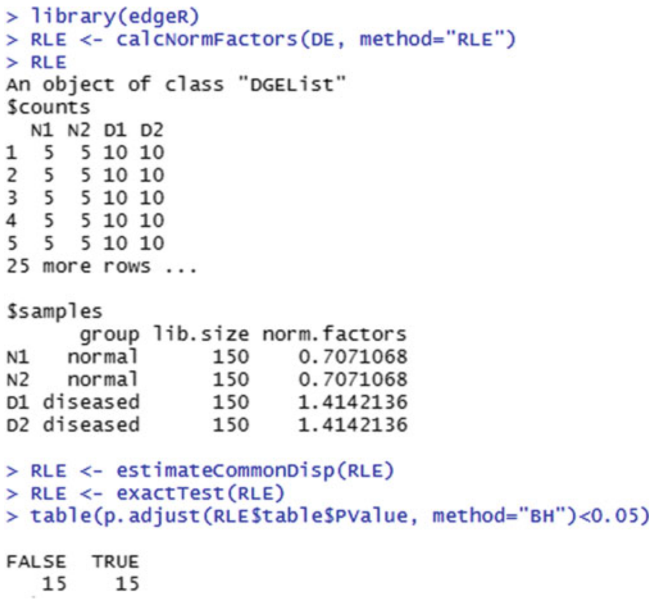


```
> library(edgeR)
> TMM <- calcNormFactors(DE, method="TMM")
> TMM
An object of class "DGEList"
$counts
  N1 N2 D1 D2
1  5  5 10 10
2  5  5 10 10
3  5  5 10 10
4  5  5 10 10
5  5  5 10 10
25 more rows ...

$samples
      group lib.size norm.factors
N1   normal      150    0.7071068
N2   normal      150    0.7071068
D1 diseased      150    1.4142136
D2 diseased      150    1.4142136

> TMM <- estimateCommonDisp(TMM)
> TMM <- exactTest(TMM)
> table(p.adjust(TMM$table$PValue, method="BH")<0.05)

FALSE  TRUE 
   15    15 
```

Console 2.2 TMM normalization

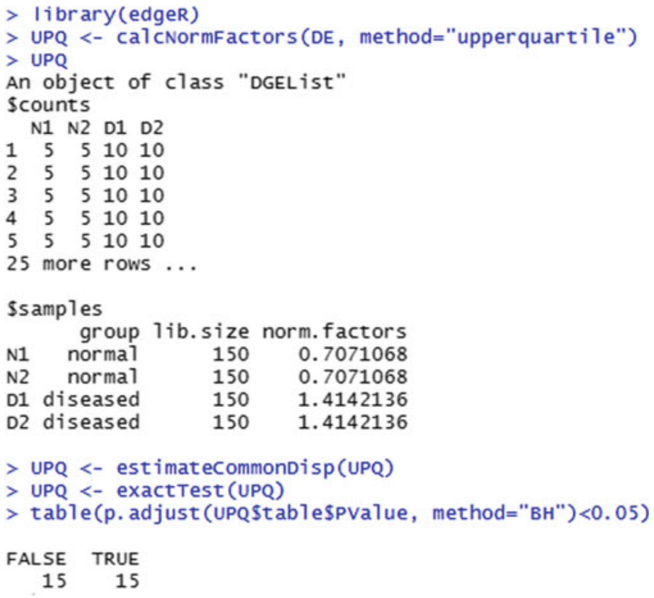
2.14 Principal Component Analysis

Principal component analysis (PCA) is used to decrease the dimension of data. Principal component analysis (PCA) plays a vital role in RNA-seq analysis as it gives an idea about insignificant features of the data which assists in reducing the dimension of data. This analysis helps us in detecting the interesting differences between samples along with the information which samples to exclude from downstream analysis. It is also less expensive from computational point of view. Principal component analysis is based on linear algebra concepts. It uses matrix algebra along with some statistical techniques like standardization, variance, covariance, independence, etc.



```
> library(edgeR)
> RLE <- calcNormFactors(DE, method="RLE")
> RLE
An object of class "DGEList"
$counts
  N1 N2 D1 D2
1  5  5 10 10
2  5  5 10 10
3  5  5 10 10
4  5  5 10 10
5  5  5 10 10
25 more rows ...

$samples
      group lib.size norm.factors
N1   normal      150    0.7071068
N2   normal      150    0.7071068
D1 diseased      150    1.4142136
D2 diseased      150    1.4142136

> RLE <- estimateCommonDisp(RLE)
> RLE <- exactTest(RLE)
> table(p.adjust(RLE$table$PValue, method="BH")<0.05)

FALSE  TRUE
   15    15
```

Console 2.3 RLE normalization



```
> library(edgeR)
> UPQ <- calcNormFactors(DE, method="upperquartile")
> UPQ
An object of class "DGEList"
$counts
  N1 N2 D1 D2
1  5  5 10 10
2  5  5 10 10
3  5  5 10 10
4  5  5 10 10
5  5  5 10 10
25 more rows ...

$samples
      group lib.size norm.factors
N1   normal      150    0.7071068
N2   normal      150    0.7071068
D1 diseased      150    1.4142136
D2 diseased      150    1.4142136

> UPQ <- estimateCommonDisp(UPQ)
> UPQ <- exactTest(UPQ)
> table(p.adjust(UPQ$table$PValue, method="BH")<0.05)

FALSE  TRUE
   15    15
```

Console 2.4 Upper-quartile normalization

2.14.1 Steps of PCA Analysis

1. Compute the covariance matrix $Z^T Z$ to see the relation between the variables of a data which is very helpful in taking a decision about a variable.
2. Find eigenvalues and eigenvectors of the covariance matrix. An eigenvector basically shows the direction of data points in a scatterplot and we can choose the most important directions by using such information.
3. Lastly, we make a supposition that more inconsistency in a particular direction correlates with elucidating the behaviour of the dependent variable. High variability generally gives signal, while small variation produces noise. Thus, the more variability there is in a particular direction is, theoretically, revealing of something significant we want to identify.

2.15 Data Analysis of Gene Expression Profiles

Generally, the main objective of the differential expression (DE) study is to create a list of genes passing multiple testing adjustments, ranked by P-values. P-values are adjusted to control the FDR by the well-known procedure Benjamini and Hochberg [29]. The procedure is stated as follows:

Suppose we have $H_1, H_2, \cdots, H_n$ null hypotheses tested and $p_1, p_2, \ldots, p_n$ be their corresponding p-values.

1. Arrange the p-value in increasing order and rename them as $P_1, P_2, \ldots, P_n$.
2. For a given α, determine the largest value of l satisfying the inequality $P_l \leq \frac{l}{n}\alpha$.
3. Reject the null hypothesis H_i for all $i = 1, 2, \ldots, l$.

The differentially expressed genes between two groups can be visualized through a plot as shown in Fig. 2.3 The x-axis describes how much a gene is transcribed in CPM (counts per million) scale, while the y-axis shows the relative difference between two groups for instance the normal and diseased samples.

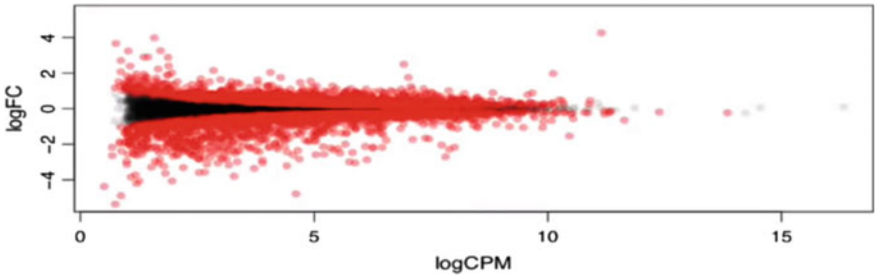


Fig. 2.3 Differentially expressed genes; black dots represent the genes which are not differentially expressed

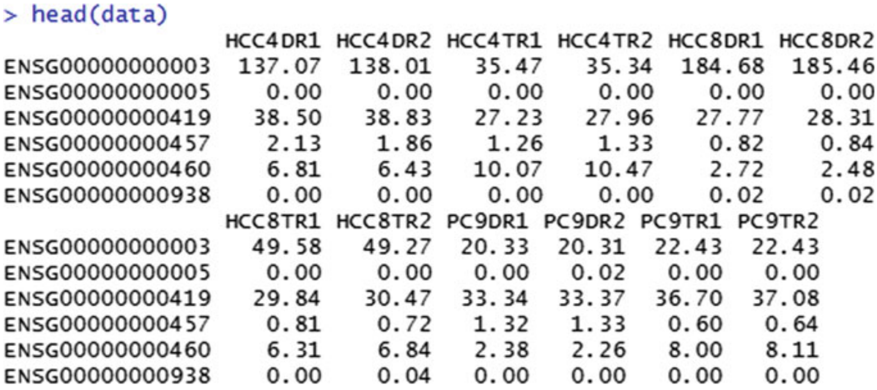
Such analysis helps us to solve two types of problems:

- Validate your hypothesis using experiment.
- Find a certain pathway which is enriched in some particular condition, for instance, normal or diseased.

2.16 An Illustration: A Differential Gene Expression Analysis Conducted on a Real Dataset

To conduct a differential expression analysis on real data set we have used the study by Kurppa et al. [30] to get gene expression profiles, the data is online accessible from GEO database under the accession number GSE131594. This data consists of three cancer cell lines PC-9, HCC827 and HCC4006. Half of the samples of data consists of DMSO treated control cells while the remaining half sample consists of dormant cells. There are two batches in the original dataset, however, we have just considered the first batch in our analysis. Both batches can be considered together by removing batch effects from the replicates. This can be done by adding a batch column in the design matrix.

The dimension of data is 6 57905 × 12, which contains 6 57905 genes and 12 samples. We have 6 groups of experiments corresponding to 3 cell lines with 2 replicates. This grouping is based on the biological conditions (DMSO vs. DORMANT). We have modified the names of samples for simplicity. The head of the data is in Console 2.5.



```
> head(data)
                HCC4DR1 HCC4DR2 HCC4TR1 HCC4TR2 HCC8DR1 HCC8DR2
ENSG00000000003  137.07  138.01   35.47   35.34  184.68  185.46
ENSG00000000005    0.00    0.00    0.00    0.00    0.00    0.00
ENSG00000000419   38.50   38.83   27.23   27.96   27.77   28.31
ENSG00000000457    2.13    1.86    1.26    1.33    0.82    0.84
ENSG00000000460    6.81    6.43   10.07   10.47    2.72    2.48
ENSG00000000938    0.00    0.00    0.00    0.00    0.02    0.02
                HCC8TR1 HCC8TR2 PC9DR1 PC9DR2 PC9TR1 PC9TR2
ENSG00000000003   49.58   49.27  20.33  20.31  22.43  22.43
ENSG00000000005    0.00    0.00   0.00   0.02   0.00   0.00
ENSG00000000419   29.84   30.47  33.34  33.37  36.70  37.08
ENSG00000000457    0.81    0.72   1.32   1.33   0.60   0.64
ENSG00000000460    6.31    6.84   2.38   2.26   8.00   8.11
ENSG00000000938    0.00    0.04   0.00   0.00   0.00   0.00
```

Console 2.5 Head (data)

Table 2.3 Sample information

Sample name	Cell type	Status
HCC4DR1	HCC4006	DMSO
HCC4DR2	HCC4006	DMSO
HCC4TR1	HCC4006	DORMANT
HCC4TR2	HCC4006	DORMANT
HCC8DR1	HCC827	DMSO
HCC8DR2	HCC827	DMSO
HCC8TR1	HCC827	DORMANT
HCC8TR2	HCC827	DORMANT
PCDR1	PC-9	DMSO
PCDR2	PC-9	DMSO
PCTR1	PC-9	DORMANT
PCTR2	PC-9	DORMANT

Detailed information about samples is given in Table 2.3.

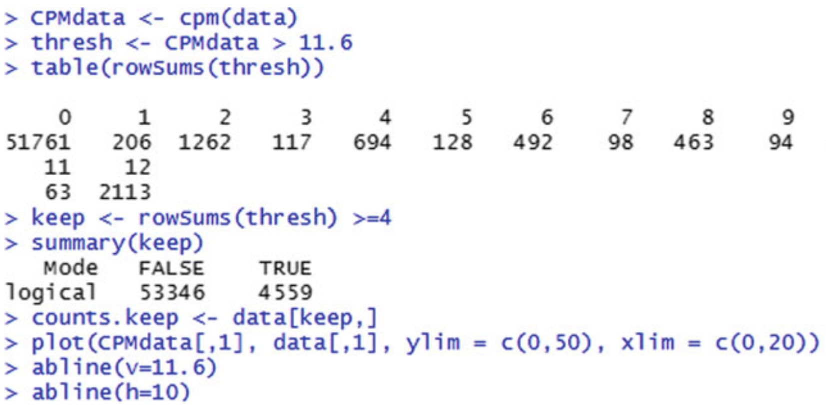
2.17 R Packages Used in the RNA-Seq Analysis

R/R-Studio packages used in the subsequent differential expressional analysis of RNA-seq data are RCurl, limma, edgeR, Glimma, RColorBrewer, gplots, statmod, org.Hs.eg.db.

Most of the packages used in the analysis are Bioconductor tools and can be installed from Bioconductor website https://www.bioconductor.org/packages/release/bioc/. Bioconductor tools have made the analysis of high throughput sequencing very auspicious for the researchers. The main programming language used in Bioconductor is R.

2.18 Removal of Lowly Transcribed Genes

Genes with low expression values are usually not very informative in differential expression study so removal of those genes is worthful as this step reduces the computational burden of downstream analysis. For this purpose, a threshold is chosen detecting the CPM that relates to a count of 10 which in our case approximates the value 11.3 as shown in Fig. 2.4. In our analysis, a gene retains if it appears in at least two groups, however, this restriction might be reduced to one group as well by selecting value 2 instead of 4. Firstly, CPM values for the entire gene counts are computed by using command cpm() as depicted in Console 2.6 then we put the threshold and obtained the required genes, only 4559 genes fulfilled our criteria.



```
> CPMdata <- cpm(data)
> thresh <- CPMdata > 11.6
> table(rowSums(thresh))

    0     1     2     3     4     5     6     7     8     9    10 
51761   206  1262   117   694   128   492    98   463    94   414 
   11    12 
   63  2113 
> keep <- rowSums(thresh) >=4
> summary(keep)
   Mode   FALSE    TRUE 
logical   53346    4559 
> counts.keep <- data[keep,]
> plot(CPMdata[,1], data[,1], ylim = c(0,50), xlim = c(0,20))
> abline(v=11.6)
> abline(h=10)
```

Console 2.6 Gene counts thresholding

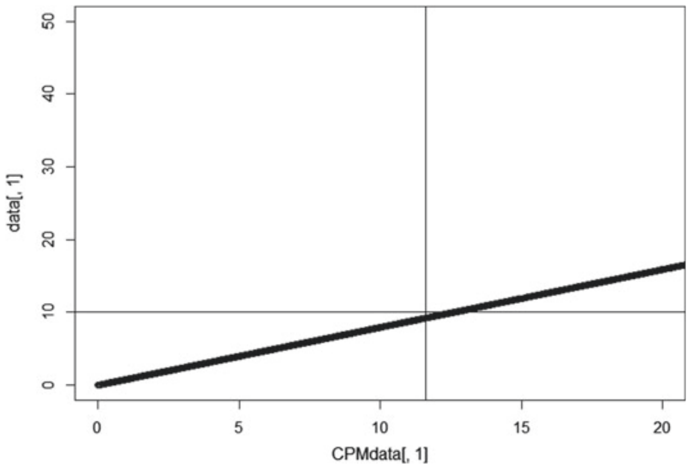
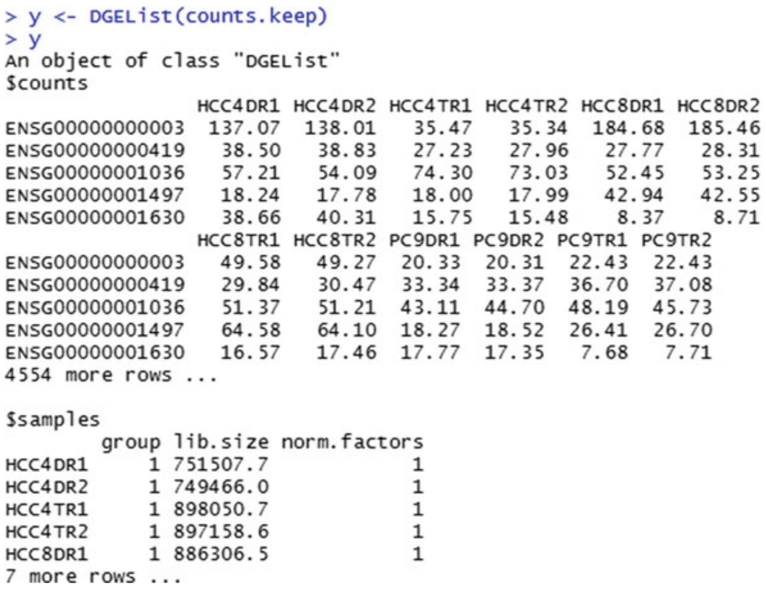


Fig. 2.4 Selection of threshold based on CPM and expression data

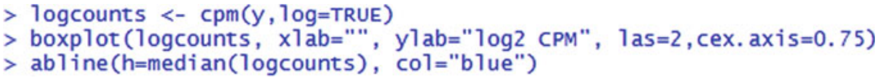
2.19 Formation of DGEList Object Using EdgeR

An object of DGE list in Console 2.7 is created that is a main data class in edgeR and consists of different slots to hold different types of information about the data.



```
> y <- DGEList(counts.keep)
> y
An object of class "DGEList"
$counts
                HCC4DR1 HCC4DR2 HCC4TR1 HCC4TR2 HCC8DR1 HCC8DR2
ENSG00000000003  137.07  138.01   35.47   35.34  184.68  185.46
ENSG00000000419   38.50   38.83   27.23   27.96   27.77   28.31
ENSG00000001036   57.21   54.09   74.30   73.03   52.45   53.25
ENSG00000001497   18.24   17.78   18.00   17.99   42.94   42.55
ENSG00000001630   38.66   40.31   15.75   15.48    8.37    8.71
                HCC8TR1 HCC8TR2 PC9DR1 PC9DR2 PC9TR1 PC9TR2
ENSG00000000003   49.58   49.27  20.33  20.31  22.43  22.43
ENSG00000000419   29.84   30.47  33.34  33.37  36.70  37.08
ENSG00000001036   51.37   51.21  43.11  44.70  48.19  45.73
ENSG00000001497   64.58   64.10  18.27  18.52  26.41  26.70
ENSG00000001630   16.57   17.46  17.77  17.35   7.68   7.71
4554 more rows ...

$samples
        group lib.size norm.factors
HCC4DR1     1 751507.7            1
HCC4DR2     1 749466.0            1
HCC4TR1     1 898050.7            1
HCC4TR2     1 897158.6            1
HCC8DR1     1 886306.5            1
7 more rows ...
```

Console 2.7 DGEList object



```
> logcounts <- cpm(y,log=TRUE)
> boxplot(logcounts, xlab="", ylab="log2 CPM", las=2,cex.axis=0.75)
> abline(h=median(logcounts), col="blue")
```

Console 2.8 Samples density distribution in logCPM

2.20 Density Distributions

To observe the density distribution of gene counts, we have measured log intensities using cpm function for raw counts Console 2.8.

Box plots in Fig. 2.5 illustrate an analogous behaviour of density distributions across all samples so we don't need to examine any sample further.

2.21 Normalization

Normalization factors have been calculated using the method TMM. By default, edgeR uses the method TMM so we don't need to specify the method in the command $y < -\text{calcNormFactors}(y)$. Table 2.4 shows the norm.factors corresponding to each library.

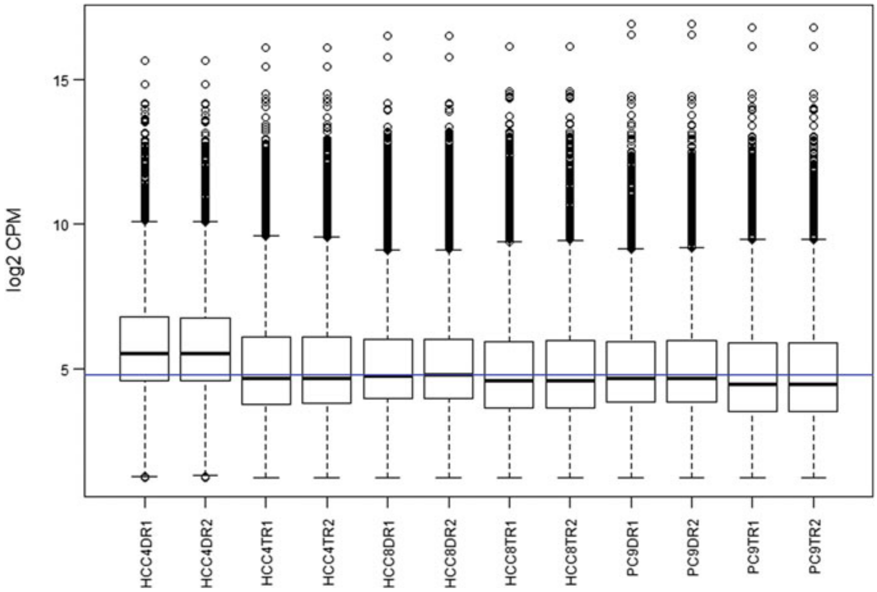


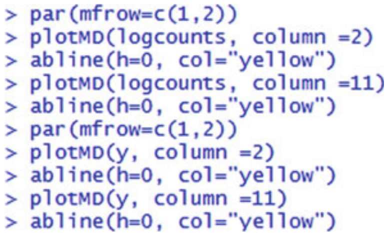
Fig. 2.5 log2 (CPM)

Table 2.4 Norm.factors

Sample	Lib. size	Norm. factors
HCC4DR1	751,507.7	1.5781530
HCC4DR2	749,466	1.5850865
HCC4TR1	898,050.7	0.9691410
HCC4TR2	897,158.6	0.9709712
HCC8DR1	886,306.5	0.9897554
HCC8DR2	886,316.2	0.9963440
HCC8TR1	957,065.4	0.8959495
HCC8TR2	955,566.4	0.9030861
PC9DR1	796,577.1	0.9024373
PC9DR2	1 795,879.1	0.9044958
PC9TR1	868,377.3	0.8055157
PC9TR2	866,746.3	0.8097616

In Table 2.4, the value lower than unity indicates that the library size must be decreased while a value above one depicts intensification in the library size. The performance of the normalization method might be checked for each sample through mean difference (MD) plotting. MD plot illustrates the log ratio of two libraries versus mean log expression of those libraries. In Table 2.4, the minimum value occurs corresponding to the sample PC9TR1, whereas the maximum value corresponds to

Console 2.9 MD plots



```
> par(mfrow=c(1,2))
> plotMD(logcounts, column =2)
> abline(h=0, col="yellow")
> plotMD(logcounts, column =11)
> abline(h=0, col="yellow")
> par(mfrow=c(1,2))
> plotMD(y, column =2)
> abline(h=0, col="yellow")
> plotMD(y, column =11)
> abline(h=0, col="yellow")
```

HCC4DR2 so MD plots for both samples before and after TMM normalization are shown in Figs. 2.6 and 2.7, respectively. Code for both figures is displayed in Console 2.9.

All libraries excluding the one under consideration are considered in the reference library. A single library might be chosen as reference as well for instance the sample with median library size. We can easily see the difference between the libraries as in Fig. 2.7 most of the genes are centred at log ratio of 0 which is not the case in the MDS plots in Fig. 2.6.

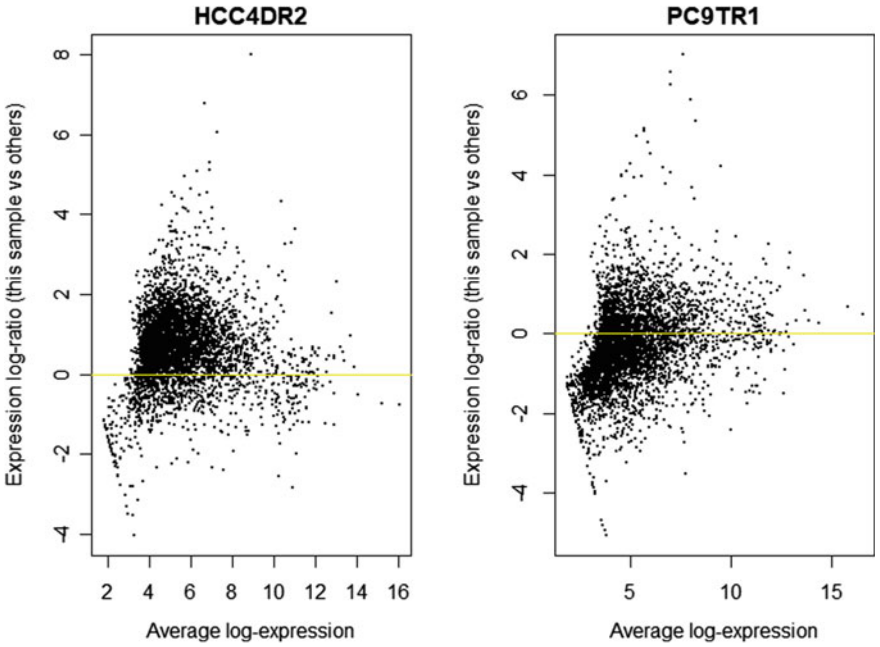


Fig. 2.6 MD plots for samples HCC4DR2 and PC9TR1 before normalization

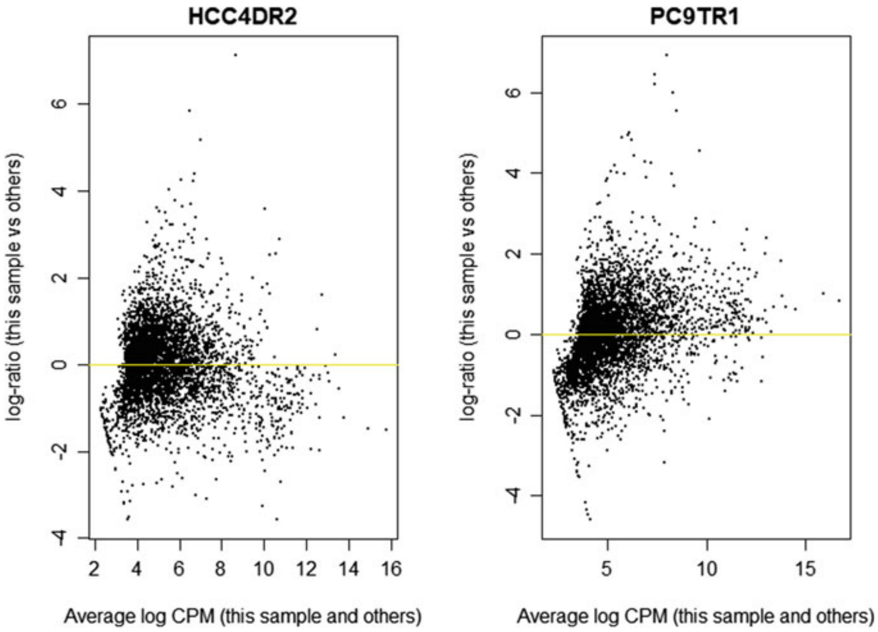


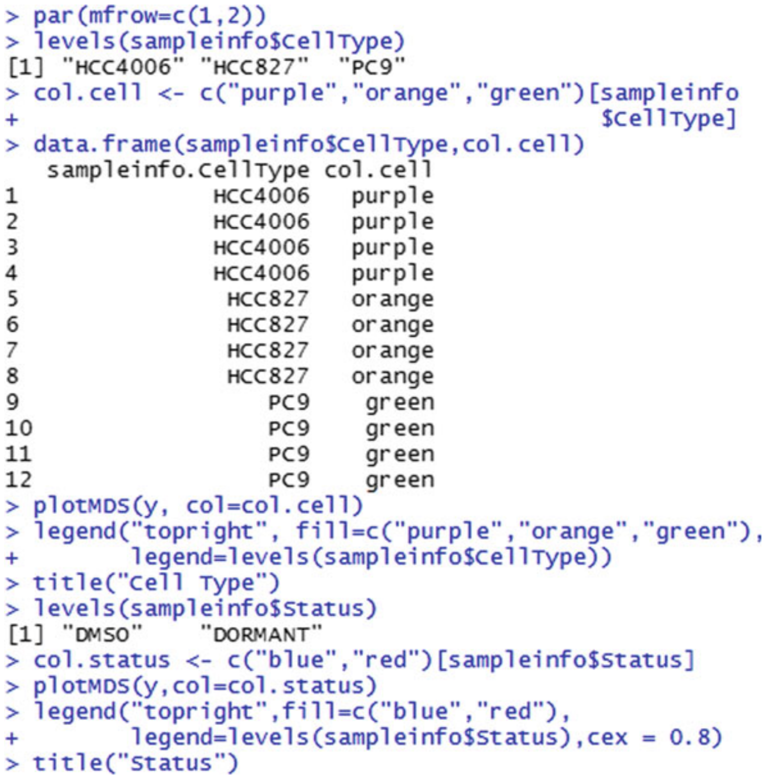
Fig. 2.7 MD plots for samples HCC4DR2 and PC9TR1 after normalization

2.22 Principal Component Analysis

MDS plots are used to visualize the principal components of the data. It assists to recognize the most important directions of the data and the direction of most important component lies along x-axis. This 2D plotting shows the distance between the samples. To find the distance between two samples is calculated by picking the highest 500 $\log_2$ FC genes between both samples and then evaluating their root mean square. Two MDS plots, code in Console 2.10, are shown in Fig. 2.8 with two distinct clutch information one with cell categories and the other with DMSO induced and DORMANT state levels. The largest distance along leading log FC dim 1 is appearing between HCC4D and PC9D samples.

2.23 Design Matrix

Next we formulated a design matrix shown in Console 2.11. Each column of the design matrix corresponds to one of six groups considered in the analysis. All the samples contained in the same group have been assigned the value 1. This setting is known as group mean parametrization.



```
> par(mfrow=c(1,2))
> levels(sampleinfo$CellType)
[1] "HCC4006" "HCC827"  "PC9"
> col.cell <- c("purple","orange","green")[sampleinfo
+                                              $CellType]
> data.frame(sampleinfo$CellType,col.cell)
   sampleinfo.CellType col.cell
1              HCC4006   purple
2              HCC4006   purple
3              HCC4006   purple
4              HCC4006   purple
5               HCC827   orange
6               HCC827   orange
7               HCC827   orange
8               HCC827   orange
9                  PC9    green
10                 PC9    green
11                 PC9    green
12                 PC9    green
> plotMDS(y, col=col.cell)
> legend("topright", fill=c("purple","orange","green"),
+        legend=levels(sampleinfo$CellType))
> title("Cell Type")
> levels(sampleinfo$Status)
[1] "DMSO"    "DORMANT"
> col.status <- c("blue","red")[sampleinfo$Status]
> plotMDS(y,col=col.status)
> legend("topright",fill=c("blue","red"),
+        legend=levels(sampleinfo$Status),cex = 0.8)
> title("Status")
```

Console 2.10 Principal components of data based on cell type and status

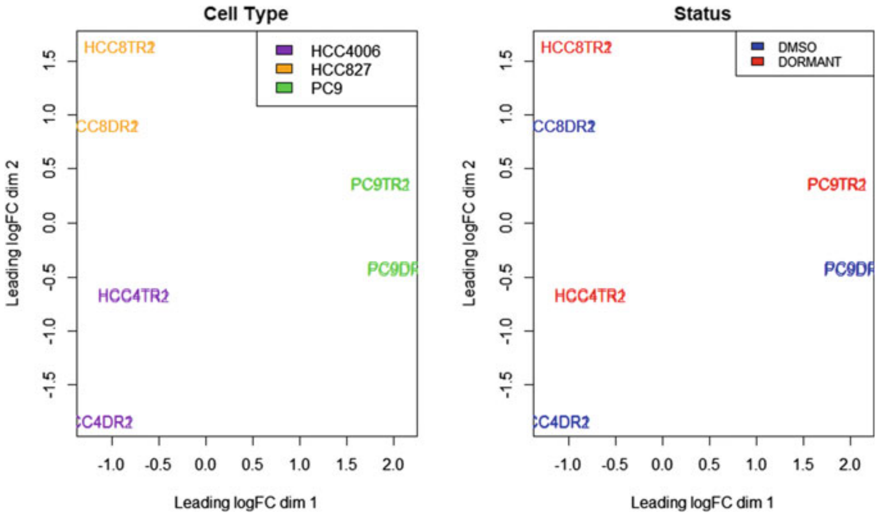
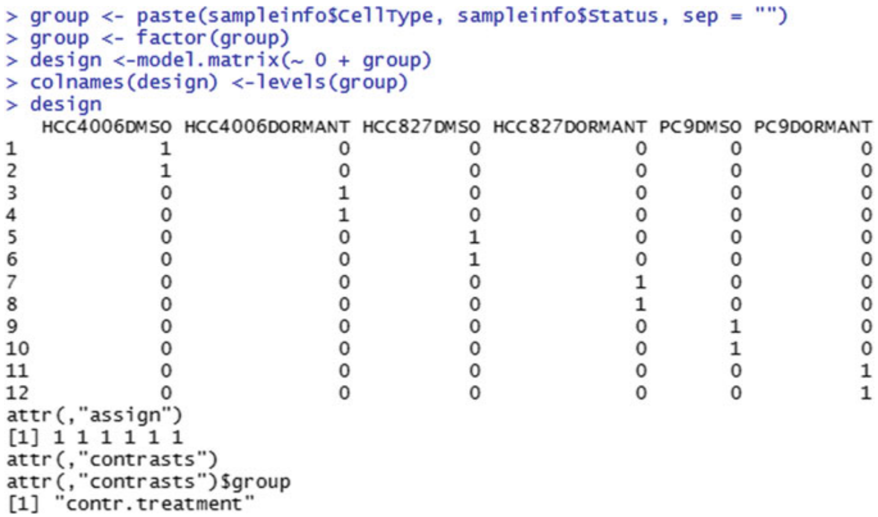


Fig. 2.8 MDS plots showing distances between samples based on cell types and status

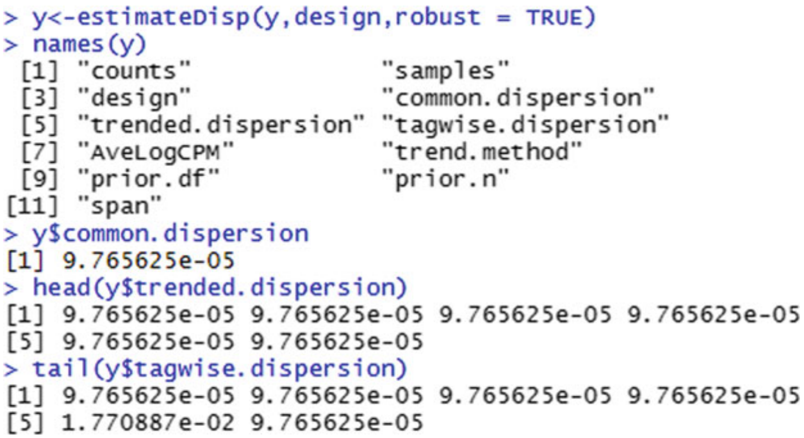


```
> group <- paste(sampleinfo$CellType, sampleinfo$Status, sep = "")
> group <- factor(group)
> design <-model.matrix(~ 0 + group)
> colnames(design) <-levels(group)
> design
   HCC4006DMSO HCC4006DORMANT HCC827DMSO HCC827DORMANT PC9DMSO PC9DORMANT
1            1              0          0             0       0          0
2            1              0          0             0       0          0
3            0              1          0             0       0          0
4            0              1          0             0       0          0
5            0              0          1             0       0          0
6            0              0          1             0       0          0
7            0              0          0             1       0          0
8            0              0          0             1       0          0
9            0              0          0             0       1          0
10           0              0          0             0       1          0
11           0              0          0             0       0          1
12           0              0          0             0       0          1
attr(,"assign")
[1] 1 1 1 1 1 1
attr(,"contrasts")
attr(,"contrasts")$group
[1] "contr.treatment"
```

Console 2.11 Design matrix

2.24 NB and QL Dispersion Evaluation

NB dispersion estimate which is also the square of coefficient of BV is evaluated using the function estimateDisp() while glmQLFit() command is used to find QL dispersion estimates. The function estimateDisp() creates an object containing estimates for three types of dispersions namely common, trended and tagwise (Console 2.12).



```
> y<-estimateDisp(y,design,robust = TRUE)
> names(y)
 [1] "counts"             "samples"
 [3] "design"             "common.dispersion"
 [5] "trended.dispersion" "tagwise.dispersion"
 [7] "AveLogCPM"          "trend.method"
 [9] "prior.df"           "prior.n"
[11] "span"
> y$common.dispersion
[1] 9.765625e-05
> head(y$trended.dispersion)
[1] 9.765625e-05 9.765625e-05 9.765625e-05 9.765625e-05
[5] 9.765625e-05 9.765625e-05
> tail(y$tagwise.dispersion)
[1] 9.765625e-05 9.765625e-05 9.765625e-05 9.765625e-05
[5] 1.770887e-02 9.765625e-05
```

Console 2.12 Dispersion estimates

The command glmQLFit() provides a DGEGLM object (Console 2.13) containing all the gene-associated GLM coefficients. The trend in NB dispersions varies inversely with gene expressions since the highly expressed genes are more stabilized comparatively to lowly expressed genes. This fact might be utilized to recognize batch effects or trended biases in highly expressed gene profiles. The function glmQLFit() provides a vector comprising of df.prior values which provides the information about variation in QL dispersions between genes. The higher values for df.prior indicates stronger EB moderation while small values show the reverse.

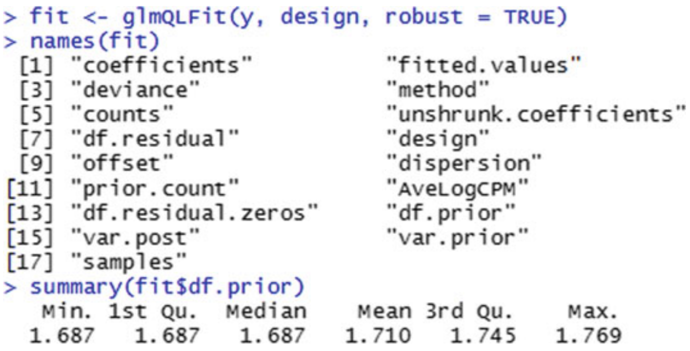
We can also visualize DGEGLM object by using command $plotQLDisp()$*as shown in* Fig. 2.9.

2.25 Annotating Genes

We can add some annotations in our downstream analysis by using the package org.Hs.eg.db, for a mouse data set the available package is org.Mm.eg.db. We can select from available options the annotation of our choice. In our analysis, we have used ENTREZID and SYMBOL as both are very important from gene enrichment analysis point of view. Console 2.14 shows how to add gene annotation information, since our data is using ENSEMBL gene ids so we have used keytype = “ENSEMBL″.

2.26 Gene Testing

Now, our subsequent task is to find a list of differentially expressed genes between two groups of our interest. Such comparisons can be defined using the makeContrasts() command. We will specify the comparisons between DMSO and DORMANT states



```
> fit <- glmQLFit(y, design, robust = TRUE)
> names(fit)
 [1] "coefficients"          "fitted.values"
 [3] "deviance"              "method"
 [5] "counts"                "unshrunk.coefficients"
 [7] "df.residual"           "design"
 [9] "offset"                "dispersion"
[11] "prior.count"           "AveLogCPM"
[13] "df.residual.zeros"     "df.prior"
[15] "var.post"              "var.prior"
[17] "samples"
> summary(fit$df.prior)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.687   1.687   1.687   1.710   1.745   1.769
```

Console 2.13 DGEGLM object

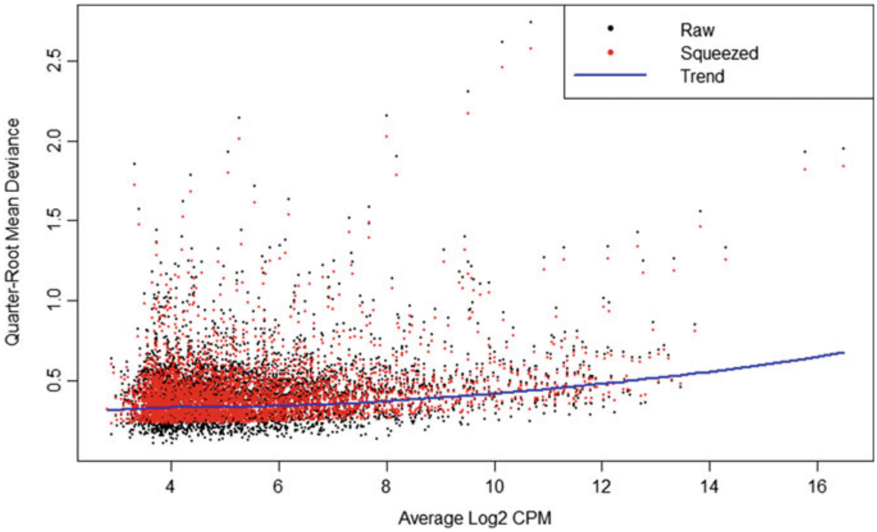


Fig. 2.9 Quarter root QL dispersion against average abundance of log2 CPM

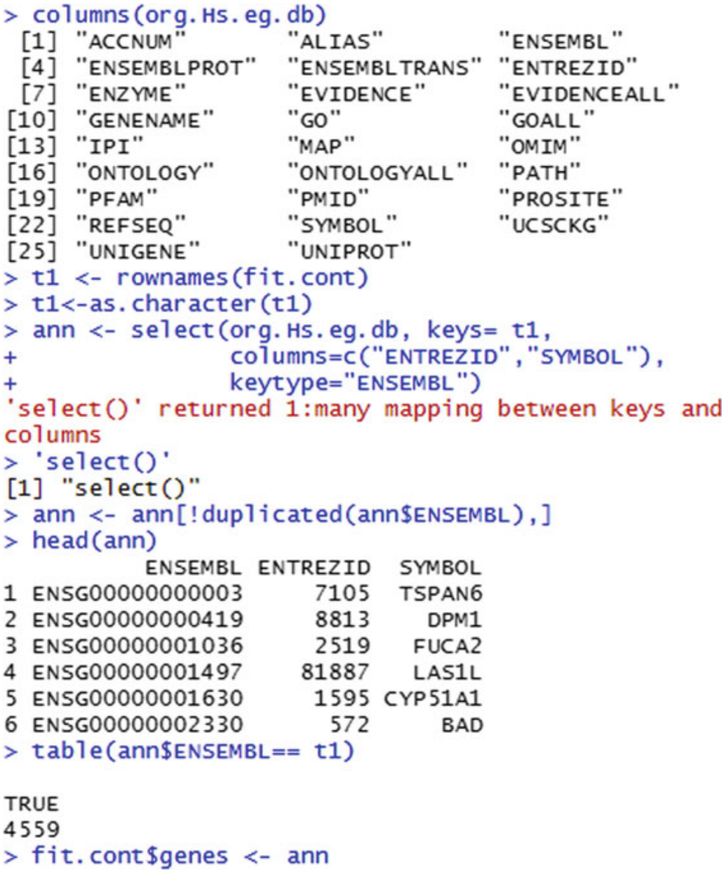
for each cell line. This chapter contains the gene testing for the cell line HCC4006 only, analyses for the other two cell lines might be conducted on the same lines. We have used the glmQLFTest to check the significant difference in each gene expression profile. The contrast is defined between the conditions DMSO and DORMANT. We can get the uppermost significant genes with respect to measure of logFC or PValue by using the command topTags. The total number of genes found to be differentially expressed in the analysis is equal to 2230, among which 1145 are down-regulated and 1085 are up-regulated. The related code is provided in Console 2.15.

Let's visualize the differentially expressed genes using MD plots with log FC vs average abundance for each gene. In Fig. 2.10**,** significantly expressed genes are shown in green and red colours where green dots represent up-regulated genes and red dots represent down-regulated. Code for the plot is given in Console 2.16.

We can have a look at the top genes ranked according to P-values by using the command topTags. We can also pick a gene we are interested in and visualize it sample-wise using stripchart command, for instance, pick CEACAM6 from Console 2.17. Code for the stripchart plot is given in Console 2.16. We can see in Fig. 2.11**,** the normalized log expression value for CEACAM6 is very high in the second sample, i.e. HCC4006DORMANT.

2.27 GO Analysis

There are several gene sets testing approaches which might be used to deduce the biological interpretation of the DE analysis. One of them is gene ontology enrichment

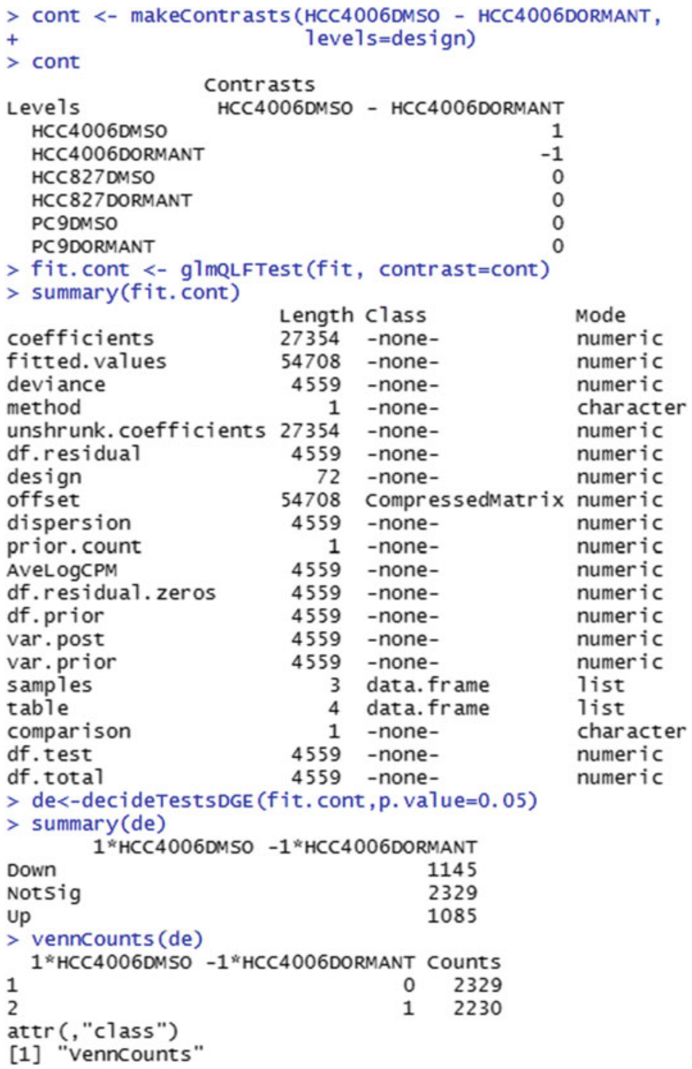


```
> columns(org.Hs.eg.db)
 [1] "ACCNUM"        "ALIAS"         "ENSEMBL"
 [4] "ENSEMBLPROT"   "ENSEMBLTRANS"  "ENTREZID"
 [7] "ENZYME"        "EVIDENCE"      "EVIDENCEALL"
[10] "GENENAME"      "GO"            "GOALL"
[13] "IPI"           "MAP"           "OMIM"
[16] "ONTOLOGY"      "ONTOLOGYALL"   "PATH"
[19] "PFAM"          "PMID"          "PROSITE"
[22] "REFSEQ"        "SYMBOL"        "UCSCKG"
[25] "UNIGENE"       "UNIPROT"
> t1 <- rownames(fit.cont)
> t1<-as.character(t1)
> ann <- select(org.Hs.eg.db, keys= t1,
+               columns=c("ENTREZID","SYMBOL"),
+               keytype="ENSEMBL")
'select()' returned 1:many mapping between keys and
columns
> 'select()'
[1] "select()"
> ann <- ann[!duplicated(ann$ENSEMBL),]
> head(ann)
          ENSEMBL ENTREZID  SYMBOL
1 ENSG00000000003     7105  TSPAN6
2 ENSG00000000419     8813    DPM1
3 ENSG00000001036     2519   FUCA2
4 ENSG00000001497    81887   LAS1L
5 ENSG00000001630     1595 CYP51A1
6 ENSG00000002330      572     BAD
> table(ann$ENSEMBL== t1)

TRUE
4559
> fit.cont$genes <- ann
```

Console 2.14 Gene annotations

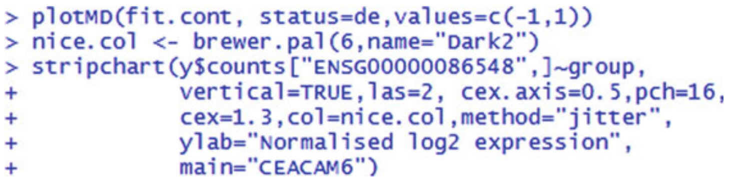
analysis also known as GO enrichment analysis. This analysis tells us which gene sets or pathways are up signified and which are down in a DE analysis. GO enrichment analysis can be conducted by using goana() function in R. We have to define a species for the analysis, for instance, in human data sets we use Hs while in mouse data we can use Mm. There are three classes of ontology used in GO analysis namely biological process, molecular function and cellular component abbreviated as BP, MF and CC, respectively. In Console 2.18 if we look at the first GO term that is GO: 0,003,735 then it is shown that in the current DE analysis the GO term structural constituent of ribosome, a molecular function, consists of 147 annotated genes among which 114 are significantly differentially expressed subject to our defined contrast.

We can also use gene lengths in gene enrichment analysis. There is an R package ‘EDASeq’ which has a built-in function specifically made to find gene lengths, i.e. getGeneLengthAndGCContent(). More details about the package ‘EDASeq’ can be found at the website https://homolog.us/Bioconductor/EDASeq.html. We assume



```
> cont <- makeContrasts(HCC4006DMSO - HCC4006DORMANT,
+                       levels=design)
> cont
                Contrasts
Levels           HCC4006DMSO - HCC4006DORMANT
  HCC4006DMSO                               1
  HCC4006DORMANT                           -1
  HCC827DMSO                                0
  HCC827DORMANT                             0
  PC9DMSO                                   0
  PC9DORMANT                                0
> fit.cont <- glmQLFTest(fit, contrast=cont)
> summary(fit.cont)
                      Length Class            Mode
coefficients          27354  -none-           numeric
fitted.values         54708  -none-           numeric
deviance               4559  -none-           numeric
method                    1  -none-           character
unshrunk.coefficients 27354  -none-           numeric
df.residual            4559  -none-           numeric
design                   72  -none-           numeric
offset                54708  CompressedMatrix numeric
dispersion             4559  -none-           numeric
prior.count               1  -none-           numeric
AveLogCPM              4559  -none-           numeric
df.residual.zeros      4559  -none-           numeric
df.prior               4559  -none-           numeric
var.post               4559  -none-           numeric
var.prior              4559  -none-           numeric
samples                   3  data.frame       list
table                     4  data.frame       list
comparison                1  -none-           character
df.test                4559  -none-           numeric
df.total               4559  -none-           numeric
> de<-decideTestsDGE(fit.cont,p.value=0.05)
> summary(de)
       1*HCC4006DMSO -1*HCC4006DORMANT
Down                              1145
NotSig                            2329
Up                                1085
> vennCounts(de)
  1*HCC4006DMSO -1*HCC4006DORMANT Counts
1                               0   2329
2                               1   2230
attr(,"class")
[1] "VennCounts"
```

Console 2.15 Contrasts, glmQLFTest and DGE list



```
> plotMD(fit.cont, status=de,values=c(-1,1))
> nice.col <- brewer.pal(6,name="Dark2")
> stripchart(y$counts["ENSG00000086548",]~group,
+            vertical=TRUE,las=2, cex.axis=0.5,pch=16,
+            cex=1.3,col=nice.col,method="jitter",
+            ylab="Normalised log2 expression",
+            main="CEACAM6")
```

Console 2.16 MD plot for differentially expressed genes and strip chart code

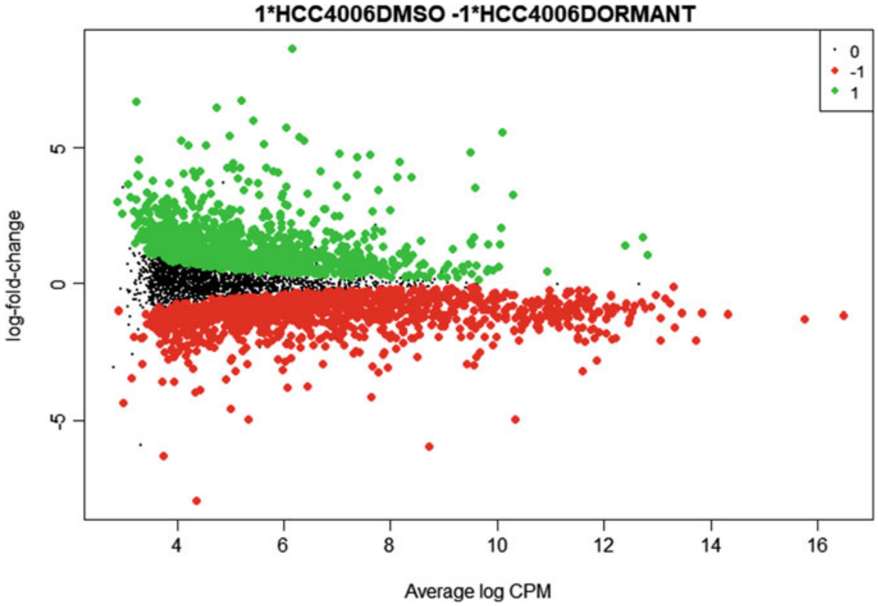
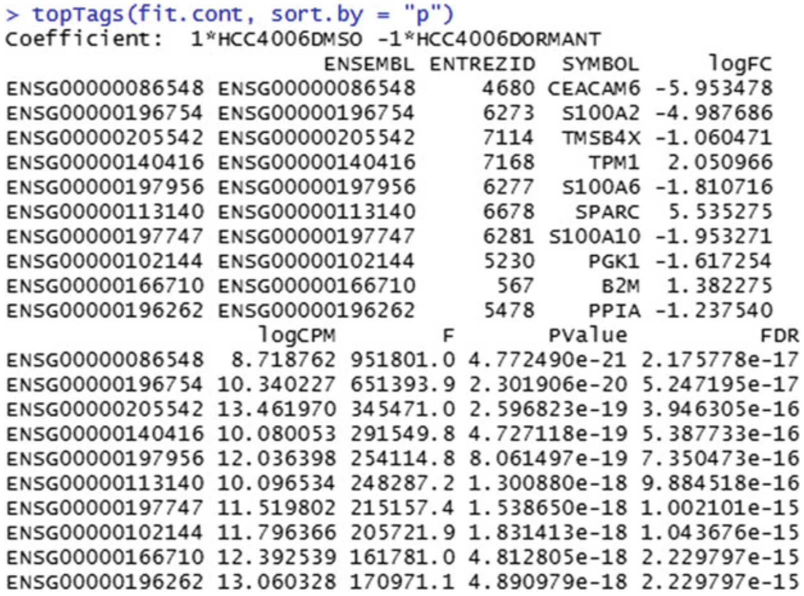


Fig. 2.10 Visualizing differentially expressed genes using MD plots



```
> topTags(fit.cont, sort.by = "p")
Coefficient:  1*HCC4006DMSO -1*HCC4006DORMANT
                        ENSEMBL ENTREZID  SYMBOL     logFC
ENSG00000086548 ENSG00000086548     4680 CEACAM6 -5.953478
ENSG00000196754 ENSG00000196754     6273  S100A2 -4.987686
ENSG00000205542 ENSG00000205542     7114  TMSB4X -1.060471
ENSG00000140416 ENSG00000140416     7168    TPM1  2.050966
ENSG00000197956 ENSG00000197956     6277  S100A6 -1.810716
ENSG00000113140 ENSG00000113140     6678   SPARC  5.535275
ENSG00000197747 ENSG00000197747     6281 S100A10 -1.953271
ENSG00000102144 ENSG00000102144     5230    PGK1 -1.617254
ENSG00000166710 ENSG00000166710      567     B2M  1.382275
ENSG00000196262 ENSG00000196262     5478    PPIA -1.237540
                   logCPM        F       PValue          FDR
ENSG00000086548  8.718762 951801.0 4.772490e-21 2.175778e-17
ENSG00000196754 10.340227 651393.9 2.301906e-20 5.247195e-17
ENSG00000205542 13.461970 345471.0 2.596823e-19 3.946305e-16
ENSG00000140416 10.080053 291549.8 4.727118e-19 5.387733e-16
ENSG00000197956 12.036398 254114.8 8.061497e-19 7.350473e-16
ENSG00000113140 10.096534 248287.2 1.300880e-18 9.884518e-16
ENSG00000197747 11.519802 215157.4 1.538650e-18 1.002101e-15
ENSG00000102144 11.796366 205721.9 1.831413e-18 1.043676e-15
ENSG00000166710 12.392539 161781.0 4.812805e-18 2.229797e-15
ENSG00000196262 13.060328 170971.1 4.890979e-18 2.229797e-15
```

Console 2.17 Top differentially expressed genes

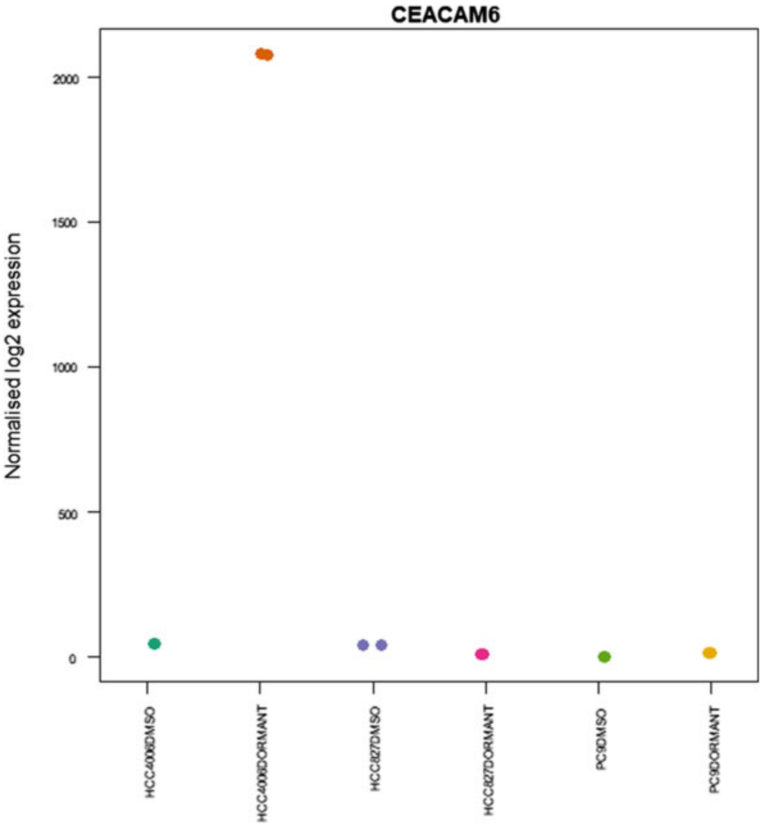
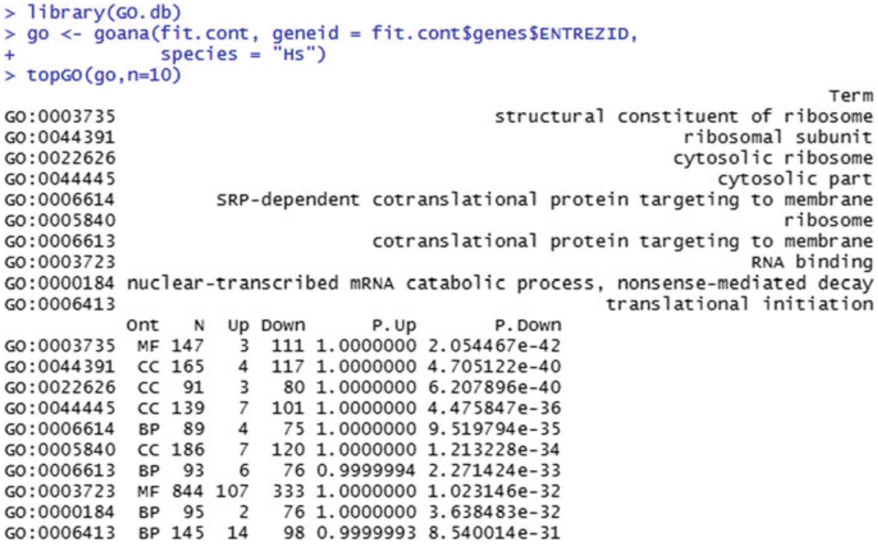


Fig. 2.11 Normalized expression of CEACAM6 across all samples

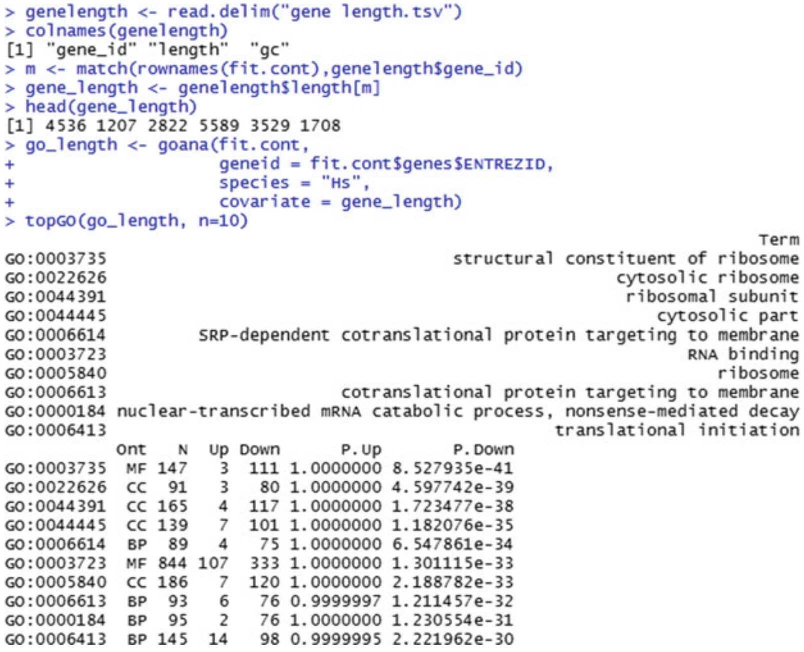
that the gene lengths have already been found before the analysis of raw data. This data is stored in the file "gene length.tsv" which has 3 columnsgene_id, length andgc. Since this data include all the genes initially involved in the analysis so the list of gene_id is further matched with the rownames of fit.cont object to extract the desired genes. Then, the GO enrichment analysis is conducted incorporating the information about gene lengths. We can easily see few differences in P.Up and P.Down columns in Consoles 2.18 and 2.19 which has also changed the ordering of associated genes.

If we are concerned with a specific domain for ontology like MF, CC, and BP, then we can explicitly define it in topGo() to extract the relevant information only.



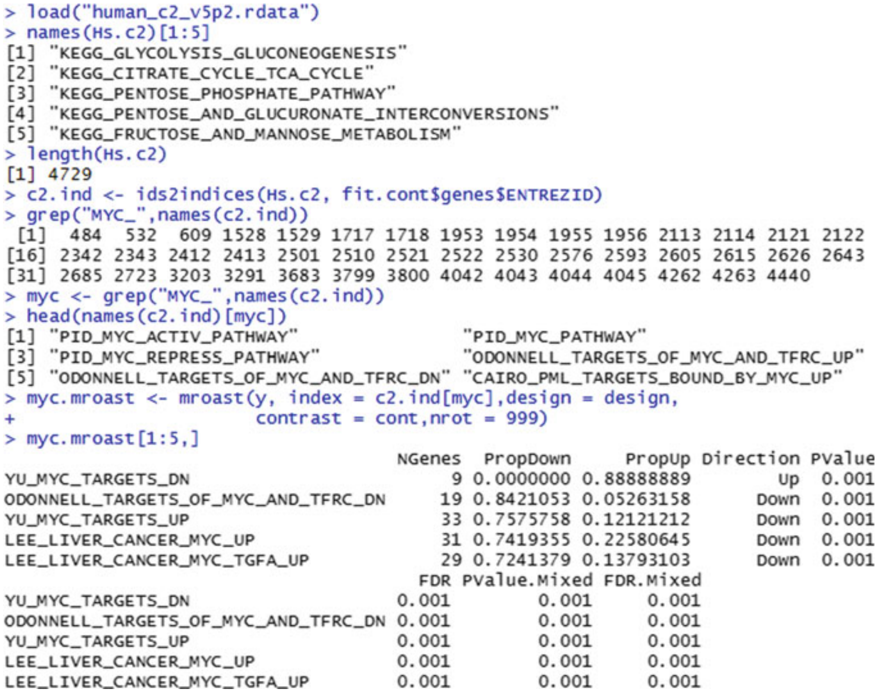
```
> library(GO.db)
> go <- goana(fit.cont, geneid = fit.cont$genes$ENTREZID,
+             species = "Hs")
> topGO(go,n=10)
                                                                        Term
GO:0003735                                 structural constituent of ribosome
GO:0044391                                                   ribosomal subunit
GO:0022626                                                  cytosolic ribosome
GO:0044445                                                      cytosolic part
GO:0006614          SRP-dependent cotranslational protein targeting to membrane
GO:0005840                                                            ribosome
GO:0006613                        cotranslational protein targeting to membrane
GO:0003723                                                         RNA binding
GO:0000184 nuclear-transcribed mRNA catabolic process, nonsense-mediated decay
GO:0006413                                           translational initiation
           Ont   N  Up Down      P.Up       P.Down
GO:0003735  MF 147   3  111 1.0000000 2.054467e-42
GO:0044391  CC 165   4  117 1.0000000 4.705122e-40
GO:0022626  CC  91   3   80 1.0000000 6.207896e-40
GO:0044445  CC 139   7  101 1.0000000 4.475847e-36
GO:0006614  BP  89   4   75 1.0000000 9.519794e-35
GO:0005840  CC 186   7  120 1.0000000 1.213228e-34
GO:0006613  BP  93   6   76 0.9999994 2.271424e-33
GO:0003723  MF 844 107  333 1.0000000 1.023146e-32
GO:0000184  BP  95   2   76 1.0000000 3.638483e-32
GO:0006413  BP 145  14   98 0.9999993 8.540014e-31
```

Console 2.18 GO enrichment analysis



```
> genelength <- read.delim("gene length.tsv")
> colnames(genelength)
[1] "gene_id" "length"  "gc"
> m <- match(rownames(fit.cont),genelength$gene_id)
> gene_length <- genelength$length[m]
> head(gene_length)
[1] 4536 1207 2822 5589 3529 1708
> go_length <- goana(fit.cont,
+                    geneid = fit.cont$genes$ENTREZID,
+                    species = "Hs",
+                    covariate = gene_length)
> topGO(go_length, n=10)
                                                                        Term
GO:0003735                                 structural constituent of ribosome
GO:0022626                                                  cytosolic ribosome
GO:0044391                                                   ribosomal subunit
GO:0044445                                                      cytosolic part
GO:0006614          SRP-dependent cotranslational protein targeting to membrane
GO:0003723                                                         RNA binding
GO:0005840                                                            ribosome
GO:0006613                        cotranslational protein targeting to membrane
GO:0000184 nuclear-transcribed mRNA catabolic process, nonsense-mediated decay
GO:0006413                                           translational initiation
           Ont   N  Up Down      P.Up       P.Down
GO:0003735  MF 147   3  111 1.0000000 8.527935e-41
GO:0022626  CC  91   3   80 1.0000000 4.597742e-39
GO:0044391  CC 165   4  117 1.0000000 1.723477e-38
GO:0044445  CC 139   7  101 1.0000000 1.182076e-35
GO:0006614  BP  89   4   75 1.0000000 6.547861e-34
GO:0003723  MF 844 107  333 1.0000000 1.301115e-33
GO:0005840  CC 186   7  120 1.0000000 2.188782e-33
GO:0006613  BP  93   6   76 0.9999997 1.211457e-32
GO:0000184  BP  95   2   76 1.0000000 1.230554e-31
GO:0006413  BP 145  14   98 0.9999995 2.221962e-30
```

Console 2.19 Go analysis including gene length information



```
> load("human_c2_v5p2.rdata")
> names(Hs.c2)[1:5]
[1] "KEGG_GLYCOLYSIS_GLUCONEOGENESIS"
[2] "KEGG_CITRATE_CYCLE_TCA_CYCLE"
[3] "KEGG_PENTOSE_PHOSPHATE_PATHWAY"
[4] "KEGG_PENTOSE_AND_GLUCURONATE_INTERCONVERSIONS"
[5] "KEGG_FRUCTOSE_AND_MANNOSE_METABOLISM"
> length(Hs.c2)
[1] 4729
> c2.ind <- ids2indices(Hs.c2, fit.cont$genes$ENTREZID)
> grep("MYC_",names(c2.ind))
 [1]  484  532  609 1528 1529 1717 1718 1953 1954 1955 1956 2113 2114 2121 2122
[16] 2342 2343 2412 2413 2501 2510 2521 2522 2530 2576 2593 2605 2615 2626 2643
[31] 2685 2723 3203 3291 3683 3799 3800 4042 4043 4044 4045 4262 4263 4440
> myc <- grep("MYC_",names(c2.ind))
> head(names(c2.ind)[myc])
[1] "PID_MYC_ACTIV_PATHWAY"                "PID_MYC_PATHWAY"
[3] "PID_MYC_REPRESS_PATHWAY"              "ODONNELL_TARGETS_OF_MYC_AND_TFRC_UP"
[5] "ODONNELL_TARGETS_OF_MYC_AND_TFRC_DN"  "CAIRO_PML_TARGETS_BOUND_BY_MYC_UP"
> myc.mroast <- mroast(y, index = c2.ind[myc],design = design,
+                      contrast = cont,nrot = 999)
> myc.mroast[1:5,]
                                    NGenes  PropDown     PropUp Direction PValue
YU_MYC_TARGETS_DN                        9 0.0000000 0.88888889        Up  0.001
ODONNELL_TARGETS_OF_MYC_AND_TFRC_DN     19 0.8421053 0.05263158      Down  0.001
YU_MYC_TARGETS_UP                       33 0.7575758 0.12121212      Down  0.001
LEE_LIVER_CANCER_MYC_UP                 31 0.7419355 0.22580645      Down  0.001
LEE_LIVER_CANCER_MYC_TGFA_UP            29 0.7241379 0.13793103      Down  0.001
                                      FDR PValue.Mixed FDR.Mixed
YU_MYC_TARGETS_DN                   0.001        0.001     0.001
ODONNELL_TARGETS_OF_MYC_AND_TFRC_DN 0.001        0.001     0.001
YU_MYC_TARGETS_UP                   0.001        0.001     0.001
LEE_LIVER_CANCER_MYC_UP             0.001        0.001     0.001
LEE_LIVER_CANCER_MYC_TGFA_UP        0.001        0.001     0.001
```

Console 2.20 ROAST analysis using human_c2_v5p2.rdata

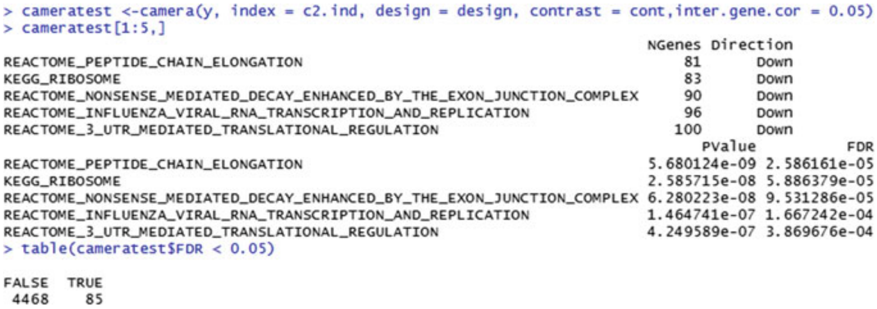
2.28 ROAST Analysis

Rotation gene set test (ROAST) [31] is another analysis that is commonly used to investigate whether the majority of genes in our prescribed comparison are DE or not. ROAST has no concern with the functioning of genes that is what kind of role they are playing in a certain process. ROAST is useful in the scenarios when all the genes involved in a specific pathway are available.

In our ROAST analysis, code given in Console 2.20, we have used human C2 gene sets, these sets are formed by assembling information from several pathway databases including KEGG and Reactome. We have run ROAST method only for the set of genes involved in MYC pathways.

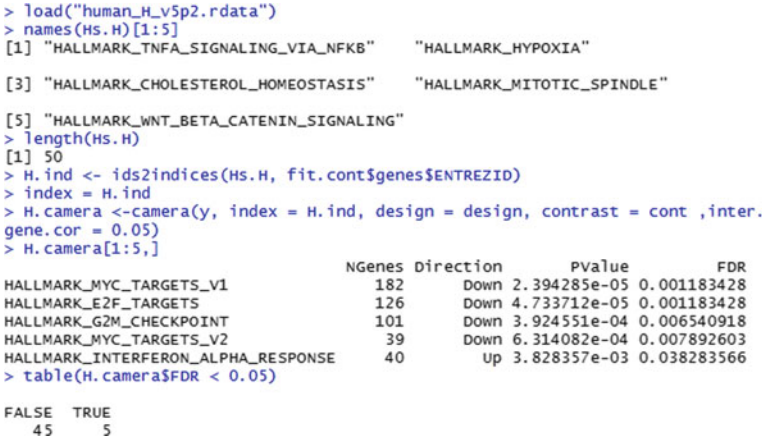
2.29 CAMERA Test

CAMERA works by fixing a small inter-gene correlation around 0.05 via the inter.gene.cor argument. We have applied CAMERA on the same human C2 gene sets used in ROAST analysis and found that the total number of significant gene sets at FDR < 0.05 is 85 (see Console 2.21).



```
> cameratest <-camera(y, index = c2.ind, design = design, contrast = cont,inter.gene.cor = 0.05)
> cameratest[1:5,]
                                                                     NGenes Direction
REACTOME_PEPTIDE_CHAIN_ELONGATION                                        81      Down
KEGG_RIBOSOME                                                            83      Down
REACTOME_NONSENSE_MEDIATED_DECAY_ENHANCED_BY_THE_EXON_JUNCTION_COMPLEX   90      Down
REACTOME_INFLUENZA_VIRAL_RNA_TRANSCRIPTION_AND_REPLICATION               96      Down
REACTOME_3_UTR_MEDIATED_TRANSLATIONAL_REGULATION                        100      Down
                                                                           PValue          FDR
REACTOME_PEPTIDE_CHAIN_ELONGATION                                      5.680124e-09 2.586161e-05
KEGG_RIBOSOME                                                          2.585715e-08 5.886379e-05
REACTOME_NONSENSE_MEDIATED_DECAY_ENHANCED_BY_THE_EXON_JUNCTION_COMPLEX 6.280223e-08 9.531286e-05
REACTOME_INFLUENZA_VIRAL_RNA_TRANSCRIPTION_AND_REPLICATION             1.464741e-07 1.667242e-04
REACTOME_3_UTR_MEDIATED_TRANSLATIONAL_REGULATION                       4.249589e-07 3.869676e-04
> table(cameratest$FDR < 0.05)

FALSE  TRUE
 4468    85
```

Console 2.21 Camera Test using human_c2_v5p2.rdata



```
> load("human_H_v5p2.rdata")
> names(Hs.H)[1:5]
[1] "HALLMARK_TNFA_SIGNALING_VIA_NFKB"    "HALLMARK_HYPOXIA"

[3] "HALLMARK_CHOLESTEROL_HOMEOSTASIS"    "HALLMARK_MITOTIC_SPINDLE"

[5] "HALLMARK_WNT_BETA_CATENIN_SIGNALING"
> length(Hs.H)
[1] 50
> H.ind <- ids2indices(Hs.H, fit.cont$genes$ENTREZID)
> index = H.ind
> H.camera <-camera(y, index = H.ind, design = design, contrast = cont ,inter.
gene.cor = 0.05)
> H.camera[1:5,]
                                  NGenes Direction       PValue         FDR
HALLMARK_MYC_TARGETS_V1              182      Down 2.394285e-05 0.001183428
HALLMARK_E2F_TARGETS                 126      Down 4.733712e-05 0.001183428
HALLMARK_G2M_CHECKPOINT              101      Down 3.924551e-04 0.006540918
HALLMARK_MYC_TARGETS_V2               39      Down 6.314082e-04 0.007892603
HALLMARK_INTERFERON_ALPHA_RESPONSE    40        Up 3.828357e-03 0.038283566
> table(H.camera$FDR < 0.05)

FALSE  TRUE
   45     5
```

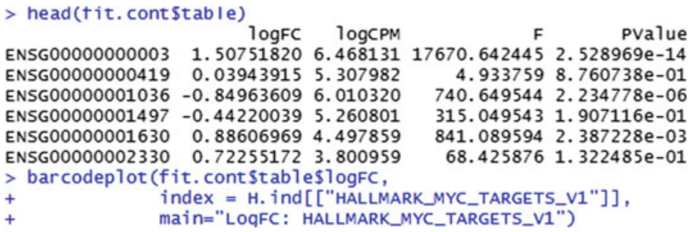
Console 2.22 Camera test using human_H_v5p2.rdata

We have run the camera test on another human gene sets human_H_v5p2.rdata including Hallmark pathways Console 2.22. In this case, only 5 genes are showing significant behaviour.

2.30 Visualizing Gene Tests

We can visualize the results for any specific gene set using barcodes. For instance, we can consider the top set HALLMARK_MYC_TARGETS_V1 obtained in CAMERA test. We have used logFC statistic for comparison between DMSO and DORMANT states (Console 2.23).

Barcode-plot in Fig. 2.12 clearly shows the down regulation of most of the genes in comparison involved in HALLMARK_MYC_TARGETS_V1 pathway.



```
> head(fit.cont$table)
                     logFC   logCPM            F       PValue
ENSG00000000003  1.50751820 6.468131 17670.642445 2.528969e-14
ENSG00000000419  0.03943915 5.307982     4.933759 8.760738e-01
ENSG00000001036 -0.84963609 6.010320   740.649544 2.234778e-06
ENSG00000001497 -0.44220039 5.260801   315.049543 1.907116e-01
ENSG00000001630  0.88606969 4.497859   841.089594 2.387228e-03
ENSG00000002330  0.72255172 3.800959    68.425876 1.322485e-01
> barcodeplot(fit.cont$table$logFC,
+             index = H.ind[["HALLMARK_MYC_TARGETS_V1"]],
+             main="LogFC: HALLMARK_MYC_TARGETS_V1")
```

Console 2.23 Code for bar plot

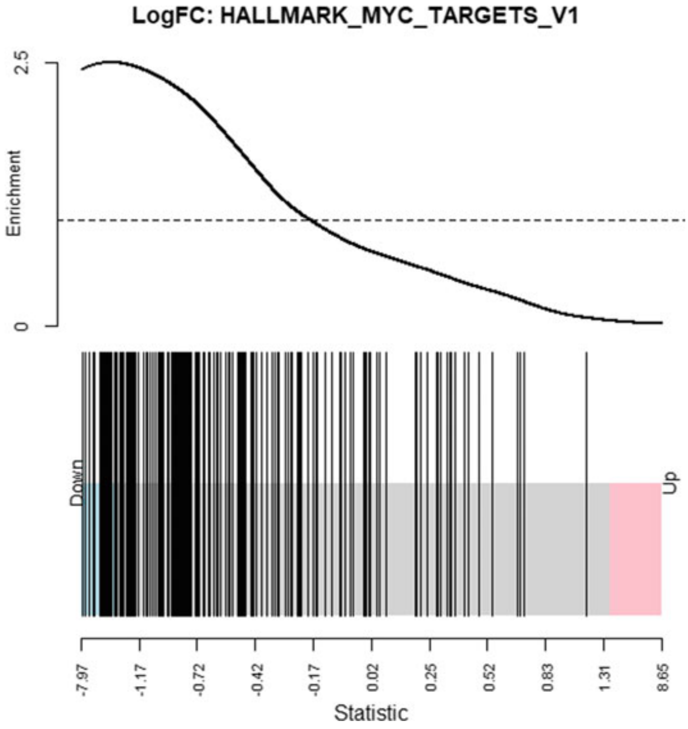
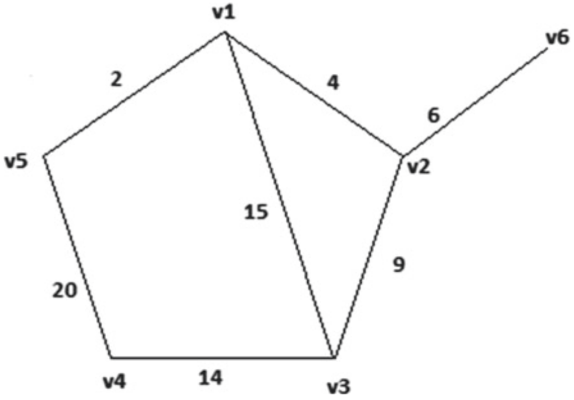


Fig. 2.12 Barcode plots for genes involved in the pathway HALLMARK_MYC_TARGETS_V1 based on LogFC

2.31 Graph Theory Terminologies

A *graph* or *network* G is a collection of vertices (nodes) $V(G)$ and edges (links) $E(G)$. Vertices are joined through edges showing a relationship among them; a graph is shown in Fig. 2.13. If the edges of the graph have directions, then we call it a *directed graph* otherwise an *undirected graph.* A *Weighted graph* has weights on the edges, the magnitude of a weight on an edge basically shows the strength of the

Fig. 2.13 A graph (network)



relationship between the vertices incident to that edge. A graph H of a graph G is called its *subgraph* or *sub-network* if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$ while H is called an *induced subgraph* of G if all the edges between the vertices in $V(G)$ from $E(G)$ are in $E(H)$. The total number of edges incident to a vertex is called *degree* of the vertex. Two edges of a graph are called *adjacent* if they share a common vertex. A *path* from a vertex u to a vertex v is a sequence of adjacent edges where u is incident to the first edge in the sequence while v is incident to the last edge of the sequence.

An adjacency matrix is a mathematical form of a graph. Each entry of the adjacency matrix shows whether we have an edge between two vertices or not. An adjacency matrix for unweighted graph has only 0 and 1 entries while the weighted adjacency matrix contains the weights of the edges. In Table 2.5, an adjacency matrix corresponding to the weighted graph given in Fig. 2.13 is shown.

Table 2.5 Adjacency matrix

Vertices	v1	v2	v3	v4	v5	v6
v1	0	4	15	0	2	0
v2	4	0	9	0	0	6
v3	15	9	0	14	0	0
v4	0	0	14	0	20	0
v5	2	0	0	20	0	0
v6	0	6	0	0	0	0

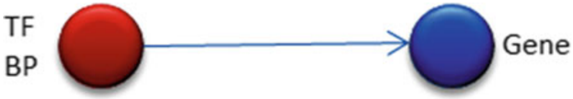


Fig. 2.14 A gene regulatory network

2.32 Gene Regulatory Network (GRN)

Many biological systems have been studied using networks. There are different types of networks used in biological processes. These types include *protein–protein interaction* (PPI), *co-expression network*, *transcriptional regulatory network* (TRN), *gene regulatory network* (GRN). PPI shows a relationship between proteins while co-expression network tells a connection between genes only. There are also some factors involved in the functioning of biological processes like TF which regulates the gene expression a network consisting of this relationship is known as TRN. GRN is used to show the relation of gene with any factor involved in its regulation either it is another gene, TF or RNA BP (RNA binding protein). Figure 2.14 shows a gene regulatory network.

2.33 Inference of Gene Regulatory Networks

Currently, a major challenge confronted by the scientists is the utilization of the data mixed out by various ‘omics’ technologies like genomics, transcriptomics, etc. Instead of encircling around an isolated gene or protein, scientists are now proceeding further to have a complete view through all stages of study including data assortment, data processing, data analysis, information achievement, hypothesis testing and consequent experimental design. This provides a new approach towards biology with dramatic effect on the way that research is implemented. Under the control of TFs, each gene affects the cellular activity through mRNA that directs the production of proteins through a complicated cellular procedure named as ribosome. During this process several biochemical reactions and molecular processes take place, some of the proteins synthesized during this process reside again in the nucleus of the cell and act as TFs to regulate gene activities. This whole scenario might be depicted through a network which is named as gene regulatory network (GRN). The inference of gene regulatory networks helps to understand the complex phenomena lying under different cellular processes by the use of information provided by the biologists. It helps us to understand the transcriptomic regulation, and the means through which gene expression profiles could be utilized to investigate the multifarious interactions among genes and proteins that make a GRN. We can do gene annotations to get information about genes by using different databases like GO, EC and KEGG which might be further incorporated to find the solution. Moreover, information about a part of network structure can give an idea to determine different features of the network like edge betweenness or network topology, to validate the inferred network.

2.34 Gene Regulatory Network Modelling

Gene expression profiles are used to construct gene regulatory networks to illustrate the phenotypic behaviour of a system under consideration. Initially, a model is developed showing the system's behaviour in a specific experimental/environmental situation to reconstruct a GRN. Further, the model predictions are compared with the given experimental data to check the performance of the model.

Subject to the reliability of the experimental data, if predicted data with certain novel conditions doesn't match the observed data then the model must be revised. This process is repeated until a reliable model is achieved.

From computational point of view, this whole process is very time consuming which motivated the researchers to develop another approach known as reverse engineering, it is a paradigm with substantial aptitude towards analysing and inferring biological networks using experimental facts [32–34]. Without prior knowledge and repeated trials, it is not possible to regenerate a network until the sufficient data is accessible to infer a gene regulatory network.

In the case of gene regulatory networks, this practice usually alters the gene network in some manner and uses computational approaches to capture the topology of underlying network. Such information can be assimilated into computational schemes to obtain an accurate model. Several machine learning and correlation-based methods have been developed to estimate parametric values for a given model to create a network which is further used to create and evaluate the simulated model by matching the behaviour of the inferred model with observed data [35–43]. Instead of using expression profiles directly, we can also incorporate the prior information deduced from the data set in regeneration of gene regulatory networks (Fig. 2.15).

A major step in reverse engineering problems is to pick a network model that is most compatible with the observed data under consideration, i.e. we can fit the data

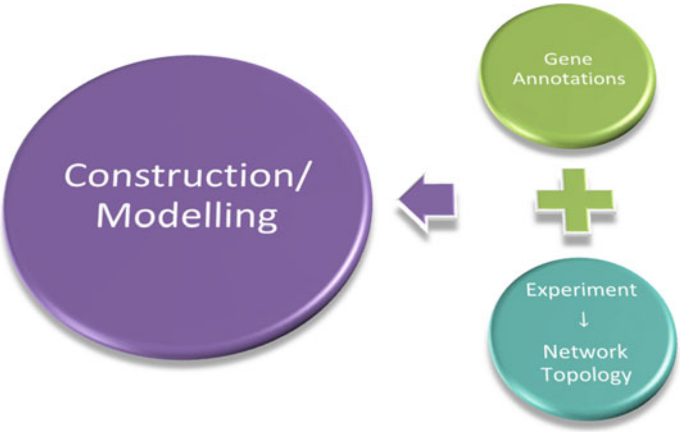


Fig. 2.15 A general framework for reverse engineering approach

into parameters of network. Many gene regulatory models have been proposed which ranges from very conceptual like Boolean to very realistic like stochastic, subject to the problem under consideration [44].

2.35 Correlation and Partial Correlation-based Methods

As discussed before mostly the expression data obtained via HT-Sequencing consists of a high number of genes (n) with a small number of experiments or samples (p). Classical statistical methods usually do not provide accurate results in the case $n > p$. To deal with such an issue, researchers have introduced the concept of sparsity in networks structures [45]. The most common approach used is based on correlation and partial correlation. Correlation-based methods generally estimate covariance matrices Σ and reconstruct co-expressed networks and modules using Σ [46–48]. Due to the transitive nature of interactions, covariance matrices also include indirect links which approximates inaccurate networks as the size of the network increases. On the other hand, partial correlation methods which are based on precision matrices $\Theta = \Sigma^{-1}$ allow only direct links among the variables. A relation between correlation and partial correlation methods is estimated in [49] using Neumann series which might be informative when analysing the performance of correlation and partial correlation-based approaches. Some of the common methods based on precision matrices include node-wise regression lasso [50], graphical lasso [51], and adaptive lasso [52] while thresholded sample covariance [48] and covariance lasso [53] are the correlation-based approaches consisting of only one parameter. A comparison between all of these methods is shown in [49].

2.36 Co-expression Networks

Co-expression network analysis is used to describe the relation between gene expression profiles. This analysis is based on finding the patterns among the genes, more precisely the groups of genes which are clustered together which are also known as modules. This analysis also provides the information about interrelated modules. Co-expression networks are basically the correlation-based networks which are commonly used to identify the genes significantly involved in the occurrence or presence of a certain disease or to find therapeutic targets. A basic pipeline to conduct this analysis on RNA-seq data is provided below.

2.37 Pre-processing of Data

Pre-processing of the data includes almost all the steps we have done in the RNA-Seq analysis earlier including $\log_2$CPM, removal of low counts genes, normalization and identification of differentially expressed genes. The step of identifying the differentially expressed genes provides more robust and accurate results in further analysis.

2.38 Construction of Covariance Matrix

To form a covariance matrix, we can set any similarity measure like Euclidean Distance (ED, Pearson, Spearman or Kendell. A combination of ED and Pearson correlation measures was introduced by [54] with a source code available at tutorial/README.md that is defined as follows:

$$S = \mathrm{sign}(\mathrm{cor}(X)) \times \frac{|\mathrm{cor}(X)| + \left(1 - \frac{\log(\mathrm{dist}(X)+1)}{\max(\log(\mathrm{dist}(X)+1))}\right)}{2}, \qquad (2.1)$$

where X is the data matrix, and cor and dist denote the correlation and distance functions, respectively. The sign function is just storing the sign of the correlation. The first term in the equation is simply the sign of the correlation function, which serves to preserve the sign of the interaction. The values closer to -1 are showing highly negatively correlated genes while a value closer to 1 is showing high similarity among the genes.

2.39 Measure of Similarity

Unsigned or signed measures are two types of similarity measures which can be defined for correlation coefficient to approximate an adjacency matrix. Let x_1 and x_2 denote the expression profiles of two genes G_1 and G_2 in an expression matrix X, respectively, and $\mathrm{cor}(x_1, x_2)$ denotes the coefficient of correlation between x_1 and x_2.

The unsigned measure between x_1 and x_2 is defined as the absolute value of the correlation between the expression values of gene x_1 and gene x_2, i.e. $s_{12} = |\mathrm{cor}(x_1, x_2)|$. It transforms all negative correlations into positive.

The signed measure between x_1 and x_2 is determined as $s_{12} = 0.5(1 + \mathrm{cor}(x_1, x_2))$. It is obvious from the definition of signed measure that it will take values on [0, 1] and two inversely expressed genes with correlation -1 will be mapped to 0.

2.40 Network Construction

Similarity, matrix $S = [s_{kl}]; k, l \in \{1, 2, \ldots, n\}$ consisting of the similarity measures s_{kl} for each pair of genes in an expression matrix X is an intermediate quantity between correlation and adjacency matrices. A hard thresholding with parameter τ or soft thresholding with parameter β is applied on the similarity measures to obtain entries of adjacency matrices of the network.

A hard thresholding approximates an unweighted adjacency matrix $A = [a_{kl}]; k, l \in \{1, 2, \ldots, n\}$ with the matrix entries a_{kl} equals 1 if the corresponding value $s_{kl} \geq \tau$ in S otherwise $a_{kl} = 0$ [48].

A weighted adjacency matrix $A = [a_{kl}]$ is estimated by a power transformation, i.e.$a_{kl} = s_{kl}^{\beta}$ for all $k, l \in \{1, 2, \ldots, n\}$; $\beta \geq 1$ [55]. On logarithmic scale, weighted adjacency measures are proportional to their corresponding similarity measures. For parameter selection, a scale-free topology criterion might be used which suggests choosing the smallest value of β needed to reach a scale free topology [55].

2.41 Module Detection

A sub-network of a network in which most of the nodes are interconnected is referred to as a module. These modules are usually signified in the form of clusters. Now, the next step is to detect distinct modules lying in a network. To accomplish this task, we need to define some measure of interconnectedness in the network. There are quite a few procedures proposed to assess network connectedness including topological overlap measure (TOM) [55–58]. TOM is used at a vast level to estimate the network interconnectedness due to its accurateness. The topological overlap of two genes is evaluated by examining neighbourhood of those genes; genes lying in the same neighbourhood are more likely to have high topological overlap.

A TOM of two nodes x_k and x_l is defined as follows [57]:

$$t_{kl} = \left\{ \begin{array}{ll} \frac{r_{kl}+a_{kl}}{\min\{m_k, m_l\}+1-a_{kl}}; & k \neq l \\ 1; & k = l \end{array} \right\}$$

where $r_{kl} = \sum_{v \neq k,l} a_{kv} a_{vl}$, $m_k = \sum_{u \neq k} a_{ku}$ and $A = [a_{kl}]$ is an $n \times n$ unweighted adjacency matrix. Since A is unweighted so the number of common neighbours of nodes k and l is equal to r_{kl}.

Hierarchical clustering is used to visualize the modules where branches of the dendogram represent the modules and can be identified by using any branch cutting method [59].

2.42 Module Enrichment

Co-expressed modules might be the indication of the existence of certain pathways or can be the results of some noise. A main objective of co-expressed module is to explore its significance from biological point of view for which gene ontology information can be employed. To do such analysis usually a gene significance measure is defined to check the importance of the gene in a particular biological process. This measure mainly specifies pathway membership of a gene and also depends on the problem under consideration.

2.43 WGCNA Package in R

Peter Langfelder introduced a software package WGCNA in R language [48] which includes several features to perform a weighted correlation network analysis. WGCNA has the ability to construct networks, identify co-expressed modules, simulate and visualize data, and selection of genes.

2.44 Co-expression Network Analysis with Real Dataset

A basic workflow to identify co-expressed modules in a network is shown below on the same real data used in the RNA-seq analysis previously. In the subsequent analysis, only DE genes between the samples of Dormant and DMSO induced cell line HCC4006 are considered. The main objective is to detect the co-expressed genes in dormant state of the tumour. To visualize networks, the platform of Cytoscape is used which is especially designed to visualize complex networks, and is freely available software and can be downloaded from https://cytoscape.org/.

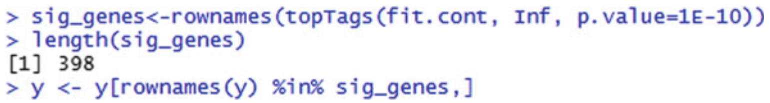
R Packages used in co-expression network analysis:

(1) ggplot2,
(2) knitr,
(3) reshape2,
(4) WGCNA.

We will start our analysis from the list of DE genes obtained in our previous RNA-seq analysis by focusing on the highly ranked genes based on P − values. We have set a cutoff for P − values see Console 2.25. We are left with only 398 genes for further analysis; this P − values cutoff is defined just for tutorial purpose.

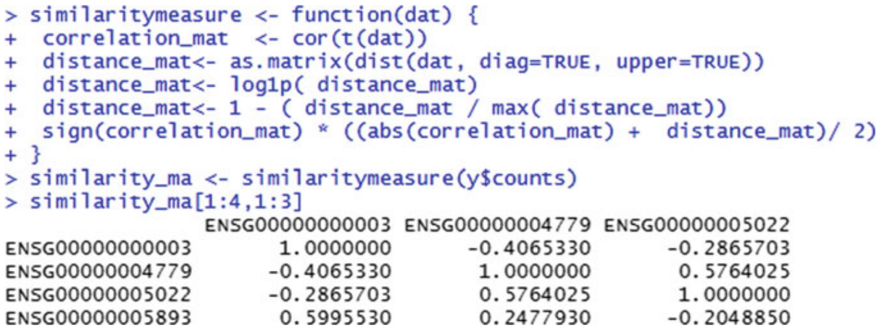
Next we will obtain a similarity matrix Console 2.25 by defining similarity measure as a combination of Pearson correlation and Euclidean distance as given in Eq. (2.1).

Let’s have a look at the similarity matrix Fig. 2.16 by using heatmap.2() function as in Console 2.26.



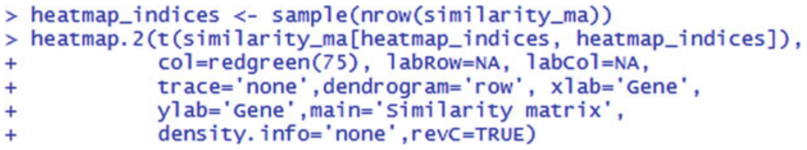
```
> sig_genes<-rownames(topTags(fit.cont, Inf, p.value=1E-10))
> length(sig_genes)
[1] 398
> y <- y[rownames(y) %in% sig_genes,]
```

Console 2.24 P-value thresholding



```
> similaritymeasure <- function(dat) {
+  correlation_mat  <- cor(t(dat))
+  distance_mat<- as.matrix(dist(dat, diag=TRUE, upper=TRUE))
+  distance_mat<- log1p( distance_mat)
+  distance_mat<- 1 - ( distance_mat / max( distance_mat))
+  sign(correlation_mat) * ((abs(correlation_mat) +  distance_mat)/ 2)
+ }
> similarity_ma <- similaritymeasure(y$counts)
> similarity_ma[1:4,1:3]
                ENSG00000000003 ENSG00000004779 ENSG00000005022
ENSG00000000003       1.0000000      -0.4065330      -0.2865703
ENSG00000004779      -0.4065330       1.0000000       0.5764025
ENSG00000005022      -0.2865703       0.5764025       1.0000000
ENSG00000005893       0.5995530       0.2477930      -0.2048850
```

Console 2.25 Similarity matrix



```
> heatmap_indices <- sample(nrow(similarity_ma))
> heatmap.2(t(similarity_ma[heatmap_indices, heatmap_indices]),
+           col=redgreen(75), labRow=NA, labCol=NA,
+           trace='none',dendrogram='row', xlab='Gene',
+           ylab='Gene',main='Similarity matrix',
+           density.info='none',revC=TRUE)
```

Console 2.26 Heatmap code for similarity matrix

Firstly, we will shift the signed measures from [−1, 1] to [0, 1] by using signed type and then we will transform the shifted measures by using power transformation (Console 2.27).

Now let's have a look at the heatmap of adjacency matrix Fig. 2.17. We can see that the major region of the plot is red which is giving an idea about the good sparsity of the network. This is our required co-expression network.

We can further analyse this network by detecting different modules. We will use hclust() function (Console 2.29) to detect modules in our network. A gene clustering dendogram is shown in Fig. 2.18.

We can write the graph into '*graphml*' format using '*igraph*' package, for more detail how to convert a graph into some other format see the following link: https://cneurocvs.rmki.kfki.hu/igraph/doc/R/write.graph.html.

We further import this file into Cytoscape to visualize the network. We firstly filtered the genes by setting the option Degree in + out is not between 0 and 1 inclusive. We have used the attribute circle layout with module and degree.layout options. Both networks are presented in Figs. 2.19 and 2.20. There are several other

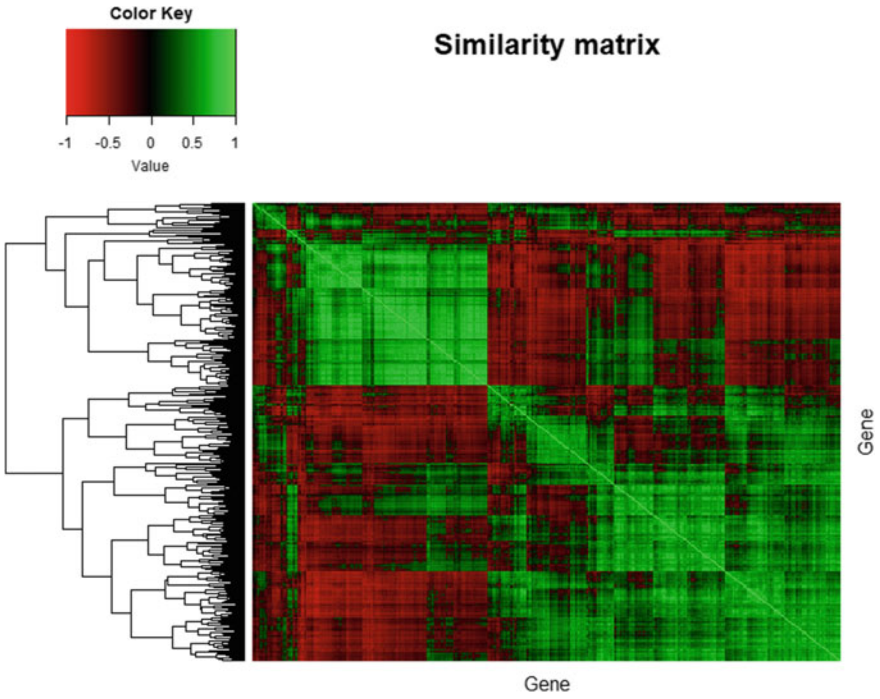
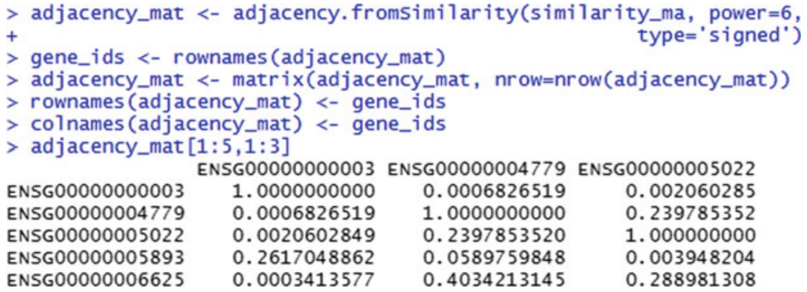
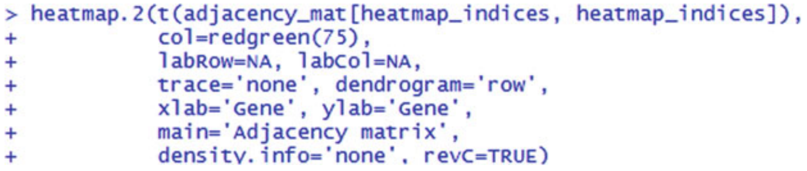


Fig. 2.16 Similarity matrix visualization



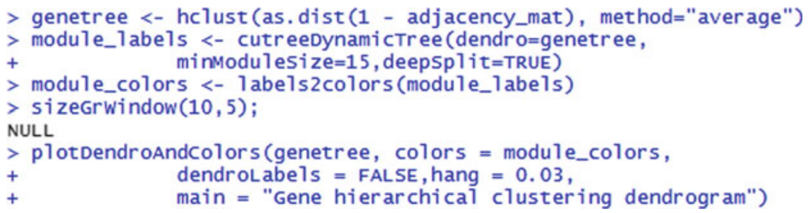
```
> adjacency_mat <- adjacency.fromSimilarity(similarity_ma, power=6,
+                                                         type='signed')
> gene_ids <- rownames(adjacency_mat)
> adjacency_mat <- matrix(adjacency_mat, nrow=nrow(adjacency_mat))
> rownames(adjacency_mat) <- gene_ids
> colnames(adjacency_mat) <- gene_ids
> adjacency_mat[1:5,1:3]
                ENSG00000000003 ENSG00000004779 ENSG00000005022
ENSG00000000003    1.0000000000    0.0006826519     0.002060285
ENSG00000004779    0.0006826519    1.0000000000     0.239785352
ENSG00000005022    0.0020602849    0.2397853520     1.000000000
ENSG00000005893    0.2617048862    0.0589759848     0.003948204
ENSG00000006625    0.0003413577    0.4034213145     0.288981308
```

Console 2.27 Adjacency matrix code



```
> heatmap.2(t(adjacency_mat[heatmap_indices, heatmap_indices]),
+           col=redgreen(75),
+           labRow=NA, labCol=NA,
+           trace='none', dendrogram='row',
+           xlab='Gene', ylab='Gene',
+           main='Adjacency matrix',
+           density.info='none', revC=TRUE)
```

Console 2.28 Adjacency matrix heatmap code



```
> genetree <- hclust(as.dist(1 - adjacency_mat), method="average")
> module_labels <- cutreeDynamicTree(dendro=genetree,
+             minModuleSize=15,deepSplit=TRUE)
> module_colors <- labels2colors(module_labels)
> sizeGrWindow(10,5);
NULL
> plotDendroAndColors(genetree, colors = module_colors,
+             dendroLabels = FALSE,hang = 0.03,
+             main = "Gene hierarchical clustering dendrogram")
```

Console 2.29 Gene dendogram code

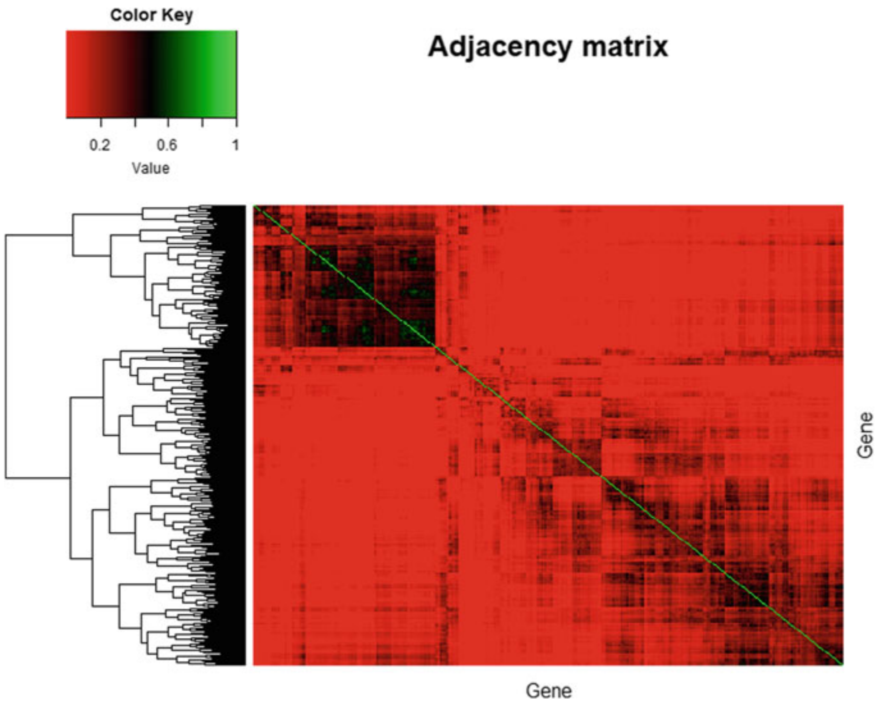


Fig. 2.17 Visualization of adjacency matrix

functions available in Cytoscape which might be used for further analysis. To have a complete look at Cytoscape features see the tutorials available at Cytoscape website https://cytoscape.org/.

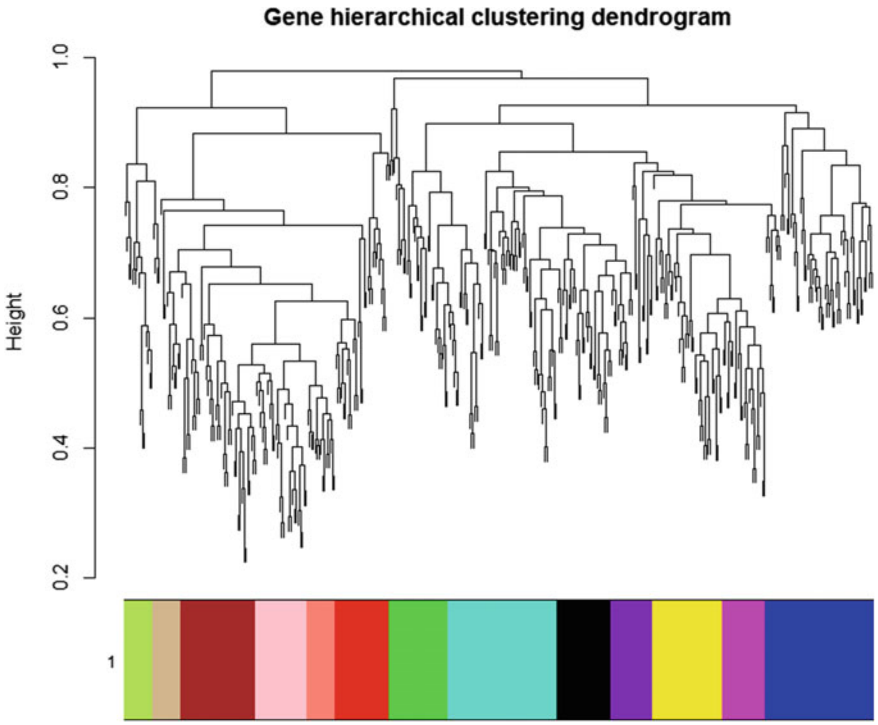


Fig. 2.18 Gene clustering dendogram

2.45 Concluding Remarks

This chapter explains the basic steps needed for an RNA-seq and co-expression network analyses. Real data of lung cancer cell lines have been selected for illustration purpose which is downloaded from GEO database. Our data contains 3 types of cell lines PC-9, HCC827 and HCC4006 along with two distinct biological conditions DMSO induced cancer lines and the DORMANT state of cancer. We have used GO analysis, CAMERA test and roast test for gene enrichment analysis. The co-expression network analysis has been conducted on the same genes found differentially expressed during RNA-seq analysis. Co-expression network analysis helped us to find the subsets of genes which are densely connected which might be the indication of their similar functioning. This chapter also briefly describes the concepts of normalization, biological and technical variations found in gene expression data, principal component analysis and generalized linear models. A description of gene regulatory networks inference is also provided in the chapter.

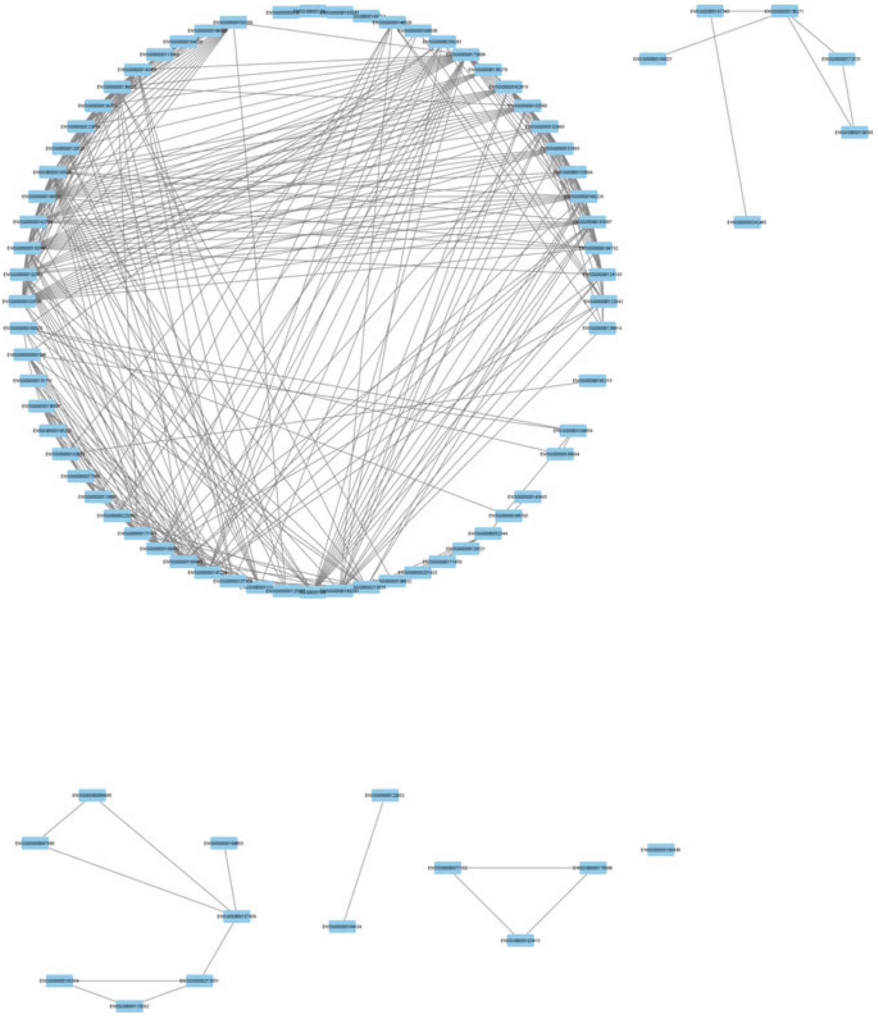


Fig. 2.19 Circle layout with module preference

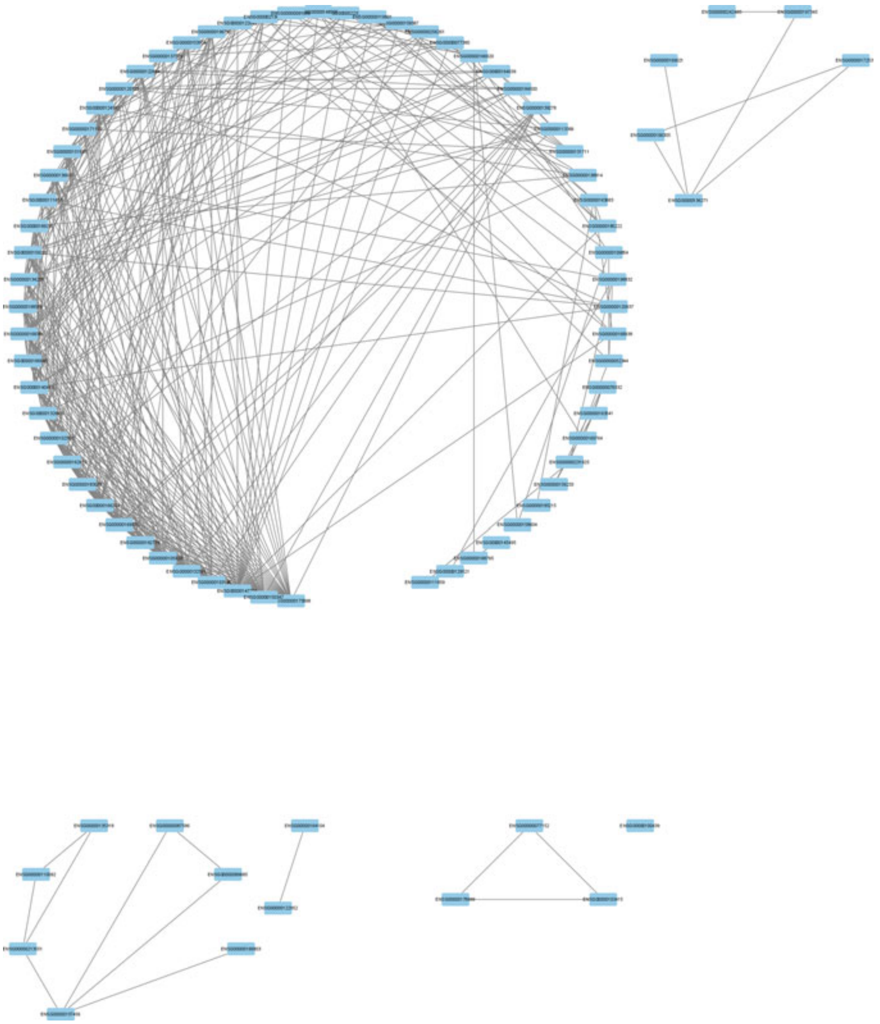


Fig. 2.20 Circle layout with degree.layout

References

1. Tavassoly I, Goldfarb J, Iyengar R (2018) Systems biology primer: the basic methods and approaches. Essays Biochem 62(4):487–500. https://doi.org/10.1042/EBC20180003
2. Longo G, Montévil M (2014) Perspectives in organisms. Lecture Notes in Morphogenesis, pp 23–27. Available at: https://link.springer.com/content/pdf/10.1007/978-3-642-35938-5.pdf
3. Bu Z, Callaway DJE (2011) Chapter 5—Proteins MOVE! Protein dynamics and long-range allostery in cell signaling. In: Donev RBT-A, P. C. and S. B. (ed.) Protein structure and diseases. Academic Press, pp 163–221. https://doi.org/10.1016/B978-0-12-381262-9.00005-7
4. Zewail AH (2008) Physical biology: from atoms to medicine. Imperial college press

5. Churko JM et al (2013) Overview of high throughput sequencing technologies to elucidate molecular pathways in cardiovascular diseases. Circ Res 112(12): 1613–1623. https://doi.org/10.1161/CIRCRESAHA.113.300939
6. Schuster SC (2008) Next-generation sequencing transforms today's biology. Nat Methods 5(1):16–18. https://doi.org/10.1038/nmeth1156
7. Zhao S et al (2014) Comparison of RNA-seq and microarray in transcriptome profiling of activated T cells. PloS one. Public Library of Science, 9(1): e78644
8. Liao Y, Smyth GK, Shi W (2014) featureCounts: an efficient general purpose program for assigning sequence reads to genomic features. Bioinformatics 30(7):923–930. https://doi.org/10.1093/bioinformatics/btt656
9. Anders S, Pyl PT, Huber W (2015) HTSeq—a Python framework to work with high-throughput sequencing data. Bioinformatics 31(2):166–169. https://doi.org/10.1093/bioinformatics/btu638
10. McCarthy DJ, Chen Y, Smyth GK (2012) Differential expression analysis of multifactor RNA-seq experiments with respect to biological variation. Nucleic Acids Res 40(10): 4288–4297. https://doi.org/10.1093/nar/gks042
11. 't Hoen PAC et al (2008) Deep sequencing-based expression analysis shows major advances in robustness, resolution and inter-lab portability over five microarray platforms. Nucleic Acids Res 36(21): e141. https://doi.org/10.1093/nar/gkn705
12. Cloonan N et al (2008) Stem cell transcriptome profiling via massive-scale mRNA sequencing. Nat Methods. United States, 5(7): 613–619. https://doi.org/10.1038/nmeth.1223
13. Langmead B, Hansen KD, Leek JT (2010) Cloud-scale RNA-sequencing differential expression analysis with Myrna. Genome Biol 11(8): R83. https://doi.org/10.1186/gb-2010-11-8-r83
14. Anders S, Huber W (2010) Differential expression analysis for sequence count data. Genome Biol 11(10): R106. https://doi.org/10.1186/gb-2010-11-10-r106
15. Robinson MD, McCarthy DJ, Smyth GK (2010) edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics (Oxford, England), 26(1): 139–140. https://doi.org/10.1093/bioinformatics/btp616
16. Robinson MD, Smyth GK (2007) Moderated statistical tests for assessing differences in tag abundance. Bioinformatics (Oxford, England). England, 23(21): 2881–2887. https://doi.org/10.1093/bioinformatics/btm453
17. Robinson MD, Smyth GK (2008) Small-sample estimation of negative binomial dispersion, with applications to SAGE data. Biostatistics (Oxford, England). England, 9(2): 321–332. https://doi.org/10.1093/biostatistics/kxm030
18. Nagalakshmi U et al (2008) The transcriptional landscape of the yeast genome defined by RNA sequencing. Science (New York, N.Y.), 320(5881): 1344–1349. https://doi.org/10.1126/science.1158441
19. Lund SP et al (2012) Detecting differential expression in RNA-sequence data using quasi-likelihood with shrunken dispersion estimates. Stat Appl Genet Mol Biol. De Gruyter, 11(5)
20. Lun ATL, Chen Y, Smyth GK (2016) It's DE-licious: a recipe for differential expression analyses of RNA-seq experiments using Quasi-Likelihood methods in edgeR. Methods in molecular biology (Clifton, N.J.). United States, vol 1418, pp 391–416. https://doi.org/10.1007/978-1-4939-3578-9_19
21. Phipson B et al (2013) Empirical Bayes in the presence of exceptional cases, with application to microarray data. Phytochemistry 26(8):2247–2250
22. Smyth GK (2004) Linear models and empirical bayes methods for assessing differential expression in microarray experiments. Stat Appl Genet Mol Biol. De Gruyter, 3(1)
23. Zhou Y-H, Xia K, Wright FA (2011) A powerful and flexible approach to the analysis of RNA sequence count data. Bioinformatics (Oxford, England), 27(19): 2672–2678. https://doi.org/10.1093/bioinformatics/btr449
24. Wu H, Wang C, Wu Z (2013) A new shrinkage estimator for dispersion improves differential expression detection in RNA-seq data. Biostatistics (Oxford, England), 14(2): 232–243. https://doi.org/10.1093/biostatistics/kxs033

25. Hardcastle TJ, Kelly KA (2010) baySeq: empirical Bayesian methods for identifying differential expression in sequence count data. BMC Bioinform 11: 422. https://doi.org/10.1186/1471-2105-11-422
26. Van De Wiel MA et al (2013) Bayesian analysis of RNA sequencing data by estimating multiple shrinkage priors. Biostatistics (Oxford, England). England, 14(1): 113–128. https://doi.org/10.1093/biostatistics/kxs031
27. Robinson MD, Oshlack A (2010) A scaling normalization method for differential expression analysis of RNA-seq data. Genome Biol 11(3):R25. https://doi.org/10.1186/gb-2010-11-3-r25
28. Bullard JH et al (2010) Evaluation of statistical methods for normalization and differential expression in mRNA-seq experiments. BMC Bioinform 11(1):94. https://doi.org/10.1186/1471-2105-11-94
29. Benjamini Y, Hochberg Y (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. J Roy Stat Soc: Ser B (Methodological). Wiley Online Library, 57(1): 289–300
30. Kurppa KJ et al (2020) Treatment-induced tumor dormancy through YAP-mediated transcriptional reprogramming of the apoptotic pathway. Cancer Cell 37(1): 104–122.e12. https://doi.org/10.1016/j.ccell.2019.12.006
31. Wu D et al (2010) ROAST: rotation gene set tests for complex microarray experiments. Bioinformatics. Oxford University Press, 26(17): 2176–2182
32. Cho K-H et al (2007) Reverse engineering of gene regulatory networks. IET Syst Biol. IET 1(3):149–163
33. Csete ME, Doyle JC (2002) Reverse engineering of biological complexity. Science. American Association for the Advancement of Science, 295(5560): 1664–1669
34. Kitano H (2000) Perspectives on systems biology. New Gener Comput. Springer, 18(3): 199–216
35. Bansal M et al (2007) How to infer gene networks from expression profiles. Mol Syst Biol 3: 78. https://doi.org/10.1038/msb4100120
36. Bellazzi R, Zupan B (2007) Towards knowledge-based gene expression data mining. J Biomed Infor. United States, 40(6): 787–802. https://doi.org/10.1016/j.jbi.2007.06.005
37. Ernst J et al (2007) Reconstructing dynamic regulatory maps. Mol Syst Biol 3: 74. https://doi.org/10.1038/msb4100115
38. Friedman N (2004) Inferring cellular networks using probabilistic graphical models. Science (New York, N.Y.). United States, 303(5659): 799–805. https://doi.org/10.1126/science.1094068
39. Gilbert D et al (2006) Computational methodologies for modelling, analysis and simulation of signalling networks. Briefings Bioinform. England, 7(4): 339–353. https://doi.org/10.1093/bib/bbl043
40. Hecker M et al (2009) Gene regulatory network inference: data integration in dynamic models-a review. Bio Syst. Ireland, 96(1): 86–103. https://doi.org/10.1016/j.biosystems.2008.12.004
41. Markowetz F, Spang R (2007) Inferring cellular networks--a review. BMC Bioinform 8(Suppl 6): S5. https://doi.org/10.1186/1471-2105-8-S6-S5
42. Schlitt T, Brazma A (2007) Current approaches to gene regulatory network modelling. BMC Bioinform 8(Suppl 6): S9. https://doi.org/10.1186/1471-2105-8-S6-S9
43. Stigler B et al (2007) Reverse engineering of dynamic networks. Ann New York Acad Sci. United States, 1115: 168–177. https://doi.org/10.1196/annals.1407.012
44. Lee WP, Tzou W-S (2009) Computational methods for discovering gene networks from expression data. Briefings Bioinform 10(4): 408–423. https://doi.org/10.1093/bib/bbp028
45. Bühlmann P, Van De Geer S (2011) Statistics for high-dimensional data: methods, theory and applications. Springer Science & Business Media
46. Dong J, Horvath S (2007) Understanding network concepts in modules. BMC Syst Biol. Springer 1(1):24
47. Horvath S, Dong J (2008) Geometric interpretation of gene coexpression network analysis. PLoS Comput Biol. Public Library of Science, 4(8): e1000117

48. Langfelder P, Horvath S (2008) WGCNA: an R package for weighted correlation network analysis. BMC Bioinform 9(1):559. https://doi.org/10.1186/1471-2105-9-559
49. Sulaimanov N, Koeppl H (2016) Graph reconstruction using covariance-based methods. EURASIP J Bioinf Syst Biol 1:19. https://doi.org/10.1186/s13637-016-0052-y
50. Meinshausen N, Bühlmann P (2006) High-dimensional graphs and variable selection with the lasso. The Ann Stat. Institute of Mathematical Statistics, 34(3): 1436–1462
51. Friedman J, Hastie T, Tibshirani R (2008) Sparse inverse covariance estimation with the graphical lasso. Biostatistics. Oxford University Press, 9(3): 432–441
52. Zou H (2006) The adaptive lasso and its oracle properties. J Am Stat Assoc. Taylor & Francis, 101(476): 1418–1429
53. Bien J, Tibshirani RJ (2011) Sparse estimation of a covariance matrix. Biometrika. Oxford University Press, 98(4): 807–820
54. Inbar E et al (2017) The Transcriptome of *Leishmania major* developmental stages in their natural sand fly vector. mBio 8(2): e00029–17 (Edited by L. D. Sibley). https://doi.org/10.1128/mBio.00029-17
55. Zhang B, Horvath S (2005) A general framework for weighted gene co-expression network analysis. Stat Appl Genet Mol Biol. De Gruyter, 4(1)
56. Li A, Horvath S (2007) Network neighborhood analysis with the multi-node topological overlap measure. Bioinformatics. Oxford University Press, 23(2): 222–231
57. Ravasz E et al (2002) Hierarchical organization of modularity in metabolic networks. Science (New York, N.Y.). United States, 297(5586): 1551–1555. https://doi.org/10.1126/science.1073374
58. Yip AM, Horvath S (2007) Gene network interconnectedness and the generalized topological overlap measure. BMC Bioinform. BioMed Central 8(1): 22
59. Langfelder P, Zhang B, Horvath S (2008) Defining clusters from a hierarchical cluster tree: the Dynamic Tree Cut package for R. Bioinformatics (Oxford, England). England, 24(5): 719–720. https://doi.org/10.1093/bioinformatics/btm563

Chapter 3
Learning Biomedical Networks: Toward Data-Informed Clinical Decision and Therapy

Marta B. Lopes and Susana Vinga

Abstract Precision medicine has emerged to tailor clinical decisions based on patient genetic features in a personalized healthcare perspective. The ultimate goal is to drive disease diagnosis and treatment selection based on the patient molecular profiles, usually given by large volumes of data, which is intrinsically high-dimensional, heterogeneous, noisy, and incomplete. Along with the notable improvement of experimental technologies, statistical learning has accompanied the associated challenges by the significant development of novel methods and algorithms. In particular, network-based learning is providing promising results toward more personalized medicine. This short survey will describe three main interconnected trends identified to address these challenges and all with a firm root in network science: differential network analysis, network-based regularization, and causal discovery and inference. An overview of the main applications is provided, along with available software. Biomedical networks support more informed and interpretable statistical learning models from patients' data, thus improving clinical decisions and supporting therapy optimization.

3.1 Biological Data and the Rise of Targeted Therapies

Precision medicine integrates clinical and molecular data (e.g., genetic aberrations and gene expression) to deciphering disease mechanisms and improve clinical decisions and patient outcomes. The ultimate goal is to drive disease diagnosis and treat-

M. B. Lopes
NOVA Laboratory for Computer Science and Informatics (NOVA LINCS), Centre for Mathematics and Applications (CMA), Faculdade de Ciências e Tecnologia, Universidade NOVA de Lisboa, 2829-516 Caparica, Portugal
e-mail: marta.lopes@fct.unl.pt

S. Vinga (✉)
INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Rua Alves Redol 9, 1000-129 Lisboa, Portugal
e-mail: susanavinga@tecnico.ulisboa.pt

T. D. Pham et al. (eds.), *Advances in Artificial Intelligence, Computation, and Data Science*, Computational Biology 31,
https://doi.org/10.1007/978-3-030-69951-2_3

ment selection based on the patient molecular profiles, such as genetic aberrations and gene expression.

As the sequencing technologies rapidly evolve, massive amounts of information per patient from multiple molecular layers by diverse omic assays are generated, which is expected to provide comprehensive insights into cellular behavior. This growth in data brings, however, many challenges to be tackled for effective knowledge discovery. Indeed, the data is usually high-dimensional, with often the number of features, such as the whole transcriptome, being much larger than the number of available observations (e.g., patients), which hampers the unequivocal identification of a model. The data usually comes from different sources and is intrinsically heterogeneous, which also calls for novel methods that can cope with a mix of different data types (nominal, ordinal, interval) and characteristics (static, temporal). Missing data and noise also pose interesting questions regarding model estimation, adding further obstacles to inferring accurate models.

Great international efforts have been put forth to the construction of biological databases to support network analysis. These comprise, e.g., genome/proteome projects, pathway annotation, and protein-protein interaction (PPI) libraries. For a non-exhaustive review of available databases, see Table 3.1 and Refs. [1–3].

Alongside the remarkable explosion of molecular data and challenges, there is, therefore, a continuous demand for the development of statistical and machine learning methods able to translate the great amounts of data into clinical-oriented and therapeutical targetable mechanisms, ultimately contributing to a proactive, preventive, and cost-effective healthcare. For example, major cancer centers worldwide are now offering Next Generation Sequencing (NGS)-based personalized oncology for clinical practice, e.g., via institutional molecular tumor boards, built upon the molecular profiling of genetic aberrations of tumors and potential matching treatments [4].

Network learning is becoming crucial to address the described challenges. Many biological processes and phenomena can indeed be represented via networks (gene regulatory, metabolic, protein-protein interaction); therefore, it not surprising that the incorporation of network science in the learning process is now the state-of-the-art procedure toward more interpretable models and better decision-making.

Among many recent endeavors in network learning, the present short survey focuses on three specific topics: (1) differential networks, (2) network-based regularization, and (3) causal inference. This overall learning process is illustrated in Fig. 3.1.

3.2 Network Analysis in Biomedical Informatics

The major goal of network discovery in molecular data is to estimate associations and the cascade effect of DNA variants on downstream phenotypes and potentially identify therapeutic targets across multilayered molecular data.

Identifying association and causality in complex disease phenotypes plays a crucial role in effective healthcare management. Network reconstruction has assumed

Table 3.1 Biological databases. Databases with information regarding the Genome/Proteome, for Pathway Analysis, and Protein-Protein (PPI) networks

Databases	Link
Genome/Proteome	
The Cancer Genome Atlas (TCGA)	https://portal.gdc.cancer.gov
International Cancer Genome Consortium	https://dcc.icgc.org
Gene Expression Omnibus (GEO)	https://www.ncbi.nlm.nih.gov/geo
European Genome-phenome Archive repository (EGA)	https://www.ebi.ac.uk/ega
Clinical Proteomic Tumor Analysis Consortium (CPTAC)	https://proteomics.cancer.gov/programs/cptac
Genotype-Tissue Expression (GTEx)	https://gtexportal.org
ArrayExpress	https://www.ebi.ac.uk/arrayexpress
Sequence Read Archive (SRA)	https://www.ncbi.nlm.nih.gov/sra
Pathway Analysis	
g:Profiler	http://biit.cs.ut.ee/gprofiler/gost
Gene Set Enrichment Analysis (GSEA)	https://www.gsea-msigdb.org/gsea/
Cytoscape	http://apps.cytoscape.org/
Kyoto Encyclopedia of Genes and Genomes (KEGG)	http://www.genome.jp/kegg
Reactome	https://reactome.org
BioCarta	http://www.biocarta.com
BioCyc	https://biocyc.org
VisANT	http://www.visantnet.org
GeneAnalytics	https://geneanalytics.genecards.org
FunRich	http://www.funrich.org/
Pathway Commons	https://www.pathwaycommons.org
PPI-Networks	
BioGrid	https://thebiogrid.org
IntAct	https://www.ebi.ac.uk/intact
STRING	https://string-db.org
MINT	https://mint.bio.uniroma2.it
HPRD	http://www.hprd.org

a pivotal role in biomedical sciences. An immediate goal is to identify gene subnetworks involved in the disease process and improve model performance to predict future outcomes. However, the increasingly available multilayered omics data on large populations comes at the cost of high dimensionality, measurement error, missing data, and multiple platform integration, which represent great challenges when recovering the network structure underlying the biological process.

Disease-specific gene biomarkers are crucial molecular tools in disease management that allow understanding the molecular pathogenesis, disease stratification and

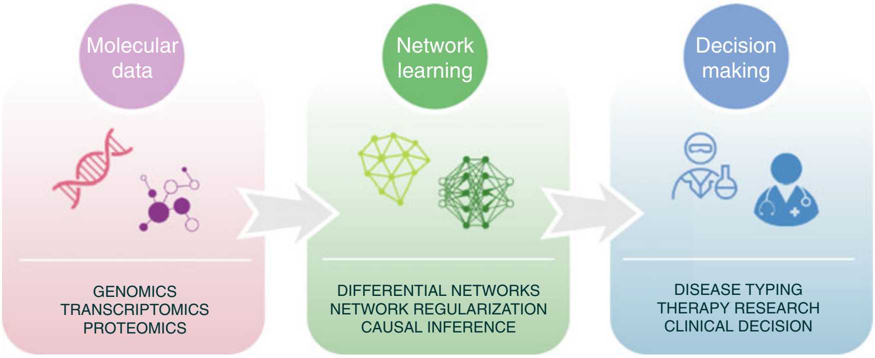


Fig. 3.1 Overview of the network learning process in precision medicine. From molecular data to decision-making, using network learning

monitoring, and therapeutic response. Although disclosing single-gene biomarkers remains clinically useful, network biomarkers are gaining attention as genes interact in a network to carry a function and define complex phenotypes. Besides allowing for an increased understanding of cellular mechanisms, such information is expected to improve estimation and inference. It is noteworthy that, in precision oncology, cancer-driven aberrations often target genes in the same pathways [5], which further strengthens the application of network-based learning for analyzing cancer data and for obtaining interpretable models.

3.2.1 Differential Network Analysis

Molecular aberrations across multiple genes and pathways are in the genesis of cancer development and progression [6]. Identifying differentially expressed genes in two groups (e.g., patients of different disease types and states, cell types, treatment response) is a common and essential task in cancer genomics. However, traditional methods for differential analysis focus on individual genes, disregarding the cooperation between genes in biological systems. Therefore, a shift from differential gene expression to a differential network is of paramount importance to identify dysfunctional regulatory networks in disease states [7]. Examples of contributions in this field are the Two-Dimensional Graphical lasso (TDJGL) [8], Integrated Differential Expression and Differential Network Analysis (INDEED) [9], Differential Network Analysis in Genomics (DINGO, iDINGO) [6, 10], and Differential Gene Regulatory Network (DiffGRN) [11], all based on network regularization (see next section), with applications in asthma and cancer domains.

3.2.2 Network-Based Regularization

Patientomics data comprise thousands of variables, largely outnumbering the number of samples, which calls for the need to find low-dimensional data explanations. Regularized optimization is a state-of-the-art strategy to promote the selection of a subset of relevant features while learning the model (sparsity) or to include a priori knowledge via constraints [12]. In this regard, the Least Absolute Shrinkage and Selection Operator (lasso) [13] and the Elastic Net (EN) [14] are among the most widely used regularizers for feature selection in high-dimensional and highly correlated data.

Beyond methods that penalize combinations of ℓ_p-norms, there is a growing interest in regularizers that impose more sophisticated constraints and structure to the obtained solutions. For example, it is possible to induce grouping formation, such as group lasso [15] and fused lasso [16, 17], to name a few. More recently, network-based regularizers have provided additional insight for structured models complying with many different assumptions (see [18] for a review).

Formally, a network is represented by a weighted graph $G = (V, E, W)$, where V is the set of the p vertices or nodes, $(i, j) \in E$ is an element of the edge set $E \subseteq V \times V$, and $\mathbf{W} = (w_{ij})$ are the weights associated with each edge.

Gene networks can be represented by graphs, in which vertices are genes, and edges represent a weighted relation between genes. A primary option for network analysis concerns identifying aberrant subnetworks derived from cancer processes and predictive of disease phenotypes. This can be achieved through graphical models. The graph Laplacian [19] and graphical lasso [20] are popular network-based regularizers.

Li and Li (2008) [21] introduced a network-constrained penalty function that penalizes the ℓ_1-norm (lasso) of the coefficients while encouraging the smoothness of the coefficients on the network through a graph Laplacian constraint (Eq. (3.1)):

$$\begin{aligned} F(\boldsymbol{\beta}) &= \lambda_1 \|\boldsymbol{\beta}\|_1 + \lambda_2 \boldsymbol{\beta}^T \mathbf{L} \boldsymbol{\beta} \\ &= \lambda_1 \sum_{i=1}^{p} |\beta_i| + \lambda_2 \sum_{(i,j)\in E} w_{ij} \left(\frac{\beta_i}{\sqrt{d_i}} - \frac{\beta_j}{\sqrt{d_j}} \right)^2, \end{aligned} \tag{3.1}$$

where $\mathbf{L} = \{l_{ij}\}$ is the symmetric normalized Laplacian matrix defined by

$$l_{ij} = \begin{cases} 1 - \frac{w_{ij}}{d_i}, & \text{if } i = j \text{ and } d_i \neq 0, \\ -\frac{w_{ij}}{\sqrt{d_i d_j}}, & \text{if } (i, j) \in E, \\ 0, & \text{otherwise,} \end{cases} \tag{3.2}$$

and d_i represents the degree of each node

$$d_i = \sum_{j:(i,j)\in E} w_{ij}. \tag{3.3}$$

The method was successful in the identification of subnetworks based on glioblastoma microarray gene-expression data, related to survival in this cancer.

In the survival analysis domain, Zhang et al. (2013) [22] proposed a network-based Cox proportional hazard model, Net-Cox, in which a graph Laplacian constraint is introduced as a smoothness requirement on the gene network. It is noteworthy that the graph Laplacian was also used in logistic and Support Vector Machines (SVM), illustrating its broad applicability ([5] and references therein).

Graphical lasso has been introduced by Friedman and co-authors (2008) [20] to estimate sparse graphs by a lasso penalty applied to the inverse covariance matrix in simulated and cell-signaling proteomics scenarios, thus allowing the estimation of partial correlation networks.

Danaher and co-authors (2014) [23] proposed the joint graphical lasso (JGL) as an extension to graphical lasso, to estimate graphical models in high-dimensional data, with observations belonging to different classes, to overcome the assumption of all samples drawn from the same distribution. JGL employs a fast ADMM algorithm with generalized fused lasso or group lasso penalties. The method enables the identification of shared edges between networks (classes) and edges present in single networks. The suitability of the method for high-dimensional scenarios was demonstrated for a lung cancer microarray dataset accounting for $\sim 18,000$ features.

To take advantage of the increasing availability of patients' data from multiple platforms of high-throughput technologies, Zhang and co-authors [8] proposed a two-dimensional joint graphical lasso (TDJGL) model to simultaneously infer group-specific gene dependency networks from gene expression data collected from different platforms. The performance of TDJGL was shown in the identification of differential networks associated with platinum response in ovarian cancer.

Also aiming at integrating multiple types of molecular (omic) data toward biomarker discovery, a modularity-constrained lasso model has been presented to analyze Alzheimer's disease data generated by multi-omic platforms jointly (e.g., genotype, gene expression, and protein expression) [24]. The penalty term maximizes the global modularity of the subnetwork of selected markers and encourages the selection of trans-omic markers with dense functional connectivity.

When the goal is to build a predictive model, the state-of-the-art modeling direction is to incorporate prior graph-based knowledge on the gene features via model constraints and drive parameter estimation toward explainable biological solutions. Network-based model regularization has been proposed to such an end. Prior information comes, e.g., from node centrality measures obtained from external available biological networks. Another option is to consider node association measures given by the correlation/covariance network structure of the input data, such as in `DegreeCox` [25], `twiner` [26, 27], and `TCox` [28], as shown in the development of survival and classification models based on bulk and single-cell RNA-sequencing data from cancer patients. In a recent work extending lasso and EN in the presence of highly correlated variables, a graph corresponding to the covariance structure of the covariates was used to regularize regression weights to promote alignment with the covariance structure [29].

3.2.3 Causal Discovery and Inference

Causal inference has been a century-old quest, whose progress has an indisputable impact in biomedical informatics [30]. Philosophers and scientists have long been debating how to identify causal relationships between variables and events beyond the highly controlled setting given by randomized controlled trials. In medicine, with the increase of available observational data, e.g., from Electronic Health Records (EHR), it has become crucial to profit from this wealth of information to identify possible causal factors that affect target variables and outcomes.

The continuous improvement, greater availability, and decreasing cost of molecular profiling opened new avenues for the discovery of disease mechanisms via causal inference, beyond association analysis [31, 32]. Understanding the causal relationships in a disease process is an actively growing research front. With the recent technological and computational advances, it became now possible to infer causality in biological processes from observational data, besides data obtained from randomized controlled trials, used in causal inference traditional approaches, often difficult to implement due, e.g., to costs, sample size, and ethical issues. In this short survey, causal inference methods, their characteristics, and applications across several biomedical domains are covered.

Pioneering work in causal discovery is rooted in Bayesian Network (BN) theory [33], in which a BN takes the form of a directed acyclic graph (DAG) together with a collection of conditional probability tables. The rationale of this approach is to factorize a joint probability distribution by using only one vertex parent.

The last years assisted in significant growth of research in causal inference algorithms. Depending on the type of method, they can be generally classified into constraint-based or score-based; some methods can also accept hidden or latent variables or confounders, and others take into account inherent causal models. For comprehensive reviews on causal discovery algorithms and the challenges posed in the causal and sampling processes to generate the observed data, readers may refer to, e.g., [30, 32, 34]. The following sections will briefly overview recent algorithms and corresponding applications to the biomedical area.

3.2.3.1 Constraint-Based Methods

Constraint-based methods use (conditional) independence tests to evaluate the existence of edges between each pair of variables [35]. They first learn an undirected skeleton using repeated conditional independent tests and then assign edge directions by solving directional constraints on this structure.

One of the first methods proposed was the PC algorithm [36], which infers a DAG from observational data (in the absence of hidden variables or latent confounders). Some variants such as the PC-simple [37] and PC-stable [38] were also developed which are computationally feasible even for a high number of variables.

The FCI algorithm [36, 39] is another constraint-based method but that now allows latent variables. It has already been applied to microarray data [40], along with its variant FCI+ [41]. Its polynomial version for sparse graphs, RFCI [42], also accounts for hidden and selection variables.

3.2.3.2 Score-Based Methods

Score-based methods use a score to measure the goodness-of-fit of different graphs, and use a search procedure to find the best one [35].

The Greedy Equivalent Search (GES) algorithm [43] uses a score-equivalent and decomposable score, such as the Bayesian information criterion (BIC) score, and greedily searches the space of DAGs, by sequentially adding edges that maximize the score. The Greedy Interventional Equivalence Search (GIES) algorithm [44] is a generalization of GES to the case where a mix of observational and interventional data is available. An exact version that uses dynamic programming was also formulated [45] and implemented in the package `pcalg` (see Table 3.2 for more details).

3.2.3.3 Hybrid Methods

The combination of constraint-based and score-based methods gave rise to hybrid methods. For example, the Greedy Fast Causal Inference (GFCI) [46] uses a combination of GES and FCI, where GES is applied to find a graph skeleton, and FCI is used as a post-processor for GES to remove the extra adjacencies, and correct the orientations in the output of GES.

The Sparsest Permutation (SP) algorithm [47] also uses conditional independence but then ranks the DAGs based on the number of edges, under a sparsity assumption.

The Greedy Sparsest Permutation (GSP), the greedy version of SP [48], performs a search over orderings of the genes to find the sparsest causal network that best fits the data, and it has been successfully applied to single-cell gene expression data before [49]. GSP [48] was recently incorporated in a computational platform to identify druggable protein targets SARS-CoV-2, integrating multiple available data modalities, e.g., transcriptomic, proteomic, and structural, with a principled causal framework [49]. Inferring causal networks is a valuable tool for drug repositioning in precision medicine to reduce the high costs associated with developing new drugs, by using existing drugs on novel targets in drug-target networks. GSP's performance was compared against PC [36] and GES [43].

Squires et al. (2020) introduced the Unknown-Target IGSP (UT-IGSP), an interventional adaptation of the Greedy Sparsest Permutation (GSP) algorithm [48], based on a new scoring function making use of the interventional data without requiring knowledge of the intervention targets. This method is implemented in the `causaldag` package [50]. Bernstein et al. (2020) proposed the sparsest poset (SPo), a hybrid method for causal structure discovery in the presence of latent confounders that uses both a scoring criterion and conditional independence testing to learn the model [51]. The authors also introduced a greedy search over the space of posets, Greedy Sparsest Poset (GSPo).

3.2.3.4 Covariate Adjustment Methods

The intervention calculus when the DAG is absent (IDA) method [52] predicts bounds for causal effects from observational data alone and without hidden variables. IDA has been used to estimate causal effects of microRNA on RNA expression of genes in cancer cell lines from epithelial-to-mesenchymal transition (EMT) datasets [53], and also to predict single gene deletions in *Saccharomyces cerevisiae*, by using whole-genome expression profiles (observational data) together with a collection of mutants information (interventional data) [54]. Notably, PC and IDA algorithms were also applied to TCGA metastatic melanoma data [55].

The Joint-IDA [56] algorithm, a generalization of the IDA method to estimate the effect of multiple simultaneous interventions, e.g., multiple gene knockouts, was also applied successfully in reverse engineering of gene regulatory networks associated with the DREAM4 In Silico Network Challenge.

Causal Stability Ranking (CStaR) [57] combines the IDA algorithm with stability selection [58], and was applied to study *Arabidopsis thaliana* flowering time, and also to analyze *Saccharomyces cerevisiae* gene expression datasets.

3.2.3.5 Difference Causal Inference

Another direction in causal inference methods has been the disclosure of the differences between different estimated graphs. Under this topic, Belyaeva et al. (2020) [59] devised the Difference Causal Inference (DCI) method to address the challenge of efficiently estimating differences in gene regulatory networks (i.e., edges that appeared, disappeared, and changed weight) in different conditions (e.g., disease states and cell types), without the need for estimating each individual graph separately. The algorithm performance was demonstrated on bulk and single-cell gene expression datasets.

3.2.3.6 Structural Equation Models (SEM)

In Structural Equation Models (SEM) or Functional Causal Models (FCM), the value of each variable is a function of its direct causes and the unmeasured disturbances [34]). Some examples include the Linear non-Gaussian acyclic model (LiNGAM) [60], the nonlinear additive noise (ANM) models [61], and the post-nonlinear (PNL) causal model [62].

Invariant Causal Prediction (ICP) and hiddenICP (allowing for hidden variables) methods [58, 63] are based on SEM and an invariance principle, and are able to deal with data from both observational and perturbation settings, with no need to specify the nature of an experimental setting.

3.2.3.7 Mendelian Randomization

Principle Mendelian Randomization-based methods adopt the Principle of Mendelian Randomization (PMR), under which the genotype only affects the disease status indirectly and is assigned randomly (given the parents' genes) at meiosis, independently of the possible confounding factors [64]. Badsha et al. (2019) [65] introduced the MRPC, which combines the PMR and machine learning, by incorporating the PMR into PC algorithms. MRPC was developed to learn causal networks from genotype and molecular phenotype data. Its performance was evaluated for Lymphoblastoid Cell Lines (LCLs). The suitability of MR and other methods (e.g., SEM and Bayesian Networks) for causal inference in genome-wide association studies (GWAS) in multi-omic settings has been investigated by GAW20 Causal Modeling Working Group [66].

3.2.3.8 Causality in Time Series

The problem of estimating causal processes can also be formulated in the time series domain. Many biomedical problems can be described by multivariate time series data, e.g., RNA expression series in patients followed for a period of time.

Inferring temporal relationships has been approached via Dynamic Bayesian Networks (DBN) [67], by extending BNs to account for temporal information. DBNs have been applied in the identification of gene regulatory networks from time-course microarray data [68].

Gong et al. (2015, 2017) [69, 70] addressed the challenge of estimating temporal causal relationships when the data sampling frequency differs from the true causal frequency or data are temporarily aggregated, to overcome limitations of Granger Causality [71, 72] methods, which usually assume that the sampling frequency matches the true frequency of the underlying causal process, and are therefore sensitive to subsampling or temporal aggregation. The authors show that under certain mild conditions on the structure of the causal relations where the noise terms are non-Gaussian, the causal relations are identifiable [70].

To overcome the limitation of Granger Causality methods with the increased dimension in the variable space compared to the sample size, variable selection has been integrated into Granger Causality models, e.g., a truncating lasso penalty for the estimation of graphical Granger models [73], with the resulting estimate providing information on the time lag between the activation of transcription factors and their effects on regulated genes. Another example is the group lasso graphical Granger model [74], which applies a regression method suited for high-dimensional data by taking advantage of the group structure among the lagged temporal variables according to the specific time series they belong to. The effectiveness of this novel methodology is shown for gene regulatory networks from human cancer cell cycle data.

Yang et al. (2017) [75] developed a framework for nonlinear causal network reconstruction from time series with limited observations. The authors designed the group lasso nonlinear conditional Granger causality (GLasso-NCGC), a method to handle nonlinearity and directionality of complex networked systems. The method was applied to gene regulatory benchmark datasets of the DREAM3 Challenge4.

3.2.3.9 Challenges in Causal Inference and Discovery

Causal Discovery in the presence of measurement error. Saeed et al. (2020) [76] addressed the problem of learning a causal graph in the presence of measurement error, a common problem in genomics, with errors in gene expression arriving from the measurement process, as is the case of the dropout phenomenon in single-cell RNA sequencing experiments. The authors developed a procedure for estimating the causal structure in a linear Gaussian structural equation model under the *Anchored Causal Model*, where each corrupted observed variable is generated from a latent uncorrupted variable. The method was applied to single-cell gene expression data collected from human pancreatic and bone marrow-derived dendritic cells. Also, work by Zhang et al. (2017) [77] concerned with the identifiability conditions for the measurement-error free causal given contaminated observations, for which the causal model is partially or even fully identifiable.

Causal Discovery in the presence of missing data. Missing data are common across many biomedical and healthcare datasets, which are generated, e.g., by anomalies in data acquisition due to technical issues or deficient data collection. Missing data entries might compromise (conditional) independence relations in the complete data generated by the underlying causal process, leading to unreliable conclusions drawn by available causal discovery methods. Tu and co-authors (2019) [78] have recently proposed Missing Value PC (MVPC), an extension to the PC algorithm to recover the underlying causal structure from observed data that are missing under different mechanisms by incorporating additional corrections on erroneous edges. The MVPC method was applied to understand the causal rations in the US Cognition dataset [79], and among various factors and healing outcomes in Achilles Tendon Rupture (ATR) data [80, 81].

3.3 Software and Biomedical Applications

The development of novel algorithms has been accompanied by the implementation of the methods in several software packages, such as in R. Table 3.2 lists some of the most common software described in this survey. Although not exhaustive, it provides references and examples of freely available code that will facilitate future work.

Table 3.2 Overview of software packages for network-based biomedical data analysis

Software	Observations	Ref.
Differential network analysis		
DINGO/iDINGO	Differential network analysis in genomics	[6, 10]
INDEED	Integrated differential expression and differential network analysis	[9]
Network-based learning and visualization		
glmSparseNet	Network centrality metrics for elastic-net regularized models	[82, 83]
glmnet	Lasso and elastic-net regularized generalized linear models	[84]
JGL	Joint graphical lasso	[23]
glasso	Estimation of Gaussian graphical models	[85]
bootnet	Bootstrap methods for various network estimation routines	[86]
grangerTlasso	Truncating lasso for graphical Granger models	[73]
NetworkComparisonTest	Statistical comparison of two networks based on invariance measures	[87]
qgraph	Network visualizations of relationships in psychometric data	[88]
igraph	The igraph software package for complex network research	[89]
Cytoscape	Network data integration, analysis, and visualization in a box	[90]
NAViGaTOR	Network analysis, visualization, and graphing TORonto	[91]
MONGKIE	Integrated network analysis and visualization platform for multi-omics	[92]
GliomaDB	Webserver for integrating glioma omics data and interactive analysis	[93]
Rgraphviz	Provide plotting capabilities for R graph objects	[94]
Gephi	An open-source software for exploring and manipulating networks	[95]
Causal inference		
InvariantCausalPrediction	Invariant causal prediction	[63]
nonlinearICP	Invariant causal prediction for nonlinear models	[96]
seqICP	Sequential invariant causal prediction (for sequential data)	[63, 97]
causalDAG	Creation, manipulation, and learning of causal DAGs	[59]
MendelianRandomization	Mendelian randomization package	[98]
MRPC	PC algorithm with the principle of Mendelian randomization	[65]
pcalg	Methods for graphical models and causal inference	[44, 99]
ParallelPC	Parallelized constraint-based causal discovery algorithms	[100]
bnlearn	Bayesian network learning and inference	[101]
The Tetrad project	CCD causal software	[102]

3.4 Conclusions and Future Work

Describing complex biological interactions is crucial to understand cell functions and disease. Network-based modeling has been pointed as a core technique to model omics data. The review has three pillars, identified as the main aspects of network-based learning. The first is related to the identification of differentially expressed genes using the inherent pathway or gene network structure. The second is related to regularized optimization using the graph of the studied features. Finally, the growing field of causal inference and discovery, with several associated challenges.

Besides overviewing these three main topics in network-based learning for biomedical data, this survey has identified a collection of software packages that support these research topics, along with biological databases that provide key information for network-statistical learning. Precision medicine is expected to significantly benefit from these new methods and algorithms, with an undoubtful impact on clinical decision-making in the future.

Acknowledgements This work was partially supported by national funds through the Portuguese Foundation for Science & Technology (FCT) with references CEECINST/00102/2018, PTDC/CCI-BIO/4180/2020, DSAIPA/DS/0026/2019, UIDB/00297/2020 (CMA), UIDB/04516/2020 (NOVA LINCS), and UIDB/50021/2020 (INESC-ID), and by the European Union's Horizon 2020 research and innovation program under grant agreement No. 951970 (OLISSIPO project).

References

1. Reimand J, Isserlin R, Voisin V, Kucera M, Tannus-Lopes C, Rostamianfar A, Wadi L, Meyer M, Wong J, Xu C, Merico D, Bader G (2019) Nat Protoc 14:482
2. Hawe J, Theis F, Heinig M (2019) Front Genet 10:55
3. Manzoni C, Kia D, Vandrovcova J, Hardy J, Wood N, Lewis P, Ferrari R (2018) Brief Bioinform 19(2):286
4. Singer J, Irmisch A, Ruscheweyh HJ, Singer F, Toussaint N, Levesque M, Stekhoven D, Beerenwinkel N (2019) Brief Bioinform 20(3):778
5. Zhang W, Chien J, Yong J, Kuang R (2017) npj Precis Oncol 1(1):25. https://doi.org/10.1038/s41698-017-0029-7
6. Ha M, Baladandayuthapani V, Do KA (2015) Bioinformatics 31(21):3413
7. Fuente A (2010) Trends Genet 26(7):326
8. Zhang XF, Ou-Yang L, Zhao XM, Yan H (2016) Nat Sci Rep 6:34112
9. Zuo Y, Cui Y, Poto C, Varghese R, Yu G, Li R, Ressom H (2016) Methods 111:12
10. Class C, Ha M, Baladandayuthapani V, Do KA (2018) Bioinformatics 34(7):1243
11. Kim Y, Hao J, Mersha YGT, Kang M (2018) Int J Data Min Bioinform 20(4):362
12. Hastie T, Tibshirani R, Wainwright M (2015) Statistical learning with sparsity: the lasso and generalizations. Chapman & Hall/CRC
13. Tibshirani R (1996) J R Stat Soc Ser B 58(1):267
14. Zou H, Hastie T (2005) J Roy Stat Soc Ser B 67(2):301
15. Yuan M, Lin Y (2006) J R Stat Soc Ser B 68(1):49
16. Hoefling H (2010) J Comput Graph Stat 19(4):984

17. Tibshirani R, Saunders M, Rosset S, Zhu J, Knight K (2005) J R Stat Soc Ser B 67:91
18. Vinga S (2020) Brief Bioinform. https://doi.org/10.1093/bib/bbaa122
19. Chung F (1997) Spectral graph theory, vol 92. American mathematical society
20. Friedman J, Hastie T, Tibshirani R (2008) Biostatistics 9:432
21. Li L, Li H (2008) Bioinformatics 24(8):1175
22. Zhang W, Ota T, Shridhar V, Chien J, Wu B, Kuang R (2013) PLoS Comput Biol 9:e1002975
23. Danaher P, Wang P, Witten D (2014) J R Stat Soc Ser B (Stat Methodol) 76(2):373
24. Xie L, Varathan P, Nho K, Saykin A, Salama P, Yan J (2020) PLoS One 15(6):e0234748
25. Veríssimo A, Oliveira A, Sagot MF, Vinga S (2016) J R Stat Soc Ser B 77:449
26. Lopes M, Casimiro S, Vinga S (2019) BMC Bioinform 20(1):356
27. Lopes M, Vinga S (2020) BMC Bioinform 21:59
28. Peixoto C, Martins MLM, Costa L, Vinga S (2020) Biomedicines 8:488
29. Li Y, Mark B, Raskutti G, Willett R, Song H, Neiman D (2019). arXiv:1803.07658v3
30. Kleinberg S, Hripcsak G (2011) J Biomed Inform 44(6):1102
31. Schölkopf B (2020). arXiv:1911.10500v2
32. Bühlmann P (2018) Invariance, causality and robustness
33. Pearl J (2009) Causalaity. Cambridge University Press, Cambridge
34. Glymour KZC, Spirtes P (2019) Front Genet 10:254
35. Yu Y, Chen J, Gao T, Yu M (2020). arXiv:1904.10098v1
36. Spirtes P, Glymour C, Scheines R (2000) 2nd edn. MIT Press, Cambridge, MA
37. Bühlmann P, Kalisch M, Maathuis M (2010) Biometrika 97(2):261
38. Colombo D, Maathuis M (2014) J Mach Learn Res 15:3921
39. Spirtes P, Meek C, Richardson T (1995) Proceedings of the Eleventh conference on Uncertainty in artificial intelligence (UAI) pp. 499–506
40. Spirtes P, Glymour C, Scheines R, Kauffman S, Aimale V, Wimberly F (2000)
41. Claassen T, Mooij JM, Heskes T (2013) Proceedings of the twenty-ninth conference on uncertainty in artificial intelligence. AUAI Press, Arlington, Virginia, USA. UAI'13, pp 172–181
42. Colombo D, Maathuis M, Kalisch M, Richardson T (2012) Ann Stat 40(1):294
43. Chickering D (2003) J Mach Learn Res 3:507
44. Hauser A, Bühlmann P (2012) J Mach Learn Res 13:2409
45. Silander T, Myllymäki P (2006) Proceedings of the twenty-second conference on uncertainty in artificial intelligence, AUAI Press, Arlington, Virginia, USA. UAI'06, pp 445–452
46. Ogarrio J, Spirtes P, Ramsey J (2016) Proceedings of the eighth international conference on probabilistic graphical models 52:368
47. Raskutti G, Wang Y, Uhler C (2019) Learning directed acyclic graphs based on sparsest permutations
48. Solus L, Wang Y, Uhler C (2020) Consistency guarantees for greedy permutation-based causal inference algorithms
49. Belyaeva A, Cammarata L, Radhakrishnan A, Squires C, Yang K, Shivashankar G, Uhler C (2020) Causal network models of sars-cov-2 expression and aging to identify candidates for drug repurposing
50. Squires C, Wang Y, Uhler C (2020) (PMLR, Virtual, 2020). Proceedings of machine learning research 124:1039–1048
51. Bernstein D, Saeed B, Squires C, Uhler C (2020) (PMLR, 2020). Proceedings of machine learning research 108:4098–4108
52. Maathuis M, Kalisch M, Bühlmann P (2009) Ann Stat 37:3133
53. Le T, Liu L, Tsykin A, Goodall G, Liu B, Sun BY, Li J (2013) J Mach Learn Res 29(6):765
54. Maathuis M, Colombo D, Kalisch M, Bühlmann P (2010) Nat Methods 7:247
55. Maddox W (2017) Discussions 13(1):329
56. Nandy P, Maathuis M, Richardson T (2017) Ann Stat 45(2):647
57. Stekhoven D, Moraes I, Sveinbjörnsson G, Hennig L, Maathuis M, Bühlmann P (2012) Bioinformatics 28(21):2819
58. Meinshausen N, Bühlmann P (2010) J R Stat Soc Ser B 72:417

59. Belyaeva A, Squires C, Uhler C (2020) bioRxiv. https://doi.org/10.1101/2020.05.13.093765
60. shimizu S, Hoyer P, Hyvärinen A, Kerminen A (2006) J Mach Learn Res 7:2003
61. Hoyer P, Janzing D, Mooij JM, Peters J, Schölkopf B (2009) Advances in neural information processing systems. In: Koller D, Schuurmans D, Bengio Y, Bottou L (eds) vol 21. Curran Associates, Inc, pp 689–696
62. Zhang K, Hyvärinen A (2009) Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence. AUAI Press, pp 647–655
63. Peters J, Meinshausen PBN (2016) J R Stat Soc Ser B 785:947
64. Didelez V, Sheehan N (2007) Stat Methods Med Res 16:309
65. Badsha B, Fu A (2019) Front Genet 10
66. Auerbach J, Howey R, Jiang L, Justice A, Li L, Oualkacha K, Sayols-Baixeras S, Aslibekyan S (2018) BMC Genet 19(Suppl 1):74
67. Dean Kanazawa (1989) Comput Intell 5(3):142
68. Zou M, Conzen S (2005) Bioinformatics 21(1):71
69. Gong M, Zhang K, Schölkopf B, Glymour C, Tao D (2017) Proceedings conference on uncertainty in artificial intelligence (UAI) 2017. Association for Uncertainty in Artificial Intelligence (AUAI), p. ID 269
70. Gong M, Zhang K, Schölkopf B, Tao D, Geiger DP (2015) Proceedings of the 32th international conference on machine learning (ICML 2015) 37:1898
71. Granger C (1980) J Econ Dyn Control 2:329
72. Granger C (1969) Econometrica 373:424
73. Shojaie A, Michailidis G (2010) Bioinformatics 26:i517
74. Lozano A, Abe N, Liu Y, Rosset S (2009) Bioinformatics 25:i110
75. Yang G, Wang L, Wang X (2017) Sci Rep 7:2991
76. Saeed B, Belyaeva A, Wang Y, Uhler C (2020) (PMLR, Virtual, 2020). Proc Mach Learn Res 124:619–628
77. Zhang K, Gong M, Ramsey J, Batmanghelich K, Spirtes P, Glymour C (2017) UAI 2917 workshop on causality: lenarning, inference, and decision-making
78. Tu R, Zhang C, Ackermann P, Mohan K, Kjellström H, Zhang K (2019) Proceedings of the 22nd international conference on artificial intelligence and statistics (AISTATS)
79. Strobl V, Visweswaran S, Spirtes P (2017) Int J Data Sci Anal 6:47
80. Praxitelous P, Edman G, Ackermann P (2018) Scand J Med Sci Sports 28(1):294
81. Hamesse C, Tu R, Ackermann P, Kjellström H, Zhang C (2019) (PMLR, 2019). Proc Mach Learn Res 106:614–640
82. Verissimo A, Carrasquinha E, Lopes M, Oliveira A, Sagot MF, Vinga S (2018) bioRxiv
83. Verissimo A, Vinga S, Carrasquinha E, Lopes M (2018) Bioconductor R package version 3.11
84. Friedman J, Hastie T, Tibshirani R, Narasimhan B, Tay K, Simon N (2020) R package version 4.0-2
85. Friedman J, Hastie T, Tibshirani R (2019) R package version 1:11
86. Epskamp S, Borsboom D, Fried E (2017) Behavior research methods
87. van Borkulo CD, Boschloo L, Kossakowski JJ, Tio P, Schoevers RA, Borsboom D, Waldorp LJ (2017) J Stat Softw
88. Epskamp S, Cramer AOJ, Waldorp LJ, Schmittmann VD, Borsboom D (2012) J Stat Softw 48(4):1
89. Csardi G, Nepusz T (2006) InterJ Complex Syst, p 1695
90. Shannon P, Markiel A, Ozier O, Baliga N, Wang J, Ramage D, Amin N, Schwikowski B, Ideker T (2003) Genome Res 13(11):2498
91. McGuffin M, Xie W, Devani B, van Toch I, Jurisica I (2009) Bioinformatics 25(24):3327
92. Jang Y, Yu N, Seo J, Kim S, Lee S (2016) Biol Direct 11:10
93. Yang Y, Sui Y, Xie B, Qu H, Fang X (2019) Genomics Proteomics Bioinform 17(4):465
94. amd KH, Gentry J, Long L, Gentleman R, Falcon S, Hahne F, Sarkar D (2020) R package version 2.34.0
95. Bastian M, Heymann S, Jacomy M (2009)
96. Heinze-Deml C, Peters J, Meinshausen N (2018) J Causal Inference 6:20170016

97. Pfister N, Bühlmann P, Peters J (2018) J Am Stat Assoc 114(527):1264
98. Yavorska O, Burgess S (2017) Int J Epidemiol 46(6):1734
99. Kalisch M, Mächler M, Colombo D, Maathuis M, Bühlmann P (2012) J Stat Softw 47(11):1
100. Le T, Hoang T, Li J, Liu L, Hu S (2000) J Mach Learn Res, pp 1–48
101. Scutari M (2010) J Stat Softw 35(3):1
102. Spirtes P, Glymour C, Scheines R (1998) Multivar Behav Res 31(1):65

Chapter 4
Simultaneous Clustering of Multiple Gene Expression Datasets for Pattern Discovery

Basel Abu-Jamous and Asoke K. Nandi

Background Healthy cells run sophisticated genetic programmes in order to carry out their various biological processes such as cellular growth, cell division, stress response, and metabolism. The regulation of these genetic programmes is realised at different levels by controlling the production of the required types of large biomolecules such as RNAs, proteins, glycans, and lipids, with different amounts, at different times, and in different sub-cellular locations. Although all cells in an organism, such as skin cells, liver cells, bone cells, and neurons nominally have the same genomic material, they differ in shape and function because of the differences in the genetic programmes that they run.

A better understanding of genetic programmes has the potential not only to improve our biological knowledge but can also lead to breakthroughs in various applications. For instance, understanding how cancerous cells deviate from nearby healthy cells in their genetic programmes can help in discovering malignancy biomarkers and identifying potential drug targets [10, 59, 65]. Beyond human health, understanding the genetics regulating the molecular mechanisms in trees and crops may help develop the next generation of plants that are more productive, more efficient in resource usage, and more resilient to environmental stress [27, 36]. Furthermore, the genetic engineering of the molecular pathways in some microorganisms may render them industrial factories of the future that will produce widely used chemical compounds [39].

As science has entered the era of data-driven discoveries, plenty of technologies are being developed to capture different views of cells and tissues. Amongst the most widely used are those that quantify the expression levels of genes in cells, namely

B. Abu-Jamous
Sensyne Health plc, Oxford, UK

Phi Science, Amman, Jordan

A. K. Nandi (✉)
Electronic and Electrical Engineering, Brunel University London, Uxbridge, UK
e-mail: Asoke.Nandi@brunel.ac.uk

T. D. Pham et al. (eds.), *Advances in Artificial Intelligence, Computation, and Data Science*, Computational Biology 31,
https://doi.org/10.1007/978-3-030-69951-2_4

the older RNA microarray technology and the more recent RNA next-generation sequencing (NGS) technology. Millions of samples linked to hundreds of thousands of gene expression datasets have become available in public repositories and in private laboratories taken from a plethora of biological conditions and a large variety of species. Publicly available repositories of gene expression data include the Gene Expression Omnibus (GEO) [30] and ArrayExpress [9].

In many cases, a dataset is generated by a research laboratory to address a specific research question and is analysed accordingly. However, this approach does not usually utilise the full capacity of that dataset, and therefore it is made available in a public repository for other researchers to exploit. Consequently, within a given biological context (e.g., a disease), many datasets have been generated from different laboratories, in different years, using different technologies, and with different specifications. Thus, not only is each one of these datasets likely to be underexploited but also the synergetic value from the exploration of these datasets collectively has been missed.

With this mine of valuable data available, the challenge is to develop and apply a new generation of computational methods that have the ability to explore multiple gene expression datasets simultaneously in order to arrive at discoveries that would not have been otherwise feasible. This ambition requires overcoming a number of challenges including dealing with heterogeneous datasets that have different structures and dissimilar numerical features as well as the correct consideration of the biological semantics of these different datasets, their relation to the research question being addressed, and the assumptions made by the designed algorithm [2]. Several methods have already been designed with this aim, such as the *binarisation of consensus partition matrices (Bi-CoPaM)* [6, 7], *clust* [5], and cluster of clusters (COCA) [17]. Nonetheless, further algorithmic developments are expected to emerge to expand the types of datasets simultaneously analysed and the quality and relevance of the results obtained by the application of these algorithms.

In this chapter, we review the state-of-the-art methods in the context of simultaneous analysis of multiple gene expression datasets and then we present three case studies that demonstrate the potential of this approach in a microorganism, namely budding yeast, in humans, and in rotifer animals.

4.1 Simultaneous Clustering Methods

Clustering methods, or generally machine learning methods, that are designed to analyse multiple datasets simultaneously can be ascribed to one of three broad classes based on the stage at which integration takes place, namely early, intermediate, and late integration. A list of some clustering methods is shown in Table 4.1.

Early integration approaches first concatenate individual datasets to form a single dataset with a higher feature space. Later clustering is applied to the concatenated dataset using traditional methods such as *k*-means [54], hierarchical clustering [31], and self-organising maps (SOMs) [42], or some of the more modern single-dataset

Table 4.1 Clustering methods

Method	Complete/Partial	Early[a]/Intermediate/Late integration	References
k-means	Complete	Early	[54]
Hierarchical clustering (HC)	Complete	Early	[31]
Markov clustering (MCL)	Partial	Early	[34]
WGCNA	Partial	Early	[48]
Cluster of clusters (COCA)	Complete	Late	[74]
Bi-CoPaM	Partial	Late	[6]
UNCLES	Partial	Late	[4]
Cross clustering (CC)	Partial	Early	[73]
Clust	Partial	Late	[5]

[a] Early integration methods are in reality methods that were not designed to perform integrative clustering; rather they were designed to perform clustering on a single dataset. Thus, early integration refers to the application of such methods to a single dataset compiled by the direct concatenation of the individual datasets involved

clustering methods such as weighted correlation network analysis (WGCNA) [48] and Markov clustering [34]. Although this approach may seem intuitive, it requires homogeneity across datasets for a valid concatenation. Standardisation may be applied to raw datasets to enable early integration. However, if there was a substantial imbalance across the datasets in their statistical distributions, dimensions of feature spaces, or sparsity, early integration approaches become invalid.

Intermediate integration approaches treat different datasets as different views of the clusters to be produced from the data. Therefore, datasets with heterogeneous feature spaces are transformed into a homogeneous feature space before being concatenated to form a single dataset. Then, clustering is applied to that single concatenated dataset. Although these approaches allow for the integration of heterogeneous datasets, the feature-space transformation step reduces the interpretability of the produced clusters in the context of their original features.

Late integration approaches apply clustering to each dataset independently before integrating these individual results into a single consensus result. This allows for the simultaneous analysis of multiple heterogeneous datasets without requiring their feature spaces to be converted to a homogeneous feature space, as the latter is not always trivial or even feasible. Examples of late integration methods include Bi-CoPaM [6], *clust* [5], and COCA [17, 74].

Clustering methods can be further classified into complete, fuzzy, and partial methods. This classification applies to clustering methods that are applicable to a single dataset or to multiple datasets. Complete clustering methods are those that partition the entire set of input genes into clusters where every gene is assigned to one and only one cluster. On the other hand, fuzzy clustering assigns each gene to each cluster with a fuzzy membership value ranging from zero to unity, where the membership values of a given gene in all clusters must sum up to unity. As for partial clustering methods, they do not force every input gene to be included in one of the clusters; they rather allow some genes to be unassigned to any of the clusters. We can also refer to complete clustering as data partitioning and to partial clustering as cluster extraction.

Partial clustering has been increasingly gaining importance in gene expression clustering because it meets the biological expectation that many genes would be irrelevant to the context of the dataset(s) being analysed or at least will not form groups of co-regulated genes, and therefore should not be included in clusters. Examples of partial clustering methods include, in addition to Bi-CoPaM and *clust*, Cross Clustering [73], Markov Clustering [34], and WGCNA [48].

4.1.1 Cluster of Clusters (COCA)

The COCA algorithm was first proposed as part of a comprehensive analysis of breast cancer data from the TCGA dataset by the Cancer Genome Atlas Network [74], and then became a popular tool in multi-dataset cancer analysis [17]. This method has been applied to multiple gene expression datasets as well as to datasets from different omic types.

Given a set of datasets $\{X_1, \ldots, X_L\}$ measuring the gene expression profiles of the same n genes but over different sets of conditions, and given one or more base clustering methods $\{f_1, \ldots, f_C\}$ such as k-means, COCA is applied by the following steps:

1. Apply each one of the base clustering methods to each one of the datasets and optimise the number of clusters separately for each case using a user-chosen cluster validation index (e.g. silhouette):

$$U_{l,c} = \text{optimise}_{k \in \mathbb{K}}(f_c(X_l, k)), \tag{4.1}$$

   where $\mathbb{K}$ is the set of k values to be tested and the optimise $(\cdot)$ function identifies the k value yielding the optimum result as per the adopted cluster validation index. $U_{l,c}$ is the clustering result of the lth dataset using the cth clustering method with the optimum k value that is denoted as $\hat{k}_{l,c}$. The clustering result $U_{l,c}$ is formatted as a partition matrix with rows representing $\hat{k}_{l,c}$ clusters, columns representing n genes, and elements of one or zero values indicating if the corresponding gene is assigned to the corresponding cluster or not, respectively. The result of this step is the generation of $R = L \times C$ partition matrices for the same set of n genes but with varying numbers of clusters.
2. Concatenate all R partition matrices to obtain one super-partition matrix $\mathcal{U}$. The matrix $\mathcal{U}$ will have n columns while the number of its rows will be the summation of the numbers of clusters in the individual partition matrices:

$$\mathcal{U} = \begin{bmatrix} U_{1,1} \\ U_{1,2} \\ \dots \\ U_{L,C} \end{bmatrix} \in \{0, 1\}^{\Bbbk \times n},$$

where

$$\Bbbk = \sum_{\forall l \forall c} \hat{k}_{l,c}. \tag{4.2}$$

3. Apply hierarchical clustering to the genes in $\mathcal{U}$ to obtain the final set of clusters.

The COCA algorithm is implemented in R as the *coca* package in the Comprehensive R Archive Network (CRAN) [17].

4.1.2 Bi-CoPaM

Given a set of datasets $\{X_1, \dots, X_L\}$, where each one has n genes with expression measured over different numbers of conditions or time-points, the Bi-CoPaM method is applied by the following steps [6]:

1. A selected set of base clustering methods $\{f_1, \dots, f_C\}$, e.g. k-means, HC, and SOMs, are applied to each one of the L datasets to produce $R = L \times C$ clustering results $\left\{U^1, \dots, U^R\right\}$.

   - Each result is formatted as a partition matrix U with rows representing k clusters and columns representing n genes.
   - $U = \left\{u_{i,j} \in \{0, 1\} | i \in \{1, \dots, k\}; j \in \{1, \dots, n\}\right\}$, where each element represents belongingness of the corresponding jth gene in the corresponding ith cluster.
   - The number of clusters k must be set by the user, either based on a priori knowledge or by systematic optimisation.

2. The partition matrices are aligned so that the ith cluster in each partition matrix corresponds to the ith cluster in each other partition matrix using the min–min approach explained in [6].

   - This realignment transforms each partition matrix U to an aligned partition matrix $\hat{U}$.

3. A consensus partition matrix (CoPaM), U^*, is generated by averaging all aligned partition matrices:

$$U^* = \frac{1}{R} \sum_{r \in \{1 \dots R\}} \hat{U}^r \tag{4.3}$$

- The $CoPaM$ matrix is a fuzzy partition matrix with elements ranging from zero to unity, in proportion to the number of individual partition matrices that assigned the corresponding gene to the corresponding cluster.

4. The fuzzy $CoPaM$ matrix is made binary by the *difference threshold binarisation (DTB)* technique to produce a final binary partition matrix B:

   - DTB assigns a gene to a cluster if its membership in that cluster in the CoPaM matrix U^* is greater than its membership in all other clusters, and the margin between this membership and its membership in the closest competitor cluster is not smaller than the tuning parameter $\delta \in [0.0, 1.0]$. Otherwise, the gene is not assigned to any cluster.

$$B = \{b_{i,j}\}; b_{i,j} = \begin{cases} 1, \forall h \neq i, \left(u^*_{i,j} - u^*_{h,j}\right) \geq \delta \\ 0, \text{ otherwise} \end{cases} \tag{4.4}$$

   - The tuning parameter δ is set by the user; if it set to zero, each gene is assigned to the cluster in which it has its maximum membership. Ties where the gene is equally assigned to multiple clusters are solved by assigning the gene to each of those clusters. If δ is set to unity, a gene is only assigned to a cluster if its fuzzy membership value in it is unity, that is, if all individual clustering results unanimously assigned it to that cluster. Therefore, $\delta = 0.0$ produces the loosest clusters, $\delta = 1.0$ produces the tightest clusters, and δ values in between these two extremes produce clusters of a corresponding tightness level.
   - Other binarisation techniques beyond DTB can be found in [6].

4.1.3 UNCLES and M–N Scatter Plots

The UNCLES method is based on the Bi-CoPaM method but extends it in two main directions [4]:

1. UNCLES has multiple modes of application allowing the researcher not only to discover clusters of genes that are co-clustered in all given datasets, but also to discover clusters of genes that are co-clustered in a given subset of datasets while being poorly co-clustered in another subset of datasets. This option is beyond the scope of this chapter but is detailed in [4].
2. The Bi-CoPaM method requires the number of clusters (k) to be specified by the researcher as well as the tuning parameter δ. The UNCLES method solves this problem by defining an algorithmic framework that incorporates Bi-CoPaM with the proposed M–N scatter plots technique in order to automate the selection of the number of clusters.

Furthermore, a modification was proposed to the UNCLES method in [1] to allow for the simultaneous analysis of gene expression datasets that do not necessarily include the exactly same set of genes.

Let $\mathbb{X} = \{X_1, \ldots, X_L\}$ be the set of L gene expression datasets, let $\mathbb{F} = \{f_1, \ldots, f_C\}$ be the set of base clustering methods (e.g. k-means, HC, and SOMs), and let $\mathbb{G}^{k,\delta} = \text{BiCoPaM}(\mathbb{X}, \mathbb{F}, k, \delta)$ be the set of k clusters $\mathbb{G}^{k,\delta} = \left\{G_1^{k,\delta}, \ldots, G_k^{k,\delta}\right\}$ generated by applying Bi-CoPaM to the $\mathbb{X}$ datasets, using the $\mathbb{F}$ base clustering methods, and with the pre-set number of clusters k and the Bi-CoPaM tuning parameter δ. Each cluster $G_i^{k,\delta} = \{g_1, \ldots, g_{n^i}\}$ is represented by the set of n^i genes belonging to it. Therefore, the steps of applying UNCLES incorporating Bi-CoPaM and the M–N scatter plots are

1. For each k value in a set of k values to test (e.g. $k \in \mathbb{K} = \{2, 4, \ldots, 18, 20\}$), and for each δ value in a set of δ values (e.g., $\delta \in \mathbb{D}\{0.0, 0.1, \ldots, 1.0\}$), apply the Bi-CoPaM method to generate a set of clusters:

$$\begin{aligned} &\text{For } k \in \mathbb{K} \\ &\quad \text{For } \delta \in \mathbb{D} \\ &\qquad \mathbb{G}^{k,\delta} = \text{BiCoPaM}(\mathbb{X}, \mathbb{F}, k, \delta) \end{aligned} \tag{4.5}$$

2. Merge all individual clusters generated by the step above into a single set of clusters ($\mathbb{G}^{\forall}$):

$$\mathbb{G}^{\forall} = \text{union}\left(\left\{\mathbb{G}^{k,\delta} | \forall k \in \mathbb{K}; \forall \delta \in \mathbb{D}\right\}\right), \tag{4.6}$$

   where $\mathbb{G}^{\forall} = \{G_1, \ldots, G_{k^{\forall}}\}$,

   where $k^{\forall} = |\mathbb{D}| \sum \mathbb{K}$ is the total number of all clusters generated by all individual Bi-CoPaM experiments.

3. Drop empty clusters from $\mathbb{G}^{\forall}$:

$$\mathbb{G}^{\forall} \leftarrow \mathbb{G}^{\forall} - \left\{G_i | n^i = 0\right\} \tag{4.7}$$

4. Calculate these two metrics for each remaining cluster G_i in $\mathbb{G}^{\forall}$:

   - The scaled average normalised mean-squared error (MSE):

$$M_i = \frac{MSE(G_i) - \min\limits_{\forall G_j \in \mathbb{G}^{\forall}} MSE\left(G_j\right)}{\max\limits_{\forall G_j \in \mathbb{G}^{\forall}} MSE\left(G_j\right) - \min\limits_{\forall G_j \in \mathbb{G}^{\forall}} MSE\left(G_j\right)} \in [0.0, 1.0], \tag{4.8}$$

   Where

$$MSE(G_i) = \frac{1}{L \cdot n^i \cdot d^i} \sum_{l \in \{1 \ldots L\}} \sum_{g_j \in G_i} ||x_{l,g} - z_{l,G_i}||^2, \tag{4.9}$$

where n^i is the number of genes in the cluster G_i, d^i is the number of dimensions in that cluster, $x_{l,g}$ is the expression profile of the gene g in the lth dataset, z_{l,G_i} is the average profile of all genes in the cluster G_i in the lth dataset, and $|| \cdot ||^2$ is the second norm.

- The scaled logarithm of the cluster size:

$$N_i = \frac{\log n^i - \min\limits_{\forall j \in \{1 \ldots k^\forall\}} \log n^j}{\max\limits_{\forall j \in \{1 \ldots k^\forall\}} \log n^j - \min\limits_{\forall j \in \{1 \ldots k^\forall\}} \log n^j} \in [0.0, 1.0]. \tag{4.10}$$

5. Construct a unit-square scatter plot for all clusters where the horizontal axis represents the cluster's dispersion as measured by M and the vertical axis represents the cluster's size as measured by N, hence the name "M–N scatter plot".
6. Initiate an empty ordered set of final clusters:

$$\mathbb{G}^{\text{final}} = \emptyset \tag{4.11}$$

7. With the aim of minimising dispersion while maximising cluster size, the optimum point on the M–N scatter plot is its top-left corner where $M_i = 0.0$ and $N_i = 1.0$. Therefore, measure the Euclidean distance between each cluster's point on the scatter plot and the top-left corner:

$$d_i = \sqrt{M_i^2 + (1 - N_i)^2}. \tag{4.12}$$

Here, $d_i \in \left[0.0, \sqrt{2}\right]$ is called the *M–N distance* for the ith cluster and is a metric for the quality of the cluster where smaller values indicate better clusters.

8. Amongst the clusters in $\mathbb{G}^\forall$, select the cluster with the smallest M–N distance (d) and append it to the end of the ordered list of final clusters:

$$G^{\text{best}} = \underset{\forall i \in \{1 \ldots |\mathbb{G}^\forall|\}}{\text{argmin}} (d_i),$$
$$\mathbb{G}^{\text{final}} \leftarrow \mathbb{G}^{\text{final}} \cup \left(G^{\text{best}}\right) \tag{4.13}$$

9. Remove all clusters that overlap with G^{best} from the set $\mathbb{G}^\forall$:

$$\mathbb{G}^\forall \leftarrow \mathbb{G}^\forall - \left\{G_i | G_i \cap G^{\text{best}} \neq \emptyset\right\} \tag{4.14}$$

10. If there are clusters remaining in $\mathbb{G}^{\forall}$, that is $\mathbb{G}^{\forall} \neq \emptyset$, go back to step 8 above; otherwise, terminate the process.

The outcome of the process above is an ordered list of non-overlapping clusters alongside their d values, where smaller values of these distances indicate "better" clusters. The researcher may choose to proceed only with the top few clusters in downstream analysis as these clusters are the largest in size with the tightest profiles.

4.1.4 Clust

Clust represents further development on top of Bi-CoPaM, UNCLES, and the M–N scatter plots methods [5]. *Clust* contributes by (1) the automation of steps with empirically validated default values, (2) the optimisation and completion of clusters in terms of removing outliers and inclusion of missing genes that have been missed by previous steps. These two points will be clarified as the steps of *clust* are described below. Despite default automation, users of *clust* are given the ability to tune some master parameters if they wish to customise them.

Let $\mathbb{X} = \{X_1, \ldots, X_L\}$ be the set of L gene expression datasets, let $\mathbb{F} = \{f_1, \ldots, f_C\}$ be the set of base clustering methods (e.g. k-means, HC, and SOMs), let $\mathbb{K} = \left\{k_1, \ldots, k_{|\mathbb{K}|}\right\}$ be the set of k values (numbers of clusters) to be considered by UNCLES, and let $\mathbb{D} = \left\{\delta_1, \ldots, \delta_{|\mathbb{D}|}\right\}$ be the set of the binarisation tuning parameters δ to be considered by UNCLES. Redefine the M_i metric of cluster dispersion, which is used by the M–N scatter plots (defined in Eq. 4.8), as

$$M_i = t \times \left(\frac{\mathrm{MSE}(G_i) - \min\limits_{\forall G_j \in \mathbb{G}^{\forall}} \mathrm{MSE}\big(G_j\big)}{\max\limits_{\forall G_j \in \mathbb{G}^{\forall}} \mathrm{MSE}\big(G_j\big) - \min\limits_{\forall G_j \in \mathbb{G}^{\forall}} \mathrm{MSE}\big(G_j\big)} \right) \in [0.0, t], \tag{4.15}$$

where $\mathbb{G}^{\forall}$, G_i, and $\mathrm{MSE}(\cdot)$ are as used in Eq. (4.8) and $t \geq 0.0$ is a parameter of tightness with the default value of 1.0 and is tuneable by the user. With this modification, the range of the M–N distance (d) values will be $\left[0.0, \sqrt{1+t^2}\right]$. As the M–N plots technique aims to maximise clusters' sizes while minimising their dispersion, t controls the weight of contribution of the dispersion to this equation. When $t < 1.0$, less tight but larger clusters are favoured, while when $t > 1.0$ tighter but smaller clusters are favoured.

Let $(\mathbb{G}, \mathbb{d}) = \mathtt{UNCLES}(\mathbb{X}, \mathbb{F}, \mathbb{K}, \mathbb{D}, t)$be the list of clusters $\mathbb{G} = \left\{G_1, \ldots, G_{|\mathbb{G}|}\right\}$ paired with their ordered list of M–N distance values $\left\{d_1, \ldots, d_{|\mathbb{G}|}\right\}$ as generated by applying UNCLES to the $\mathbb{X}$ datasets, using the $\mathbb{F}$ base clustering methods, with the $\mathbb{K}$ set of numbers of clusters, the $\mathbb{D}$ set of δ values, and the user-defined tightness parameter t. Therefore, *clust* is applied by the following steps [5]:

1. Apply UNCLES to obtain an ordered list of clusters alongside their M–N distance values:

$$(\mathbb{G}, \mathbb{d}) = \mathrm{UNCLES}(\mathbb{X}, \mathbb{F}, \mathbb{K}, \mathbb{D}, t) \tag{4.16}$$

2. Keep "good" clusters and filter out the rest:
The distribution of d values in $\mathbb{d}$has been seen in many examples to have few small (good) values and a long tail of large (bad) values. Therefore, define the point in the ordered list $\mathbb{d}$that splits between good and bad clusters as the largest gap (difference) between any two consecutive d values after weighting these gaps by their reversed order. In other words, larger weights are given to the gaps closer to the top of the list, to avoid selecting the large gaps that tend to appear towards the higher tail of the distribution:

$$\begin{aligned} \Delta_i &= (|\mathbb{d}| - i) \times (d_{i+1} - d_i), \ \ \forall i \in \{1 \ldots (|\mathbb{d}| - 1)\} \\ k^{good} &= \underset{i \in \{1 \ldots (|\mathbb{d}|-1)\}}{\operatorname{argmax}} \ \Delta_i \\ \mathbb{G}^{good} &= \left\{G_1 \ldots G_{k^{good}}\right\}, \end{aligned} \tag{4.17}$$

where Δ_i is the weighted gap size (difference) between the two values d_i and d_{i+1} and k^{good} is the number of good clusters.

3. Optimise and complete the set of good clusters by removing outlier genes and including genes that were missed despite fitting within clusters' dynamic ranges. This is done by the following steps:

- For each dataset, calculate the absolute difference between the expression values of each gene and the mean expression profile of the cluster to which that gene is assigned:

$$\begin{aligned} \mathbb{G}_l^{\mathrm{nonzero}} &= \left\{G \in \mathbb{G}^{\mathrm{good}} | \left(\exists z_{l,G,s} \neq 0.0\right) \text{ for any } s\right\} \\ e_{l,i,s} &= \sum_{G \in \mathbb{G}^{\mathrm{nonzero}}} \lambda_{i,G} \cdot \left|x_{l,i,s} - z_{G,l,s}\right|, \\ \lambda_{i,G} &= \begin{cases} 1, (\text{gene } i) \in (\text{cluster } G) \\ 0, (\text{gene } i) \notin (\text{cluster } G) \end{cases} \in \{0, 1\}, \end{aligned} \tag{4.18}$$

where $\mathbb{G}^{\mathrm{nonzero}}$ is the set of good clusters with average expression profiles that are not equal to zero across all samples, $\lambda_{i,G}$ is the membership of the ith gene in the Gth cluster, $x_{l,i,s}$ is the gene expression value of the ith gene in sth sample (condition or time-point) of the lth dataset, and $z_{G,l,s}$ is the average expression of the values of the sth sample in the lth dataset for all genes belonging to the Gth cluster:

$$z_{G,l,s} = \frac{1}{\sum \lambda_{i,G}} \sum \lambda_{i,G} x_{l,i,s}. \tag{4.19}$$

Note that $e_{l,i,s}$ is not calculated for all genes in all datasets, but only for genes belonging to clusters with expression profiles that are not flat and equal to zero over all samples. As some clusters may have zero average profiles in some datasets but not in others, some genes may have $e_{l,i,s}$ values based on some but not all datasets.

- Calculate the third quartile of all $e_{l,i,s}$ values separately for each sample in each dataset:

$$q_{l,s} = \underset{\forall s \in l}{\text{third\_quartile}}\left(e_{l,i,s}\right). \tag{4.20}$$

The distribution of e values usually is one-tailed with its mode being at zero or another close-to-zero value.

- Calculate data-driven lower and upper boundaries (margins around average) of each cluster at each sample of each dataset:

$$\begin{aligned} \text{low}_{G,l,s} &= z_{G,l,s} - \gamma \cdot q_{l,s} \\ \text{up}_{G,l,s} &= z_{G,l,s} + \gamma \cdot q_{l,s}, \end{aligned} \tag{4.21}$$

where γ has the default value of 2.0 but can be customised by the user. Here the boundaries are calculated for each sample in each dataset with the assumption that different samples even in a single dataset might have inherently different dynamics and distributions.

- Resolve overlapping boundaries between clusters. This problem happens when the boundaries overlap across all samples in all datasets. Let $t_{l,s,G1,G2} \in \{-1, 0, 1, 2\}$ be the type of the overlap between the boundaries of the clusters $G1$ and $G2$ at the sth sample of the lth dataset:

$$t_{l,s,G1,G2} = \begin{cases} -1, & (\text{low}_{G2,l,s} \le \text{low}_{G1,l,s} \le \text{up}_{G1,l,s} \le \text{up}_{G2,l,s}) \\ & \qquad\qquad\qquad \cup \\ & (\text{low}_{G1,l,s} \le \text{low}_{G2,l,s} \le \text{up}_{G2,l,s} \le \text{up}_{G1,l,s}) \\ 0, & (\text{low}_{G1,l,s} < \text{low}_{G2,l,s} \le \text{up}_{G1,l,s} \le \text{up}_{G2,l,s}) \\ 1, & (\text{low}_{G2,l,s} < \text{low}_{G1,l,s} \le \text{up}_{G2,l,s} \le up_{G1,l,s}) \\ 2, & \qquad \text{otherwise(no overlap)} \end{cases}. \tag{4.22}$$

The overlap type t^{ov} of -1 indicates that one of the two clusters' boundaries is entirely encompassed by the other cluster's boundaries, the types 0 and 1 indicate that the upper boundary of the cluster $G1$ or $G2$, respectively, is within the boundaries of the other cluster, and the type 2 indicates that there is no overlap between the two clusters at this sample. Let the value of the overlap at the sth sample of the lth dataset between the clusters $G1$ and $G2$ be:

$$ov_{l,s,G1,G2} = \begin{cases} \infty, & t_{l,s,G1,G2} = -1 \\ \mathrm{up}_{G1,l,s} - \mathrm{low}_{G2,l,s}, & t_{l,s,G1,G2} = 0 \\ \mathrm{up}_{G2,l,s} - \mathrm{low}_{G1,l,s}, & t_{l,s,G1,G2} = 1 \\ -1, & t_{l,s,G1,G2} = 2 \end{cases}. \tag{4.23}$$

Finally, resolve the overlap with minimum disruption by altering the boundaries at the sample with the minimum overlap. Also, make that change in the cluster with higher (worse) M–N distance d amongst the pair of the clusters to introduce less change to the clusters with better (lower) d values. For each cluster pair $(G1, G2)$ where $d_{G1} \leq d_{G2}$, let $l^{\min}$ and $s^{\min}$ be the dataset and the sample at which the overlap value is the minimum:

$$\left(l^{\min}, s^{\min}\right) = \underset{\forall l \forall s}{\operatorname{argmin}}\ ov_{l,s,G1,G2}. \tag{4.24}$$

Then resolve the overlap by:

$$\begin{cases} \mathbb{G}^{\text{good}} \leftarrow \mathbb{G}^{\text{good}} - \{G2\}, & t_{l^{\min},s^{\min},G1,G2} = -1 \\ \mathrm{low}_{G2,l^{\min},s^{\min}} = \mathrm{up}_{G1,l^{\min},s^{\min}} + \varepsilon, & t_{l^{\min},s^{\min},G1,G2} = 0 \\ \mathrm{up}_{G2,l^{\min},s^{\min}} = \mathrm{low}_{G1,l^{\min},s^{\min}} - \varepsilon, & t_{l^{\min},s^{\min},G1,G2} = 1 \\ \text{do nothing}, & t_{l^{\min},s^{\min},G1,G2} = 2 \end{cases} \tag{4.25}$$

where ε is a very small positive real number.

- Drop all genes included in the set of good clusters and reconstruct them from scratch by including in each cluster all genes that have expression profiles consistently fitting within its boundaries at all samples in all datasets. This is done by revisiting all genes in the dataset and not only those previous included in $\mathbb{G}^{good}$.

$$\begin{aligned} \mathbb{G}^{\text{final}} &\leftarrow \left\{G^{\text{final}}, \forall G \in \mathbb{G}^{\text{good}}\right\} \\ G^{\text{final}} &\leftarrow \left\{g | \mathrm{low}_{l,G,s} \leq x_{l,g,s} \leq \mathrm{up}_{l,G,s}, \forall l \forall s\right\}. \end{aligned} \tag{4.26}$$

Note that the original genes in clusters were used to learn the average profiles of these clusters and their upper and lower boundaries around those averages. However, if some of genes originally included in the clusters were outliers, that is, had expression values outside of the data-driven boundaries, they will not be included in the reconstructed clusters. Furthermore, it had been empirically seen that it is likely to find many genes with profiles that fit within the generated clusters' boundaries but were missed from them; such genes will be included in the reconstructed clusters.

- Remove clusters with fewer genes that the size of the smallest acceptable cluster sc, which is set to 11 by default by can be customised by the user:

$$\mathbb{G}^{\text{final}} \leftarrow \left\{ G \in \mathbb{G}^{\text{final}} \middle| |G| \geq sc \right\} \tag{4.27}$$

A Python implementation of *clust* along with instructions for users are openly available at https://github.com/BaselAbujamous/clust. It can be installed from the Python Package Index (*PyPi*) with the direct command line "*pip install clust*", and a webpage implementation is available at http://clust.baselabujamous.com/.

4.1.5 Deep Learning Approaches

Deep learning has emerged as the state-of-the-art machine learning enabling more complex pattern recognition and prediction tasks and powered by the developments in parallel computing. Various studies have proposed deep learning approaches to analyse multiple gene expression datasets, or in many cases, multiple *omic* datasets, simultaneously.

For instance, Chaudhary and colleagues proposed a framework in which tens of thousands of omic features from multiple datasets were submitted in parallel to an autoencoder architecture to embed into a space with a lower dimension [19]. Then, downstream analysis was applied to this embedded space aiming to train a model that classifies cancer patients into clusters with differential survival risk scores. This approach in fact was not designed to find clusters of genes based on their profiles over multiple samples; rather it aimed at clustering patients based on their gene expression profiles as well as their profiles in other omic datasets, namely microRNA expression and methylomics. However, it can be adapted to gene clustering based on multiple gene expression datasets by considering single genes as the training samples of the deep learning architecture and by considering the pooled collection of samples or conditions from all datasets as the feature space. This method is an example of an intermediate integration approach.

In another study, Cheerla and Gevaert proposed an intermediate integration approach aiming to predict patients with different survival profiles based on their profiles in gene expression datasets as well as in datasets of other types such as clinical data and image data [20]. Their model embeds different datasets of different types into the same space of 512 dimensions before applying downstream analysis [20]. This is carried out through a separate feedforward network for each dataset suiting its type. For example, highway networks were used for gene expression and microRNA datasets and convolutional neural networks (CNN) were used for imaging datasets. Given multiple data vectors for the same patient in different datasets, the networks were trained to maximise the similarity between the embedded values of each of these vectors. This architecture assumes that the same patient has multiple profiles in different datasets. However, for gene clustering, the model can be altered to assume that the same gene has multiple profiles in multiple datasets across different sets of conditions and time points.

Taken together, the ability of deep learning models to process high dimensional data in complex formats enables the design of novel architectures that do not only analyse multiple gene expression datasets simultaneously, but also incorporate further types of related data. The variety in possible deep learning architectures allows them to be customised to fit the structures of datasets more flexibly. However, these methods require large numbers of samples for their training to converge sufficiently and should only be employed where applicable.

4.2 Case Study 1: A Novel Subset of Genes with Expression Consistently Oppositely Correlated with Ribosome Biogenesis in Forty Yeast Datasets

Budding yeast *Saccharomyces cerevisiae* represents a well-studied eukaryotic model species due to its short cell-cycle duration, ease in handling, and low-cost accessibility. This has resulted in an accumulation of many gene expression datasets investigating this species under a variety of conditions. Although each one of these datasets was generated within a specific context in order to answer a specific research question, a biological question of value would be to learn if there are centrally regulated processes in budding yeast regardless of the biological context and specific conditions. In order to achieve that, a simultaneous analysis of a diverse collection of budding yeast gene expression datasets could be performed.

4.2.1 Data and Approach

Forty budding yeast microarray datasets, listed in Table 4.2, were analysed simultaneously using the binarisation of consensus partition matrices (Bi-CoPaM) method [6] in order to extract the subsets of genes that are consistently co-expressed in all of these datasets [3].

4.2.2 Results and Discussion

Two Consistently Anti-correlated Clusters of Genes were Identified in Budding Yeast by Simultaneous Clustering

The application of Bi-CoPaM to the forty datasets revealed that there are two subsets of genes consistently co-expressed in all forty datasets, which are labelled as C1 and C2 [3]. The profiles of these two clusters in each one of the forty datasets are shown in Fig. 4.1. The first cluster, C1, has 257 genes, and the second cluster, C2, has 47 genes. The full lists of gene names in these two clusters can be found in Supplementary Table S1 in [3].

Table 4.2 Forty yeast microarray datasets (from Table 1 in [3])

ID	GEO accession	N[b]	Description	References
D01	GSE8799	15	Two mitotic cell-cycles (w/t)	[60]
D02	GSE8799	15	Two mitotic cell-cycles (mutated cyclins)	[60]
D03	E-MTAB-643[a]	15	Response to an impulse of glucose	[28]
D04	E-MTAB-643[a]	15	Response to an impulse of ammonium	[28]
D05	GSE54951	6	Response of *dal80Δ* mutant yeast to oxidative stress induced by linoleic acid hydroperoxide	–
D06	GSE25002	9	Osmotic stress response and treatment of transformants expressing the *C. albicans* Nik1 gene	–
D07	GSE36298	6	Mutations of OPI1, INO2, and INO4 under carbon-limited growth conditions	[24]
D08	GSE50728	8	120 h time-course during fermentation	–
D09	GSE36599	5	Stress adaptation and recovery	[78]
D10	GSE47712	6	Combinations of the yeast mediator complex's tail subunits mutations	[50]
D11	GSE21870	4	Combinations of mutations in DNUP60 and DADA2	–
D12	GSE38848	6	Various strains under aerobic or anaerobic growth	[52]
D13	GSE36954	6	Response to mycotoxic type B trichothecenes	[71]
D14	GSE33276	6	Response to heat stress for three different strains	–
D15	GSE40399	7	Response to various perturbations (heat, myriocin treatment, and lipid supplement)	–
D16	GSE31176	6	W/t, *rlm1Δ*, and *swi3Δ* cells with or without Congo Red exposure	[64]
D17	GSE26923	5	Varying levels of GCN5 F221A mutant expression	[49]
D18	GSE30054	31	CEN.PK122 oscillating for 2 h	–
D19	GSE30051	32	CEN.PL113-7D oscillating for 2 h	[23]
D20	GSE30052	49	CEN.PL113-7D oscillating for 4 h	[23]
D21	GSE32974	15	About 5 h of cell-cycle (w/t).	[44]
D22	GSE32974	15	About 4 h of cell-cycle (mutant lacking Cdk1 activity)	[44]
D23	GSE24888	5	Untreated yeast versus yeasts treated with *E. arvense* herbs from the USE, China, Europe, or India	–
D24	GSE19302	6	Response to degron induction for w/t and nab2-td mutant	[38]
D25	GSE33427	5	Untreated w/t, and wt/t, *yap1Δ*, *yap8Δ*, and double mutant treated with AsV	[35]
D26	GSE17716	7	Effect of overexpression and deletion of MSS11 and FLO8	[12]
D27	GSE31366	4	Presence and absence of multi-inhibitors for parental and tolerant strains	–

(continued)

Table 4.2 (continued)

ID	GEO accession	N[b]	Description	References
D28	GSE26171	4	Response to patulin and/or ascorbic acid	[70]
D29	GSE22270	4	PY1 and Met30 strains in room temperature or 35 °C	–
D30	GSE29273	4	Time-series during yeast second fermentation	–
D31	GSE29353	5	Different haploid strains growing in low glucose medium	[63]
D32	GSE21571	8	Different combinations of mutations in HTZ1, SWR1, SWC2, and SWC5	[58]
D33	GSE17364	4	Untreated w/t and Slt2-deficient yeasts, or treated with sodium arsenate for two hours	[55]
D34	GSE15352	8	24 h time-course of yeast grown under a low temperature (10 °C)	[69]
D35	GSE15352	8	24 h time-course of yeast grown under a normal temperature (28 °C)	[69]
D36	GSE15352	8	24 h time-course of yeast grown under a high temperature (37 °C)	[69]
D37	GSE16799	21	UC-V irradiation of w/t, *mig3Δ*, *SNF1Δ*, *RAD23Δ*, *RAD4Δ*, and *snf1Δrad23Δ*	[75]
D38	GSE16346	4	BY474 cells grown to mid-log under presence versus absence of L-carnitine and/or H_2O_2	–
D39	GSE14227	10	Two hours of wild-type yeast growth	[37]
D40	GSE14227	9	Two hours of *sch9Δ* mutant yeast growth	[37]

[a] D03 and D04 have accession numbers in the European Bioinformatics Institute (EBI) repository rather than GEO accession numbers

[b] N is the number of conditions or time-points in the dataset, i.e. the number of dimensions

The fact that genes in any of these two clusters are consistently co-expressed in all forty datasets does not necessitate that their average profiles in all forty datasets are the same. In fact, the biological conditions or time-points of these datasets, represented by their x-axes, differ substantially, and are not aligned. For instance, the 15 time-points on the x-axis of D01 and D02 measure gene expression profiles over two mitotic cell-cycles, the 8 time-points on the x-axis of D08 measure expression profiles over two hours during fermentation, and the 4 conditions in D29 measure gene expression of PY1 or Met30 yeast strains, each under room temperature or 35 °C. Note that the latter does not resemble a time-series dataset as there is no temporal element ordering its four dimensions. Therefore, the fact that a subset of genes is consistently co-expressed in all forty datasets means that these genes are upregulated together and down-regulated together, in each one of these datasets.

Functional Characterisation of the Two Clusters of Genes

Gene ontology (GO) term analysis revealed that C1 is enriched with rRNA processing and ribosome biogenesis, and therefore was labelled as RRB. As for C2, it contained

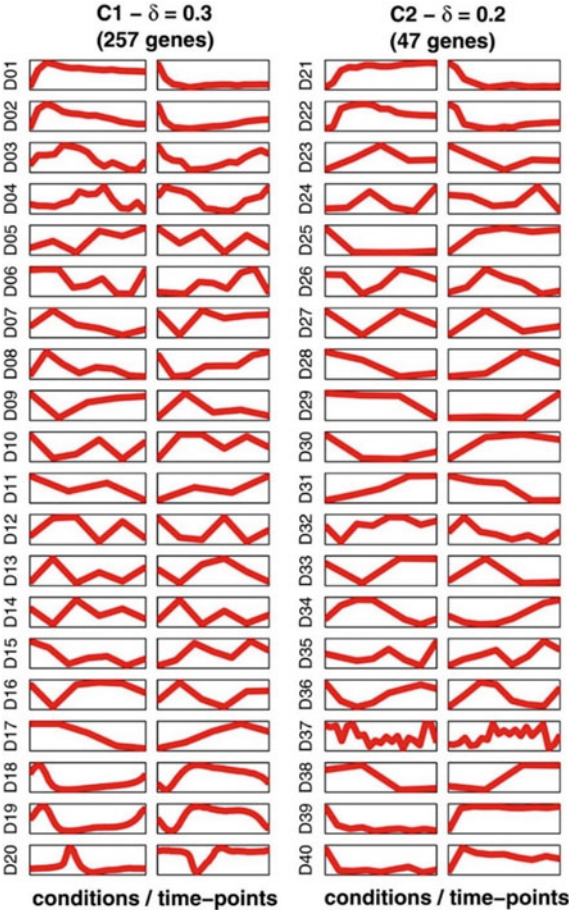


Fig. 4.1 The average profiles of the two consistently co-expressed clusters C1 and C2 in each one of the forty budding yeast microarray datasets. Taken from Fig. 3 in [3]

some stress response genes and numerous genes with unknown function. Therefore, C2 was labelled as “anti-phase with ribosome biogenesis”, or *Apha-RiB*.

Transcriptional Regulators of the Two Anti-correlated Clusters

It may be hypothesised that the consistent anti-correlated gene expression profiles of C1 and C2 are due to their being co-regulated by the same regulatory machinery, but in two opposite directions. In order to search for potential mutual regulators of the two clusters, and in order to understand the regulatory network controlling both clusters in general, the MEME tool was used to identify overrepresented DNA sequences in the upstream regions of genes in the C1 or C2 clusters. Three motifs, labelled as C1-1, C1-2, and C1-3, were found in the upstream regions of C1 genes, and two motifs, labelled as C2-1 and C2-2, were found in the upstream regions of C2 genes (Fig. 4.2). Furthermore, the TOMTOM tool was used to identify known

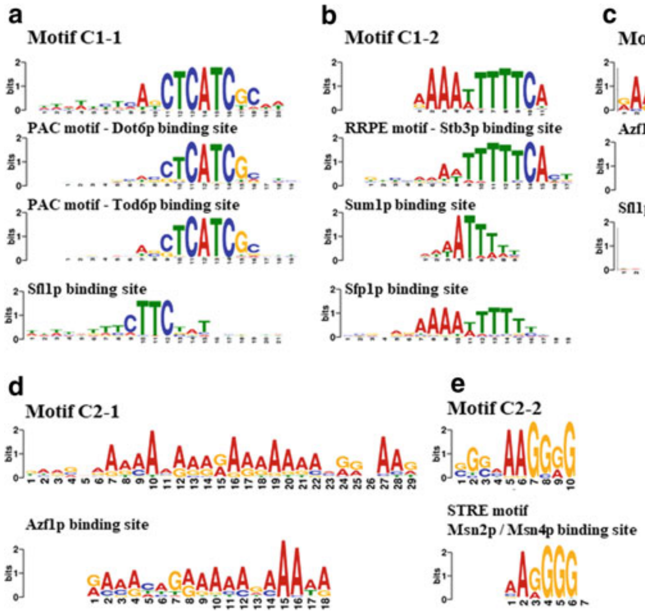


Fig. 4.2 DNA motifs (short sequences) overrepresented in the upstream regions of genes in C1 and C2 aligned with the known binding sites of some transcription factors. The motifs were labelled as C1-1 (A), C1-2 (B), C2-3 (C) for those in the upstream sequences of C1 genes, and C2-1 (D) and C2-2 (E) for those in the upstream sequences of C2 genes. This Figure has been compiled from Figs. 4 and 5 in [3]

transcription factor binding sites that significantly match the identified DNA motifs (Fig. 4.2).

Notably, the PAC motif, which is the binding site of the Dot6p transcription factor, and the RRPE motif, which is the binding site of the Stb3p transcription factor, were both found significantly overrepresented in the upstream regions of C1 (Fig. 4.2a, b). This result is in line with the observation that C1 is enriched with rRNA processing and ribosome biogenesis genes, as these two DNA motifs are known to be associated with these processes [14]. Also, the overrepresentation of the stress response element (STRE), which is the binding site of the Msn2p/Msn4p transcription factors, is in agreement with the known role of this element and these transcription factors in response to stress [13].

With the hypothesis that both clusters may be co-regulated by the same regulator(s) but in opposite directions, it is interesting to note that the binding site of Azf1p is overrepresented in both clusters. Therefore, Azf1p might be part of the hypothesised mutual regulatory machinery. Furthermore, the binding site of Stb1, which is TGAAAA, substantially overlaps with the motif C2-1 (Fig. 4.2), and therefore it might also be part of the mutual regulatory machinery, especially given that its overexpression was seen to increase resistance to oxidative stress [29].

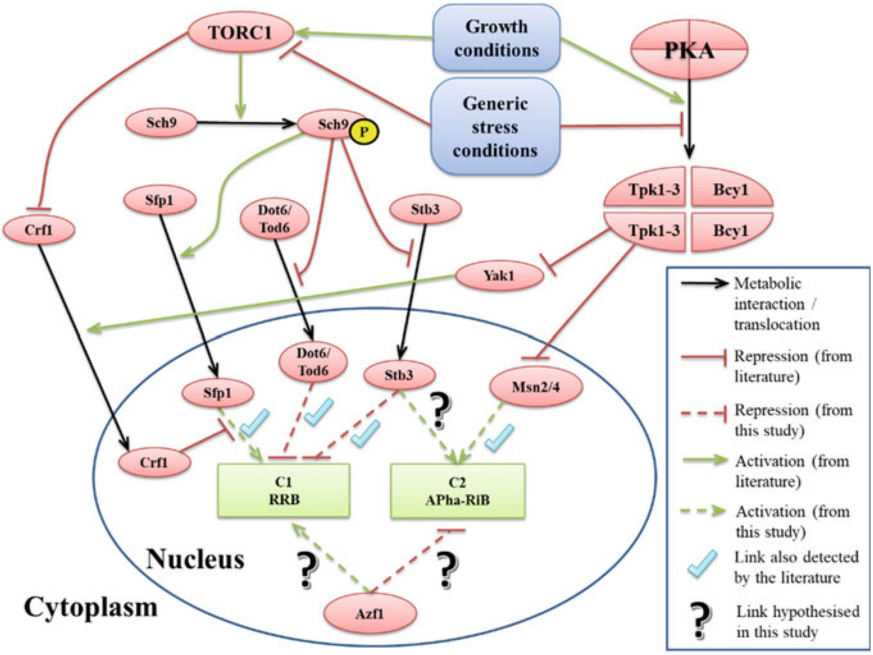

TORC1
Growth conditions
PKA
Generic stress conditions
Sch9
Sch9
P
Sfp1
Dot6/ Tod6
Stb3
Tpk1-3
Bcy1
Tpk1-3
Bcy1
Crf1
Yak1
Dot6/ Tod6
Stb3
Msn2/4
Sfp1
Crf1
C1 RRB
C2 APha-RiB
Nucleus
Cytoplasm
Azf1
?
?
?
Metabolic interaction / translocation
Repression (from literature)
Repression (from this study)
Activation (from literature)
Activation (from this study)
✓ Link also detected by the literature
? Link hypothesised in this study

Fig. 4.3 A summary diagram of the hypothesised regulatory network controlling the expression of the genes in the two clusters C1 and C2 under growth and stress conditions

Figure 4.3 summarises the speculated regulatory network stimulated by growth and stress conditions resulting in anti-correlated expression profiles of the C1 (RRB) genes and the C2 (APha-RiB) genes. Further confirmatory experiments will be required to validate the hypothesised links in this network.

4.2.3 Summary and Conclusions

Two subsets of genes are consistently co-expressed in forty different yeast gene expression datasets. These two subsets are also consistently anti-correlated with each other, indicating that they may be regulated by a mutual regulatory machinery but in opposite directions. The first subset is enriched with genes participating in rRNA processing and ribosome biogenesis, and is labelled as RRB, while the second subset has some stress response genes and many genes of unknown function, as is therefore labelled as being in "antiphase with ribosome biogenesis" (Apha-RiB). An investigation of the upstream sequences of genes in these two subsets revealed that the RRB group is regulated by Sfp1, Dot6/Tod6, and Stb3, while the Apha-RiB group is regulated by Msn2/Msn4. Speculations on potential common regulators for both groups highlighted Stb3 and Azf1 as potential candidates, which have to be further validated by biological experiments.

4.3 Case Study 2: A Transcriptomic Signature Derived from a Study of Sixteen Breast Cancer Cell-Line Datasets Predicts Poor Prognosis

Hypoxia, which is the deficiency in oxygen supply, is a characteristic of poor prognosis in cancers with solid tumours such as breast cancer [40, 67]. Altered transcriptional programmes are induced in tumour cells to produce resistance to the hypoxic conditions mediated by stabilising the hypoxia-inducible factor (HIF) proteins [67, 68]. Therefore, numerous studies attempted to identify the subset of genes directly regulated by HIF proteins, or by hypoxic conditions in general, and thereafter sought to use their expression levels as prognostic markers [16, 41].

Consequently, several datasets have been produced measuring gene expression in cancer under different conditions related to hypoxia. Simultaneous clustering of these datasets can identify a data-driven hypoxia transcriptomic signature with prognostic power.

4.3.1 Data and Approach

Table 4.3 lists sixteen microarray gene expression datasets of breast cancer cell-lines under such conditions. The consensus clustering method UNCLES [4] was applied to these datasets to identify subsets of genes that are consistently co-expressed under all of these conditions [1].

4.3.2 Results and Discussion

Consensus Clustering Identifies Two Anti-correlated Clusters of Genes

As the sixteen datasets were generated using different microarray platforms, some genes are not represented by probes in all these datasets. A total of 15,588 genes were present in at least thirteen datasets and were therefore submitted to the UNCLES method for simultaneous clustering. UNCLES analysis followed by cluster quality filters identified two clusters of genes, which were labelled as C1 (504 genes) and C2 (598 genes). The remaining 14,486 genes included in the input were not assigned to any of these two clusters. The gene expression profiles of these clusters are displayed in Fig. 4.4.

Notably, these two discovered clusters are consistently anti-correlated over all sixteen datasets ($\rho < -0.85$). In other words, whenever C1 genes are upregulated, C2 genes are downregulated, and vice versa. This observation may indicate that these two clusters are anti-regulated by the same regulatory machinery, directly or indirectly.

C2 is Induced Under Hypoxic Conditions and is Enriched with HIF Targets

A closer look into the expression profiles of C1 and C2 in the conditions of the sixteen datasets shows that C1 is consistently upregulated under growth conditions, such as normal oxygen levels, and is downregulated under hypoxic conditions, while C2 is expressed the other way around.

Table 4.3 List of the 16 microarray datasets of breast cancer cell-lines under conditions related to hypoxia. This table is constructed from Table 1 in [1]

ID	GEO acc.	Year	N[b]	Cell-lines(s)	Description	References
D01	GSE3188[a]	2005	7	MCF7	Same samples of the last two datasets in the same order, but a different platform	[32]
D02	GSE47533	2014	4	MCF7	Time-series data through 48 h of exposure to hypoxia (1% O)	[18]
D03	GSE41491	2012	8	MCF7	Time-series data through 24 h of exposure to hypoxia (1% O)	[43]
D04	GSE47009	2014	3	MCF7	Samples at normoxia, hypoxia, and anoxia, respectively	–
D05	GSE18494	2009	4	MDA-MB-231	Time-series data through 12 h of exposure to hypoxia (0.5% O)	[77]
D06	GSE3188[a]	2005	3	MCF7	Samples at normoxia, hypoxia, and normoxia exposed to DMOG, respectively	[32]
D07	GSE17188[a]	2010	3	SCP2 subline of MDA-MB-231	Time-series data through 24 h of exposure to hypoxia	[53]
D08	GSE17188[a]	2010	3	LM2 subline of MDA-MB-231	Time-series data through 24 h of exposure to hypoxia	[53]
D09	GSE15530	2010	4	MCF7	Normoxia samples versus hypoxia samples, each is either transfected with non-specific shRNA or with reptin shRNA	[51]

(continued)

Table 4.3 (continued)

ID	GEO acc.	Year	N[b]	Cell-lines(s)	Description	References
D10	GSE45362	2013	4	MB231RN-LM derived from MDA-MB-231	Non-transfected samples versus transfected with has-miR-18a, each is in either a control medium or treated with Cobalt(II) chloride ($CoCl_2$) hypoxia-mimicking agent	[45]
D11	GSE29406	2012	4	MCF7	Normoxia samples versus hypoxia samples, each is either untreated or treated with lactic acid	[72]
D12	GSE18384	2010	4	MCF7	Normoxia samples versus hypoxia samples, each is either non-transfected or transfected with siRNA#1 against JMJD2B	[79]
D13	GSE3188[a]	2005	4	MCF7	Samples exposed to/transfected with oligogectamine, HIF1α siRNA, HIF2α siRNA, or both siRNAs, respectively. All samples were grown under hypoxia (1% O) for 16 h	[32]
D14	GSE33438	2011	4	MCF7 & ZR-75-1	Samples from each of the two cell-lines were exposed to hypoxia for 24 or 48 h	[8]
D15	GSE49953	2013	4	T47D & MDA-MB-231	A control sample and an XBP1-knocked-down sample from each of the two cell-lines. All samples are under hypoxia and glucose deprivation	[22]

(continued)

Table 4.3 (continued)

ID	GEO acc.	Year	N^b	Cell-lines(s)	Description	References
D16	GSE30019	2012	6	MCF7	Time-series data through 24 h of reoxygenation after having been in hypoxia (0.5% O) for 24 h	[47]

*a*These accession numbers refer to datasets that include samples that semantically form multiple different datasets

*b*N is the number of conditions in the dataset, i.e. the number of dimensions

The two clusters were functionally characterised by gene ontology (GO) enrichment analysis and pathway enrichment analysis; the latter was performed using the GeneCodis online tool over the KEGG pathways database. This revealed that C1 is enriched with genes that participate in ribosomal RNA (rRNA) processing and ribosome biogenesis (aka RRB), which is in line with the fact that this cluster is upregulated under growth conditions. On the other hand, C2 is enriched with genes involved in signal transduction, positive regulation of I-kappaB kinase/NF-kappaB cascade, carbohydrate metabolism, glycolysis, and response to hypoxia; these processes are induced to respond to the hypoxic stress.

The two clusters were further assessed in their content of targets of the HIF hypoxia-induced factor. Seven lists of potential HIF1α and HIF2α targets were fetched from different studies and are described in Table 4.4. Surprisingly, the intersection of these seven lists contains only two genes, while the union of all these lists has 1521 (Table 4.5). As shown in Table 4.5, C2 is enriched with HIF targets regardless of which combination of lists of HIF potential targets was considered, while C1 is not. This adds to the evidence that the upregulation of C2 represents a transcriptomic signature of hypoxia in breast cancer cell-lines.

In order to identify potential regulators of the genes in C1, usage of the Enrichr analysis tool [21, 46] revealed that C1 is enriched with targets of the MYC transcription factor, which is an oncogene encoding a transcription factor that selectively amplifies the expression of genes participating in cell growth and proliferation. The same analysis tool, when applied to the C2 cluster, confirmed that it is enriched with targets of HIF1α.

C1 and C2 form Multiple Sub-clusters in Clinical Tumours

Although this data-driven hypoxia signature is consistent across sixteen breast cancer cell-line datasets and was further supported by functional characterisation, it is vital to confirm its prognostic value in clinical samples of breast tumours. Therefore, their expression profiles were investigated in The Cancer Genome Atlas (TCGA) dataset, which includes gene expression profiles of 1026 breast tumour samples. Interestingly, each one of the two clusters C1 and C2 formed multiple sub-clusters of genes with substantially different expression profiles in these tumour samples (Fig. 4.5). This indicates the disagreement between clinical samples and what appears

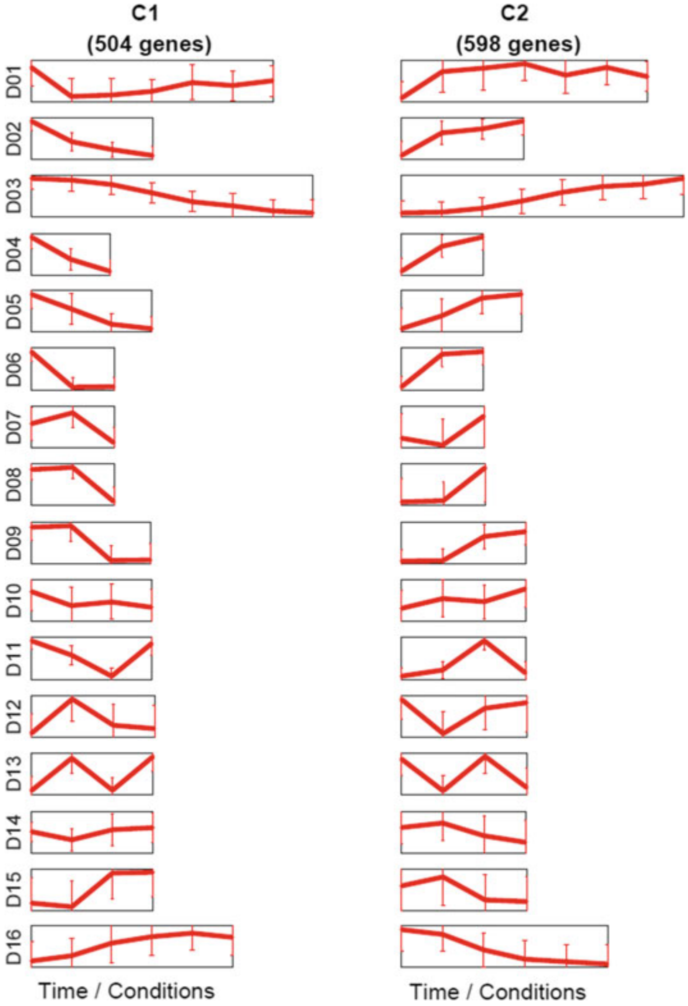


Fig. 4.4 Average expression profiles of the genes in the clusters C1 and C2 in each one of the 16 datasets. Horizontal axes represent different biological conditions or time-points, and the vertical axes represent normalized gene expression values (z-scores). This figure was taken from Fig. 2 in [1]

to be consistent across sixteen different cell-line datasets, hinting to the fact that research findings from cell-line studies must always be considered with care. The major sub-clusters formed from C1 and C2 were labelled as C1a, C1b, C2a, C2b, and C2c.

To assess if these sub-clusters are significantly co-expressed in clinical datasets beyond the TCGA dataset, the mean-squared error (MSE) was calculated amongst their genes in a number of other breast cancer clinical datasets listed in Table 4.6.

Table 4.4 Description of seven lists of genes which were identified as potential targets for HIF (from Table 3 in [1])

List	TF[a]	N[b]	Approach	References
L1	HIF1α	500	Integrative genomes (computational & experimental)	[11]
L2	HIF1α	394	Genome-wide ChIP	[57]
L3	HIF2α	131	Genome-wide ChIP	[57]
L4	HIF1α	311	ChIP-chip	[77]
L5	HIF	216	Genome-wide computational approaches	[61]
L6	HIF1α	323	ChIP-seq.	[66]
L7	HIF2α	268	ChIP-seq.	[66]

[a] The HIF transcription factor (TF) which was considered in the study
[b] The number of distinct genes included in the list (N)

Table 4.5 Different combinations of the seven lists of HIF potential targets (from Table 4 in [1])

Combination	N[a]	In this study[b]	C1	C2
Union of all	1521	1172	42 ($p = 0.26$)	145 ($p = 1\times10^{-38}$)
Intersection of all	2	2	0 (p = 1.00)	1 ($p = 8\times10^{-2}$)
Intersection of studies with HIF1α (L1, L2, L4, L5, and L6)	11	10	0 (p = 1.00)	6 ($p = 6\times10^{-7}$)
In 3 lists or more	144	126	0 (p = 1.00)	61 ($p = 1\times10^{-52}$)
In 4 lists or more	60	49	0 (p = 1.00)	30 ($p = 2\times10^{-30}$)
In 5 lists or more	28	22	0 (p = 1.00)	15 ($p = 6\times10^{-17}$)

[a] The number of genes in the corresponding combination of lists
[b] The number of genes in the corresponding combination of lists that are included in the input 15,588 genes in our study

An empirical p-value was calculated to quantify the significance of the observed MSE values by calculating an empirical distribution of MSE values via generating 1000 random clusters of similar sizes. The result was that four out of these five sub-clusters showed significant co-expression in all six clinical datasets with p-values ranging from 10^{-6} to 10^{-300}, namely C1a, C1b, C2a, and C2b. Interestingly, the full C2 cluster showed far fewer significant p-values of co-expression in these clinical datasets (p-values ranging from 0.99 to 10^{-6}).

C1, But Not C2, Sub-clusters Demonstrate Significant Prognostic Power

The prognostic power of each one of the sub-clusters of C1 and C2 was examined using COX proportional hazards (COX-PH) analysis and was compared with two published transcriptomic signatures of prognostic value. These two signatures include 51 [16] and 20 genes, respectively, and therefore were labelled as the 51-gene signature and the 20-gene signature [41], respectively.

The sub-clusters of the hypoxia-repressed cluster C1 showed significant prognostic power as their upregulation in breast cancer tumours predicts poor prognosis.

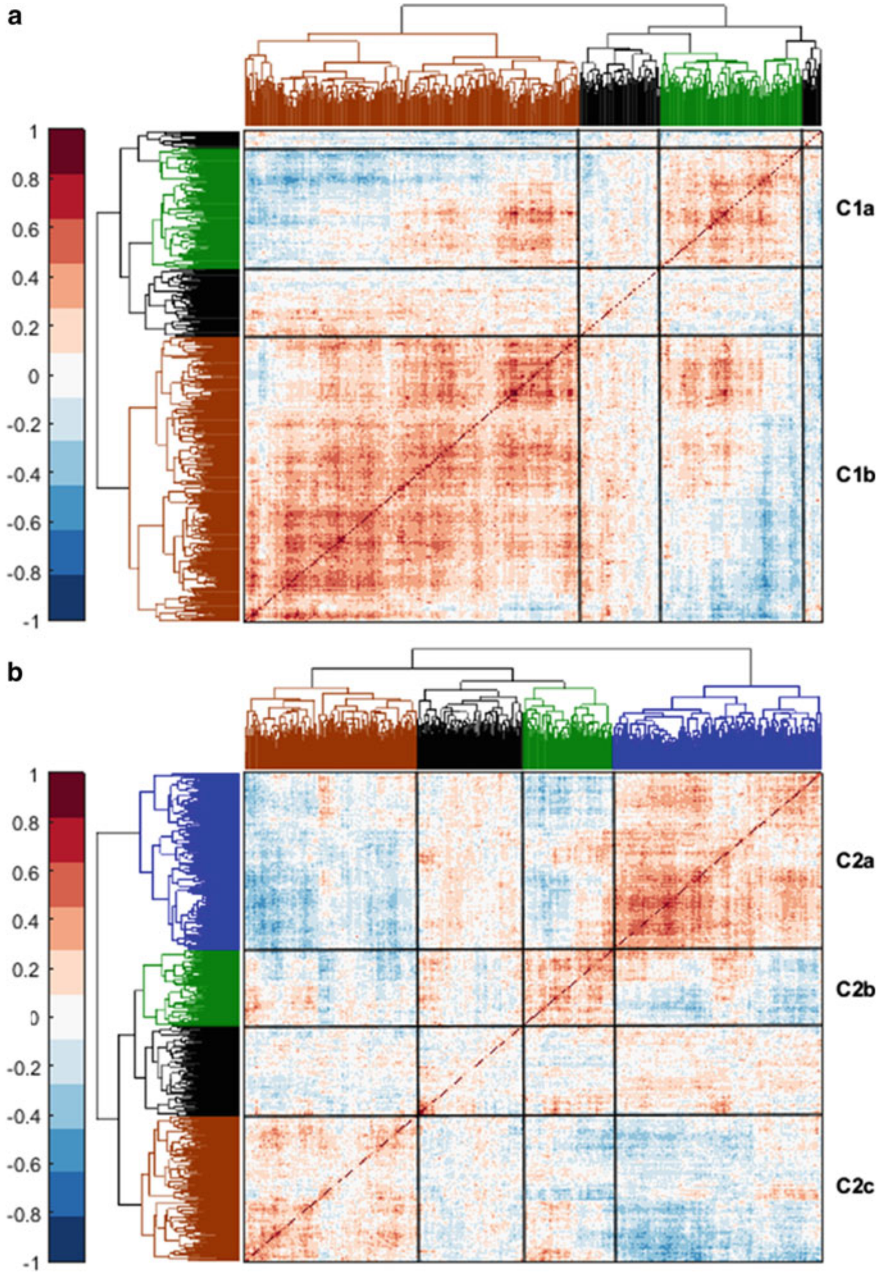


Fig. 4.5 C1 and C2, derived from consensus clustering of 16 cell-line datasets, form multiple sub-clusters of co-expressed genes in the TCGA breast cancer clinical dataset (from Fig. 6 in [1])

Table 4.6 Breast cancer clinical datasets from Table 5 in [1]

Title	Number of samples	OS and ER information?	References
TCGA	1026	Yes	[25]
GE	196	No	[15]
GSE2034	286	Yes	[76]
GSE3494	251	Yes	[56]
METABRIC-disc	997	Yes	[26]
METABRIC-val	995	Yes	[26]

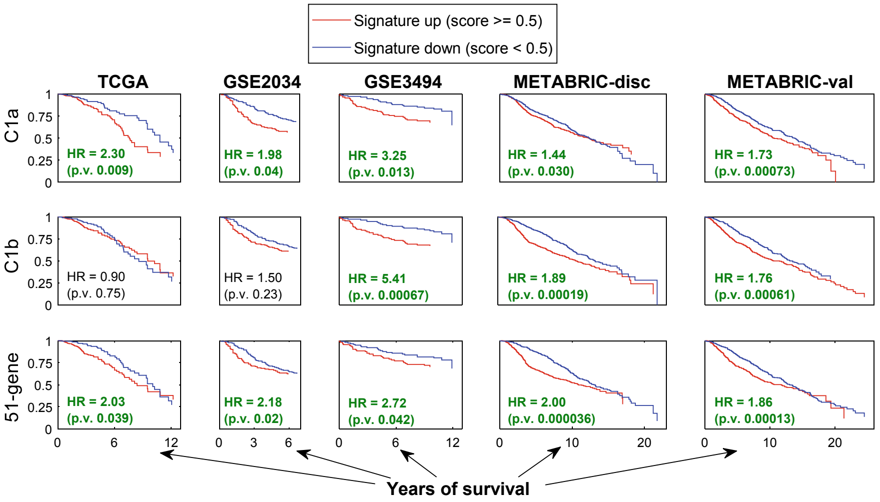


Fig. 4.6 Kaplan Meier (KM) survival curves for the sub-clusters C1a and C1b as well as the 51-gene signature in each one of the five named clinical datasets. The displayed hazard ratio (HR) values were calculated using COX-PH regression and the p-values were calculated using the log-rank test (from Fig. 7 in [1])

This appears to be similar to the 51-gene signature or even more powerful than that in Fig. 4.6.

4.3.3 Summary and Conclusions

Analysis of sixteen breast cancer cell-line datasets revealed two clusters of genes that are consistently co-expressed in all datasets. The first cluster has genes participating in various growth-related processes such as the cell-cycle and proliferation, while the second cluster is enriched with genes participating in the stress response. Consistent anti-correlation between the expression profiles of these two clusters was observed, where the first cluster is upregulated under normal oxygen levels suitable for growth and repressed under hypoxic stress conditions, while the second cluster behaves oppositely. Moreover, and as expected, the first cluster is enriched with targets of the MYC transcription factor and the second cluster is enriched with targets of the hypoxia-induced factor (HIF) protein, which is induced under hypoxia. Interestingly, both clusters do not show a similar consistent co-expression pattern when examined in clinical breast tumour samples; rather each of them was split into smaller subclusters therein. The more striking observation was that the upregulation of the first cluster, and not the second cluster, in clinical tumours predicts poor prognosis, which is counterintuitive. In other words, the upregulation of growth-related genes, and not hypoxia response genes, predicts poor outcomes in breast tumours. Finally, this study demonstrated an example of mismatch between cancer cell-lines and clinical tumours, where a pattern that was observed consistently in sixteen cell-line datasets did not hold in clinical tumour datasets. Therefore, analysis driven from cell-line data must be considered with caution.

4.4 Case Study 3: Cross-Species Application of Clust Reveals Clusters with Contrasting Profiles Under Thermal Stress in Two Rotifer Animal Species

Response to thermal shock has been studied extensively in various species as it influences the genetic and physiological behaviours of organisms. An analysis of two closely related microscopic rotifer animal species that are known to respond differently to thermal stress was conducted in order to identify the underlying genetic basis for the observed divergence in response. The two species were the heat-tolerant *Brachionus calyciflorus s.s.* and its closely related heat-sensitive *Brachionus fernandoi* [62].

Gene expression profiles were measured for each one of the two species while cultured under control, mild, and high levels of heat exposure. Control heat level was set to 20 °C for both species. However, as *B. calyciflorus s.s.* is more heat-tolerant than *B. fernandoi*, mild temperatures were defined as 26 and 23 °C for the two species, respectively, and high temperatures were defined as 32 °C and 26 °C, respectively. Attempts to culture *B. fernandoi* under 32 °C yielded high mortality rates and low usable sample sizes. Two to four replicates from each species at each temperature were taken with approximately 1000 individuals per sample. RNA was extracted from all samples and sequenced using the Illumina HiSeq 4000 platform. RNA-seq reads were pre-processed followed by gene expression quantification according to the protocol described in [62].

Orthologous genes across the two species were identified using OrthoFinder [33]. This identified *orthogroups (OGs)*, where each OG contains none, one, or multiple genes from each one of the two species such that all genes in an OG are orthologues. As the *clust* method allows for the analysis of multiple datasets from multiple species

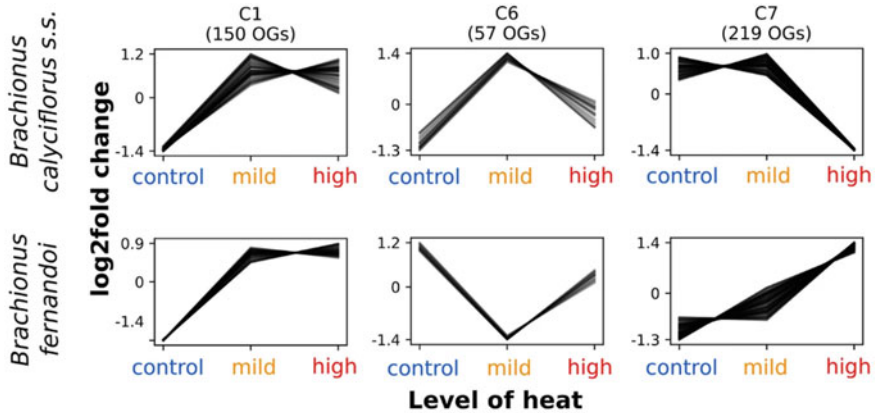


Fig. 4.7 Clusters of biological interest containing contrasting co-expressed orthogroups (OGs) between the heat-tolerant species *Brachionus calyciflorus s.s.* and the related heat-sensitive species *Brachionus fernandoi*. This figure is Fig. 6 in [62]

simultaneously while taking orthology information into account [5], gene expression datasets from both species along with the OrthoFinder's output of OGs were submitted to *clust* in order to extract clusters of co-expressed genes, or rather OGs.

Clust automatically extracted eight clusters of OGs that are consistently co-expressed in both datasets across their three levels of heat. Out of these clusters, three clusters were identified as interesting in terms of their similar or contrasting gene expression profiles, namely, clusters C1, C6, and C7 (Fig. 4.7).

C1 showed a similar profile in both species as it is upregulated under mild and high levels of heat. In contrast, C6 and C7 showed anti-correlated expression profiles across the two species. The heat-tolerance in *B. calyciflorus s.s.* might be mediated by the differential regulation of these two clusters in comparison to its closely related heat-sensitive *B. fernandoi*. Understanding these clusters does not only provide insights into which molecular mechanisms control heat-tolerance, but it also provides potential biomarkers of heat sensitivity in other less-understood related species.

4.5 Summary and Conclusions

With the acceleration of emergence of high-throughput biological datasets, it has become more crucial to develop computational methods and algorithms that can exploit such very large amounts of data to derive useful insights. Amongst the most abundant types of such datasets are gene expression datasets, which measure the expression levels of each gene in the analysed organism over multiple biological conditions or time points. Given any biological context of interest, such as a given type of human disease or a particular biological process in bacteria, it is likely to find multiple gene expression datasets already generated but with different specific designs and detailed conditions. Each one of these datasets would have been generated to answer a specific question. It can be argued that there is a benefit in analysing such collections of datasets simultaneously in order to extract synergistically insights and discoveries that would not have otherwise been found by analysing each dataset separately. The above three case studies offer credence to this argument.

Cluster analysis is one of the domains of computational methods that have been extensively used in the analysis of gene expression datasets. With recent developments, clustering algorithms that can simultaneously analyse multiple gene expression datasets have emerged, such as the Bi-CoPaM, *clust*, COCA, and deep learning methods. The former two have the extra benefit of performing partial clustering, that is, the extraction of tight clusters of genes from input gene expression data without enforcing every input gene to be included in one of the output clusters. This behaviour is consistent with the biological fact that, given a biological context, not all genes are expected to form subsets of co-regulated genes. In fact, the *clust* algorithm, which has a publicly available implementation at https://github.com/BaselAbujamous/clust, is based on the Bi-CoPaM algorithm but with added automation of parameter selection as well as added steps of cluster optimisation and completion. Nonetheless, further developments in the computational methods within this domain will be needed to analyse gene expression datasets as well as other types of datasets simultaneously.

References

1. Abu-Jamous B, Buffa FM, Harris AL, Nandi AK (2017) In vitro downregulated hypoxia transcriptome is associated with poor prognosis in breast cancer. Molec Cancer 16:105
2. Abu-Jamous B, Fa R, Nandi AK (2015) Integrative cluster analysis in bioinformatics, 1st edn. Wiley, Pondicherry
3. Abu-Jamous B, Fa R, Roberts DJ, Nandi AK (2014) Comprehensive analysis of forty yeast microarray datasets reveals a novel subset of genes (APha-RiB) consistently negatively associated with ribosome biogenesis. BMC Bioinform 15:322
4. Abu-Jamous B, Fa R, Roberts DJ, Nandi AK (2015) UNCLES: method for the identification of genes differentially consistently co-expressed in a specific subset of datasets. BMC Bioinform 16:184
5. Abu-Jamous B, Kelly S (2018) Clust: automatic extraction of optimal co-expressed gene clusters from gene expression data. Genome Biol 19(1):172
6. Abu-Jamous B, Fa R, Roberts DJ, Nandi AK (2013) Paradigm of tunable clustering using binarization of consensus partition matrices (Bi-CoPaM) for gene discovery. PLOS One 8(2)
7. Abu-Jamous B, Fa R, Roberts DJ, Nandi AK (2013) Yeast gene CMR1/YDL156W is consistently co-expressed with genes participating in DNA-metabolic processes in a variety of stringent clustering experiments. J Royal Soc Interface 10:20120990
8. Askautrud HA et al (2014) Global gene expression analysis reveals a link between NDRG1 and vesicle transport. PLoS One 9(1):
9. Athar A et al (2019) ArrayExpress update—from bulk to single-cell expression data. Nucleic Acids Res 47(D1):D711–D715
10. Bazaga A, Leggate D, Weisser H (2020) Genome-wide investigation of gene-cancer associations for the prediction of novel therapeutic targets in oncology. Sci Rep 10:10787
11. Benita Y et al (2009) An integrative genomics approach identifies Hypoxia Inducible Factor-1 (HIF-1)-target genes that form the core response to hypoxia. Nucleic Acids Res 37(14):4587–4602

12. Bester MC, Jacobson D, Bauer FF (2012) Many Saccharomyces cerevisiae cell wall protein encoding genes are coregulated by Mss11, but cellular adhesion phenotypes appear only Flo protein dependent. G3 (Bethesda) 2(1):131–141.
13. Boer CGD, Hughes TR (2011) YeTFaSCo: a database of evaluated yeast transcription factor sequence specificities. Nucl Acids Res 40(D1):D169–D179
14. Bosio MC, Negri R, Dieci G (2011) Promoter architectures in the yeast ribosomal expression program. Transcription 2(2):71–77
15. Buffa FM et al (2011) microRNA-associated progression pathways and potential therapeutic targets identified by integrated mRNA and microRNA expression profiling in breast cancer. Can Res 71(17):5635
16. Buffa FM, Harris AL, West CM, Miller CJ (2010) Large meta-analysis of multiple cancers reveals a common, compact and highly prognostic hypoxia metagene. Br J Cancer 102:428–435
17. Cabassi A, Kirk PDW (2020) Multiple kernel learning for integrative consensus clustering of omic datasets. Bioinformatics (In press)
18. Camps C et al (2014) Integrated analysis of microRNA and mRNA expression and association with HIF binding reveals the complexity of microRNA expression regulation under hypoxia. Molec Cancer 13:28
19. Chaudhary K, Poirion OB, Lu L, Garmire LX (2018) Deep learning-based multi-omics integration robustly predicts survival in liver cancer. Clin Cancer Res 24(6):1248–1259
20. Cheerla A, Gevaert O (2019) Deep learning with multimodal representation for pancancer prognosis prediction. Bioinformatics 35:i446–i454
21. Chen EY et al (2013) Enrichr: interactive and collaborative HTML5 gene list enrichment analysis tool. BMC Bioinform 14:128
22. Chen X et al (2014) XBP1 promotes triple-negative breast cancer by controlling the HIF1α pathway. Nature 508(7494):103–107
23. Chin SL, Marcus IM, Klevecz RR, Li CM (2012) Dynamics of oscillatory phenotypes in Saccharomyces cerevisiae reveal a network of genome-wide transcriptional oscillators. FEBS J 279(6):1119–1130
24. Chumnanpuen P, Nookaew I, Nielsen J (2013) Integrated analysis, transcriptome-lipidome, reveals the effects of INO-level (INO2 and INO4) on lipid metabolism in yeast. BMC Syst Biol 7(Suppl 3):S7
25. Ciriello G et al (2015) Comprehensive molecular portraits of invasive lobular breast cancer. Cell 163(2):506–519
26. Curtis C et al (2012) The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. Nature 486:346–352
27. Danila FR et al (2018) Multiple mechanisms for enhanced plasmodesmata density in disparate subtypes of C4 grasses. J Exp Bot 69(5):1135–1145
28. Dikicioglu D et al (2011) How yeast re-programmes its transcriptional profile in response to different nutrient impulses. BMC Syst Biol 5:148, 163
29. Drobna E et al (2012) Overexpression of the YAP1, PDE2, and STB3 genes enhances the tolerance of yeast to oxidative stress induced by 7-chlorotetrazolo[5,1-c]benzo[1,2,4]triazine. FEMS Yeast Res 12:958–968
30. Edgar R, Domrachev M, Lash AE (2002) Gene expression omnibus: NCBI gene expression and hybridization array data repository. Nucl Acids Res 30(1):207–210
31. Eisen MB, Spellman PT, Brown PO, Botstein D (1998) Cluster analysis and display of genome-wide expression patterns. PNAS 95(25):14863–14868
32. Elvidge GP et al (2006) Concordant regulation of gene expression by hypoxia and 2-oxoglutarate-dependent dioxygenase inhibition: the role of HIF-1alpha, HIF-2alpha, and other pathways. J Biol Chem 281(22):15215–15226
33. Emms DM, Kelly S (2015) OrthoFinder: solving fundamental biases in whole genome comparisons dramatically improves orthogroup inference accuracy. Genome Biol 16:157
34. Enright AJ, Van Dongen S, Ouzounis CA (2002) An efficient algorithm for large-scale detection of protein families. Nucl Acids Res 30(7):1575–1584

35. Ferreira RT et al (2012) Arsenic stress elicits cytosolic Ca(2+) bursts and Crz1 activation in Saccharomyces cerevisiae. Microbiology 158(Pt 9):2293–2302
36. Foo M et al (2018) A framework for engineering stress resilient plants using genetic feedback control and regulatory network rewiring. ACS Synth Biol 7(6):1553–1564
37. Ge H et al (2010) Comparative analyses of time-course gene expression profiles of the long-lived sch9Delta mutant. Nucl Acids Res 38(1):143–158
38. González-Aguilera C et al (2011) Nab2 functions in the metabolism of RNA driven by polymerases II and III. Mol Biol Cell 22(15):2729–2740
39. Gosset G (2017) Engineering of microorganisms for the production of chemicals and biofuels from renewable resources, 1st edn. Springer International Publishing, Cham
40. Harris AL (2002) Hypoxia—a key regulatory factor in tumour growth. Nat Rev Cancer 2:38–47
41. Harris B, Barberis A, West C, Buffa F (2015) Gene expression signatures as biomarkers of tumour hypoxia. Clin Oncol 27(10):547–560
42. Kohonen T (1982) Self-organized formation of topologically correct feature maps. Biol Cybern 43(1):59–69
43. Koritzinsky M et al (2013) Two phases of disulfide bond formation have differing requirements for oxygen. J Cell Biol (JCB) 203(4):615–627
44. Kovacs LAS et al (2012) Cyclin-dependent kinases are regulators and effectors of oscillations driven by a transcription factor network. Mol Cell 45(5):669–679
45. Krutilina R et al (2014) MicroRNA-18a inhibits hypoxia-inducible factor 1α activity and lung metastasis in basal breast cancers. Breast Cancer Res 16:R78
46. Kuleshov MV et al (2016) Enrichr: a comprehensive gene set enrichment analysis web server 2016 update. Nucl Acids Res 44(W1):W90–W97
47. Lai L-C et al (2011) Down-regulation of NDRG1 promotes migration of cancer cells during reoxygenation. PLoS One 6(8):e24375
48. Langfelder P, Horvath S (2008) WGCNA: an R package for weighted correlation network analysis. BMC Bioinform 9:559
49. Lanza AM, Blazeck JJ, Crook NC, Alper HS (2012) Linking yeast Gcn5p catalytic function and gene regulation using a quantitative, graded dominant mutant approach. PLoS One 7(4):e36193
50. Larsson M et al (2013) Functional studies of the yeast med5, med15 and med16 mediator tail subunits. PLoS One 8(8):e73137
51. Lee JS et al (2010) Negative regulation of hypoxic responses via induced Reptin methylation. Mol Cell 39(1):71–85
52. Liu Z et al (2013) Anaerobic α-amylase production and secretion with fumarate as the final electron acceptor in Saccharomyces cerevisiae. Appl Environ Microbiol 79(9):2962–2967
53. Lu X et al (2010) In vivo dynamics and distinct functions of hypoxia in primary tumor growth and organotropic metastasis of breast cancer. Can Res 70(10):3905–3914
54. MacQueen J (1967) Some methods for classification and analysis of multivariate observations. University of California Press, pp 281–297
55. Matia-González AM, Rodríguez-Gabriel MA (2011) Slt2 MAPK pathway is essential for cell integrity in the presence of arsenate. Yeast 28(1):9–17
56. Miller LD et al (2005) An expression signature for p53 status in human breast cancer predicts mutation status, transcriptional effects, and patient survival. PNAS 102(38):13550–13555
57. Mole DR et al (2009) Genome-wide association of hypoxia-inducible factor (HIF)-1alpha and HIF-2alpha DNA binding with expression profiling of hypoxia-inducible transcripts. J Biol Chem 284:16767–16775
58. Morillo-Huesca M, Clemente-Ruiz M, Andújar E, Prado F (2010) The SWR1 histone replacement complex causes genetic instability and genome-wide transcription misregulation in the absence of H2A.Z. PLOS One 5(8):e12143
59. Nguyen PA et al (2019) Phenotypes associated with genes encoding drug targets are predictive of clinical trial side effects. Nat Commun 10:1579
60. Orlando DA et al (2008) Global control of cell-cycle transcription by coupled CDK and network oscillators. Nature 453:944–947

61. Ortiz-Barahona A et al (2010) Genome-wide identification of hypoxia-inducible factor binding sites and target genes by a probabilistic model integrating transcription-profiling data and in silico binding site prediction. Nucl Acids Res 38(7):2332–2345
62. Paraskevopoulou S, Dennis AB, Weithoff G, Tiedemann R (2020) Temperature-dependent life history and transcriptomic responses in heat-tolerant versus heat-sensitive Brachionus rotifers. Sci Rep 10:13281
63. Parreiras LS, Kohn LM, Anderson JB (2011) Cellular effects and epistasis among three determinants of adaptation in experimental populations of Saccharomyces cerevisiae. Eukaryot Cell 10(10):1348–1356
64. Sanz AB et al (2012) Chromatin remodeling by the SWI/SNF complex is essential for transcription mediated by the yeast cell wall integrity MAPK pathway. Mol Biol Cell 23(14):2805–2817
65. Schneider P et al (2020) Rethinking drug design in the artificial intelligence era. Nat Rev Drug Discovery 19:353–364
66. Schödel J et al (2011) High-resolution genome-wide mapping of HIF-binding sites by ChIP-seq. Blood 117(23):e207–e217
67. Semenza GL (2014) Oxygen sensing, hypoxia-inducible factors, and disease pathophysiology. Ann Rev Pathol 9:47–71
68. Shen C, Kaelin WGJ (2013) The VHL/HIF axis in clear cell renal carcinoma. Semin Cancer Biol 23(1):18–25
69. Strassburg K et al (2010) Dynamic transcriptional and metabolic responses in yeast adapting to temperature stress. OMICS 14(3):249–259
70. Suzuki T, Iwahashi Y (2011) Gene expression profiles of yeast Saccharomyces cerevisiae sod1 caused by patulin toxicity and evaluation of recovery potential of ascorbic acid. J Agric Food Chem 59(13):7145–7154
71. Suzuki T, Iwahashi Y (2012) Comprehensive gene expression analysis of type B trichothecenes. J Agric Food Chem 60(37):9519–9527
72. Tang X et al (2012) Functional interaction between responses to lactic acidosis and hypoxia regulates genomic transcriptional outputs. Can Res 72(2):491–502
73. Tellaroli P et al (2016) Cross-clustering: A partial clustering algorithm with automatic estimation of the number of clusters. PLoS One 11(3):
74. The Cancer Genome Atlas Network (2012) Comprehensive molecular portraits of human breast tumors. Nature 490(7418):61–70
75. Wade SL, Poorey K, Bekiranov S, Auble DT (2009) The Snf1 kinase and proteasome-associated Rad23 regulate UV-responsive gene expression. EMBO J 28(19):2919–2931
76. Wang Y et al (2005) Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer. Lancet 365(9460):671–679
77. Xia X, Kung AL (2009) Preferential binding of HIF-1 to transcriptionally active loci determines cell-type specific response to hypoxia. Genome Biol 10:R113
78. Xue-Franzén Y, Henriksson J, Bürglin TR, Wright AP (2013) Distinct roles of the Gcn5 histone acetyltransferase revealed during transient stress-induced reprogramming of the genome. BMC Genom 14:479
79. Yang J et al (2010) The histone demethylase JMJD2B is regulated by estrogen receptor alpha and hypoxia, and is a key mediator of estrogen induced growth. Can Res 70(16):6456–6466

Chapter 5
Artificial Intelligence for Drug Development

Muhammad Waqar Ashraf

Abstract Drugs are treated as life-saving medicines against life-threatening diseases. However, drug developments pass through very complex and closely monitored phases to ensure the safety and efficacy of the intended purpose. The efforts are to keep highly toxic drugs from reaching even clinical trials. Even after the approval for drug distribution in the market, the drug's post-marketing safety is analyzed by the number of reported Adverse Events (AEs). It requires the analysis and interpretation of massive data in all three stages namely pre-clinical, clinical and post-marketing stages. In this article, we explore the use of Artificial Intelligence (AI) in interpreting the huge data that is generated in the pre-clinical and clinical trials for safety purposes.

5.1 Introduction

The use of computers is not new in the field of Chemistry. It has been used in drug development, for fertilizers of a crop, immunization programs through vaccinations in human beings and many available cures of diseases. From the molecular structures to the delivery of a drug in the market, there are several stages where the safety and efficacy of the drug need to be testified. All these steps involve massive data that needs to be analyzed fully and their correlation with each other has to be interpreted rightly in a limited time. In the era before computer involvement in computation, all of these steps were done by an expert and experienced chemist. But many linear/nonlinear patterns in that either were overlooked or not rightly interpreted by them. As the computational powers are increased, many methods have been developed to analyze and interpret the data correctly. Drug development for the treatment of disease needs extra careful examination to avoid any harm to human health. It generally has two major stages: (1) clinical trials and (2) post-market analysis. Even before the clinical trial, there are various testings performed on animals

M. W. Ashraf (✉)
Department of Mathematics and Natural Sciences, Prince Mohammad Bin Fahd University, Al Khobar 31952, Saudi Arabia
e-mail: mashraf@pmu.edu.sa

T. D. Pham et al. (eds.), *Advances in Artificial Intelligence, Computation, and Data Science*, Computational Biology 31,
https://doi.org/10.1007/978-3-030-69951-2_5

to keep the toxicities from reaching clinical trials. These practices are essential to ensure the safety of clinical trials in human beings. Clinical trials make sure that the drug works well for its intended purpose safely. Post-market surveillance is generally analyzed by reported Adverse Events (AE). Unexpected toxicities are the major factor of mortalities during clinical trials. Even some drugs have to be pulled off from the market due to unexpected toxicities in post-market surveillance. Moreover, drug development faces other challenges, too, like the associated cost of the process [1], complexity of analyzing the safety data [2], etc. There are huge data involved in all of these processes. It, therefore, is of utmost value to analyze the data and interpret the results correctly during each stage. Artificial intelligence (AI) plays a very important role in deciding the optimizing values of these parameters (e.g. molecular structure, associated cost, safety in clinical trials and post-market safety record) in the study of drug development [3–5]. Machine learning (ML) [6–8] is a data-driven approach. It produces quite efficient results in analyzing various parameters related to safety and the toxicity of medicine that is generally overlooked due to rare AEs. AI application in analyzing vast data of the spread of a disease and clinical trials in drug development has saved time in the development process of drugs and improved their prediction and efficacy [9, 10]. These quick results give the decision-makers ample time to be ready in time in a health emergency (pandemic, outbreak of a disease, etc.) and make the right decisions to save lives. AI can be used to find the best treatment strategies based on methods of diagnosis, drug and associated efficacy, safety and least or no AEs [11]. A fast diagnosis of the COVID-19 pandemic can be made by using AI in medical imaging. AI can pick the nonlinearities in the patterns without the requirement of human involvement.

A significant research field that can benefit from implementing modern machine learning and AI-based methods is the study of huge chemical datasets. The NCI Open Database [12], published in 1999 and containing around 250,000 molecules, was the only more extensive public chemical dataset available for several years. The emergence of PubChem [13] and later ChEMBL [14] databases significantly increased the quantity of chemical data accessible to the public for model training and validation purposes. At present, PubChem has more than 100 million specific compounds. In its latest 26th release, ChEMBL holds data on almost 2 million compounds, 13,000 targets and 16 million connections between these compounds and targets. The ZINC database [15], which provides information on over 230 million currently accessible compounds, is another valuable source of public chemical data. These three database channels provide a user-friendly web user interface, as the data can be retrieved and locally analyzed. Several new analysis and visualization tools have also been developed [16, 17]. The quantity of available data has recently increased by different orders of magnitude through two new experimental advances. DNA-based library synthesis [18] is one of those technologies where tens or even hundreds of millions of molecules can be found in a single library. Today, chemical databases contain large quantities of usable data but not all of these are publicly available. Their advantages include many data points containing many chemical information types, including toxicity, statistical results and even metadata [19].

The approaches based on AI and machine learning can play an essential role in navigating vast chemical spaces and independently converging on the right areas. Vernek et al. [20] presented the AI approaches based on Generative Topographic Mapping, a sophisticated dimensionality reduction technique used to compare in the company database of a major pharmaceutical company with more than 8 million commercially available samples. In this study, the accuracy of the approaches was increased by using the AutoZoom function, which targets and automatically extracts the data from the populated regions of chemical space. The technique was used to classify commercial molecules collections that optimize the chemical space covered by the molecules already accessible in the company archive under investigation. Such approaches allow compound sets to be adaptively enriched. Tetko et al. [21] identify a centered library generator capable of producing molecules with a greater probability of displaying the desired characteristics. The generator is based on the recurrent deep neural network of long short-term memory (LSTM) with outcomes guided to a particular target by the reinforcement learning process.

The community expects more increases in available chemical matter with these advances in mind, so we are likely to experience datasets with several billion compound structures in the next century. Classical methods of computational chemistry always struggle with such very large datasets, while some major advancements are promising [22, 23].

Below, we discuss the literature of artificial intelligence (AI) methods, both ML and DL, on new collected huge data for pre-clinical and post-marketing observations.

5.2 Methodologies in Pre-clinical and Clinical Trials

Quantitative Structure-Activity Relationship (QSAR) is a method that establishes a quantitative relationship between the molecular structure and pharma activity. QSARM has first been reported by Hanch and Fujita [24] and Free and Wilson [25]. Hanch and Fujita [24] have performed a regression analysis of molecular structure on biological activity. Motivated by the study, many researchers have used the regression analysis model of chemical structures and their characteristic properties. Moreover, various drug safety points can be modeled and rightly interpreted by the QSAR model [26, 27]. In particular, a good QSAR model can analyze the correlation between molecular structure and biological activity.

Early QSAR methods have used multivariate linear regression to analyze the drug's chemical properties [28]. Due to the high dimensionality of data, many of these methods follow certain data points more closely and don′t give a good fit. It, therefore, interprets the results wrongly. To overcome these limitations, regression models have been developed. The linear regression model consists of biological activity (y) as a dependent variable and molecular descriptor of the drug as a dependent variable (x). So the prediction of biological property can be mathematically represented by $y = mx + c$. The slope of above line "m" is called the Regression Coefficient.

In the multiple linear regression (MLR) model [29], there will be more than one molecular descriptor of the drug, and the property of the drug is determined by more than one descriptor. Regression-based models have significantly improved the interpretations of a QSAR model but still, it has limitations. High data dimensionality and assumption of linearity in regression models have restricted the use of QSAR models to limited tasks. Support Vector Machines (SVMs) [30] are used as an alternative to interpreting the results than QSAR models better.

In SVMs, one finds a hyperplane in n-dimensional vector space. Here n refers to a number of features. SVMs classify the data into two non-overlapping categories. Wesley et al. [31] have reported that SVMs perform better than QSAR models. SVMs can also be used in combination with LR/MLR models for better prediction of biological properties. Recently, Nekoei [32] has combined a generic algorithm with SVM to develop a QSAR model of the novel 4-aminopyrimidine-5-carbaldehyde oxime derivatives as effective and selective inhibitors of potent VEGFR-2.

Artificial Intelligence uses an artificial neural network (ANN), which consists of three layers, namely input, intermediate and output layers. The input layer carries the molecular structure, which is processed in the intermediate layer, and drug property is received as an output. The ANN model is quite useful to highly variable datasets [33].

5.3 Post-Market Trials

Phocomelia in newborns was often thought of due to genetic inheritance before it was revealed in 1962 by post-market trials. The disease is due to a commonly prescribed sleeping pill given to pregnant women [34]. It shows that pre-clinical and clinical trials are not full proof and there may be a chance of long-term negative effects on human health. Therefore post-market trials are very important to remove toxic drugs from the market. The post-market surveillance is done based on reported Adverse Events (AEs). FDA in the United States maintains a database for reporting AEs. However, it faces various challenges in terms of under-reporting and interpretation of the dataset. To overcome the challenge, a new database and sophisticated computational techniques are required. The database should contain the adverse effects of the drug on target and off-target. AI is beneficial to meet the challenge. Lorberbaum et al. [35] proposed a modular assembly of drug safety subnetwork (MADSS) that combines various biological and chemical data sources into a common network and identifies an AE-module. The AE-module, along with the network, is used to find out the correlation of drugs with reported AEs.

5.4 Concluding Remarks

AI has attracted the attention of researchers due to its usefulness in a variety of research fields, including chemistry and biology. AI has made it easy to study the correlation between various parameters and find out the nonlinear behavior, if any. The huge data generated in pre-clinical, clinical and post-clinical trials can be interpreted without human interference quickly. AI can also correlate the medical history of patients with the reported AEs to find out the shortcomings of trials. This is a work in progress and many correlations (genetically, lifestyle, etc.) are still to be explored using AI in predicting AEs.

References

1. Adams CP, Brantner VV (2010) Spending on new drug development 1. Health Econ 19(2):130–141
2. Xia HA, Jiang Q (2014) Statistical evaluation of drug safety data. Ther Innov Regul Sci 48(1):109–120
3. Mak K-K, Pichika MR (2019) Artificial intelligence in drug development: present status and future prospects. Drug Discovery Today 24(3):773–780
4. Lamberti MJ, Wilkinson M, Donzanti BA, Wohlhieter GE, Parikh S, Wilkins RG, Getz K (2019) A study on the application and use of artificial intelligence to support drug development. Clin Ther 41(8):1414–1426
5. Abdul Rashid MBM (2020) Artificial intelligence effecting a paradigm shift in drug development. SLAS Technol: Transl Life Sci Innov 20:3–15
6. Murphy RF (2011) An active role for machine learning in drug development. Nature Chem Biol 7(6):327–330
7. Wale N (2011) Machine learning in drug discovery and development. Drug Dev Res 72(1):112–119
8. Meng H-Y, Jin W-L, Yan C-K, Yang H (2019) The application of machine learning techniques in clinical drug therapy. Curr Comput Aided Drug Des 15(2):111–119
9. Gunther EC, Stone DJ, Gerwien RW, Bento P, Heyes MP (2003) Prediction of clinical drug efficacy by classification of drug-induced genomic expression profiles in vitro. Proc Natl Acad Sci 100(16):9608–9613
10. Li H, Zhang W, Chen Y, Guo Y, Li G-Z, Zhu X (2017) A novel multi-target regression framework for time-series prediction of drug efficacy. Sci Rep 7(1):1–9
11. Silva Á, Cortez P, Santos MF, Gomes L, Neves J(2006) Mortality assessment in intensive care units via adverse events using artificial neural networks. Artif Intell Med 36(3):223–234
12. Hohne BA, Pierce TH (1989) Expert system applications in power systems. Am Chem Soc 408
13. Hanessian S, Franco J, Gagnon G, Laramee D, Larouche B (1990) Computer-assisted analysis and perception of stereochemical features in organic molecules using the chiron program. J Chem Inf Comput Sci 30(4):413–425
14. Hanessian S, Botta M, Larouche B, Boyaroglu A (1992) Computer-assisted perception of similarity using the chiron program: a powerful tool for the analysis and prediction of biogenetic patterns. J Chem Inf Comput Sci 32(6):718–722
15. Hanessian S (1993) Reflections on the total synthesis of natural products: art, craft, logic, and the chiron approach. Pure Appl Chem 65(6):1189–1204
16. Wipke WT, Rogers D (1984) Artificial intelligence in organic synthesis. SST: starting material selection strategies. An application of superstructure search. J Chem Inf Comput Sci 24(2):71–81

17. Azario P, Barone R, Chanon M (1988) Microcomputer and organic synthesis. 3. The MARSEIL/SOS expert system, a new graphic approach. An electronic lab note for organic synthesis. J Org Chem 53(4):720–724
18. Mehta G, Barone R, Azario P, Barberis F, Arbelot M, Chanon M (1992) New computer-based approach for seeking a key step in the synthesis of complex structures. Application to taxane and crinipellin diterpenoid frameworks. Tetrahedron 48(41):8953–8962
19. Feller D (1996) The role of databases in support of computational chemistry calculations. J Comput Chem 17(13):1571–1586
20. Huang Q, Li L-L, Yang S-Y (2011) RASA: a rapid retrosynthesis-based scoring method for the assessment of synthetic accessibility of drug-like molecules J Chem Inf Model 51(10):2768–2777
21. Salatin TD, Jorgensen WL (1980) Computer-assisted mechanistic evaluation of organic reactions. 1. Overview. J Org Chem 45(11):2043–2051
22. Goh GB, Hodas NO, Vishnu A (2017) Deep learning for computational chemistry. J Comput Chem 38(16):1291–1307
23. Korotcov A, Tkachenko V, Russo DP, Ekins S (2017) Comparison of deep learning with multiple machine learning methods and metrics using diverse drug discovery data sets. Mol Pharm 14(12):4462–4475
24. Hansch C, Fujita T (1995) Classical and three-dimensional QSAR in agrochemistry. ACS Publications, Washington, DC
25. Free SM, Wilson JW (1964) A mathematical contribution to structure-activity studies. J Med Chem 7(4):395–399
26. Luco JM, Ferretti FH (1997) QSAR based on multiple linear regression and PLS methods for the anti-HIV activity of a large group of HEPT derivatives. J Chem Inf Comput Sci 37(2):392–401
27. Tibshirani R (1996) Regression shrinkage and selection via the lasso. J Roy Stat Soc: Ser B (Methodol) 58(1):267–288
28. Ferreira MMC (2016) Multivariate QSAR. In: Encyclopedia of physical organic chemistry. Wiley, Hoboken, pp 1–38
29. Chakraborty A, Goswami D (2017) Prediction of slope stability using multiple linear regression (MLR) and artificial neural network (ANN). Arab J Geosci 10(17):385
30. Hearst MA, Dumais ST, Osuna E, Platt J, Scholkopf B (1998) Support vector machines. IEEE Intell Syst Appl 13(4):18–28
31. Wesley L, Veerapaneni S, Desai R, Mcgee F, Joglekar N, Rao S, Kamal Z (2016) 3d-QSAR and SVM prediction of braf-v600e and HIV integrase inhibitors: a comparative study and characterization of performance with a new expected prediction performance metric. Am J Biochem Biotechnol 12(4):253–262
32. Nekoei M, Mohammadhosseini M, Pourbasheer E (2015) QSAR study of vegfr-2 inhibitors by using genetic algorithm-multiple linear regressions (GA-MLR) and genetic algorithm-support vector machine (GA-SVM): a comparative approach. Med Chem Res 24(7):3037–3046
33. Agatonovic-Kustrin S, Beresford R (2000) Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. J Pharm Biomed Anal 22(5):717–727
34. Taussig HB (1962) A study of the German outbreak of phocomelia. Obstet Gynecol Surv 17(6):840–844
35. Lorberbaum T, Nasir M, Keiser MJ, Vilar S, Hripcsak G, Tatonetti NP (2015) Systems pharmacology augments drug safety surveillance. Clin Pharm Therap 97(2):151–158

Chapter 6
Mathematical Bases for 2D Insect Trap Counts Modelling

Danish A. Ahmed, Joseph D. Bailey, Sergei V. Petrovskii, and Michael B. Bonsall

Abstract Pitfall trapping is a predominant sampling method in insect ecology, invasive species and agricultural pest management. Once samples are collected, their content is analyzed, different species are identified and counted and then used to provide reliable estimates of relative population abundance. Such estimates are essential for a variety of reasons, such as the general survey of insect diversity, detection of new insect invasions or simply for monitoring population levels. However, interpreting trap counts is a challenging task, since captures can depend on a variety of factors, such as environmental conditions, trap or survey design, insect movement behaviour, etc. Mathematical models provide an extremely useful description of how insects move in the field and in turn, can simulate the trapping process. In this chapter, we present the mathematical bases for 2D insect trap counts modelling, at the mean-field level using the diffusion equation and on an individual level using random walks. We reveal the intricacies of the trap counts dynamics, with details on how trap geometries and movement types can affect captures. We also describe the mathematical details for other trapping methods, such as baited trapping, where an attractant is used to lure the insects towards the trap location.

D. A. Ahmed (✉)
Center for Applied Mathematics and Bioinformatics, Department of Mathematics and Natural Sciences, Gulf University for Science and Technology, Hawally, Kuwait
e-mail: Ahmed.D@gust.edu.kw

J. D. Bailey
Department of Mathematical Sciences, University of Essex, Colchester, UK

S. V. Petrovskii
School of Mathematics and Actuarial Science, University of Leicester, University road, Leicester, UK

M. B. Bonsall
Mathematical Ecology Research Group, Department of Zoology, University of Oxford, Oxford, UK

T. D. Pham et al. (eds.), *Advances in Artificial Intelligence, Computation, and Data Science*, Computational Biology 31,
https://doi.org/10.1007/978-3-030-69951-2_6

6.1 Introduction

Insect trapping is central to many ecological studies [1, 2]. For ground-dwelling (surface-active) insects, trap count samples are almost entirely collected with pitfall traps [3]. These counts are then manipulated to provide key ecological information on activity patterns, spatial distributions, total population abundance etc [4–7]. The interpretation of trap counts is a notoriously difficult task, mainly due to the fact that captures can be heavily influenced by many factors, for example, experimental design, insect movement behaviour, individual differences, weather conditions, habitat type etc [8–11]. For some time now, field ecologists have highlighted that since population estimates are drawn from the analysis of samples and, in some cases used to inform control action policy, a better understanding of trap counts dynamics is required [12].

Mathematical models and simulations provide an effective approach to investigate trap counts [13, 14]. Simulations are cost effective, easy to replicate, and can provide useful theoretical insights, that would normally be difficult to obtain from real field experiments [15]. In this chapter, we present the mathematical bases behind the modelling of 2D insect movement and trap counts.

- In Sect. 6.2, we describe the simplest models of insect movement in an isotropic environment, using two different yet equivalent approaches. The mean-field approach using the diffusion equation describes the spatio-temporal population density, and the random walk framework provides a mechanistic model for individual insect movement. We also present the mathematical details of how to compute trap counts.
- In Sect. 6.3, we focus more on the intricacies of trap counts based on geometrical aspects. We investigate simulation artefacts, and how the shape or size of the trap or arena can impact trap counts.
- In Sect. 6.4, we provide the reader with an overview of more complicated movement models, which more closely describe real insect movement. These models allow for short-term and/or long-term directional persistence in the movement. We present their mathematical properties and summarize those metrics that are useful for the analysis of movement paths.
- In Sect. 6.5, we reveal how captures can be affected by the type of movement behaviour, and the mechanisms involved behind baited trapping, where insects are receptive to an attractant, and therefore, are lured towards the trap.

A better understanding of the trap counts dynamics contributes towards improved trap count interpretations, and has critical implications for spatial ecology and for understanding the distribution and abundance of insect species. We hope that this chapter will serve as an introductory read for ecologists, entomologists and field experimentalists, who may be interested in the mathematical or theoretical aspects of trap counts modelling.

6.2 Mean Field and Mechanistic Models of Insect Movement with Trapping

6.2.1 Isotropic Diffusion Model and Computing Trap Counts

Consider a population of insects browsing within a field with a single trap located at the centre. At a simple level, if the environment is homogenous (isotropic) and the insects are considered to be identical and moving randomly, then the population density u can be modelled using the diffusion equation [16–18]

$$\frac{\partial u}{\partial t} = D\nabla^2 u \tag{6.1}$$

where ∇^2 is the Laplacian operator, $u = u(\mathbf{x}, t)$ is a function of space $\mathbf{x} = (x, y)$ and time t, and D is the (constant) diffusion coefficient. Alongside this movement description, information is required related to the trap/field geometry, and how the insects are distributed across space [13, 19]. For the choice of trap shape, we consider a circular pitfall trap, which is the most frequently used in field studies, and also shown to be the most capture efficient [8, 20]. Also, we consider the arena boundary to be of circular shape and impenetrable, so that the whole population is confined at all times. The choice of the arena boundary shape or size is not important, as long as it is sufficiently larger than the trap size, see later Sect. 6.3.2 for more details.

Following these geometrical considerations, we use polar co-ordinates to describe spatial location

$$x = r\cos\theta, \quad y = r\sin\theta, \quad r > 0, \quad -\pi < \theta \le \pi, \tag{6.2}$$

with inverse

$$r = \sqrt{x^2 + y^2}, \quad \theta = \text{atan}_2\,(y, x)\,, \tag{6.3}$$

where $\text{atan}_2\,(y, x)$ is equal to $\arctan\left(\frac{y}{x}\right)$ for $x \ge 0$ and to $\arctan\left(\frac{y}{x}\right) \pm \pi$ for $x < 0$. The diffusion equation in (6.1) can then be expressed as

$$\frac{\partial u}{\partial t} = D\left(\frac{1}{r}\frac{\partial u}{\partial r} + \frac{\partial^2 u}{\partial r^2}\right) \tag{6.4}$$

where we have assumed that the diffusion is everywhere radial and thus the population density $u = u(r, t)$ is axisymmetric i.e. independent of the angle θ. The domain of the annular arena can be formally written as

$$\mathfrak{D}_{\text{arena}} = \{(r, \theta) : a < r < R, -\pi < \theta \le \pi\} \tag{6.5}$$

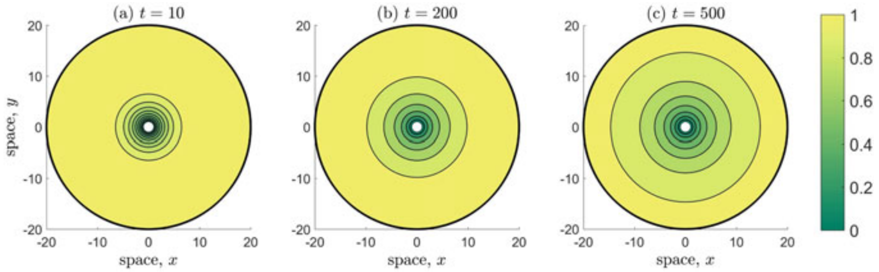


Fig. 6.1 Snapshots of the population density u. The diffusion equation (6.4) was solved numerically alongside the boundary conditions (6.6)–(6.7) with initial condition (6.8). Even for this simple geometry, alongside these conditions, an analytical solution is difficult to obtain and it may not always be instructive (e.g. being expressed as an infinite series of Bessel functions). Model parameters include: diffusion coefficient $D = 1$, trap radius $a = 1$, outer boundary radius $R = 20$, and initial density $u_0 = 1$ (corresponding to a population size $N_0 \approx 1254$). The colour bar represents density values

with trap radius a and outer boundary radius R. For the boundary conditions, we have the zero-density condition on the trap boundary

$$u = 0 \text{ for } r = a, \tag{6.6}$$

and the zero-flux condition on the outer boundary

$$\frac{\partial u}{\partial r} = 0 \text{ for } r = R. \tag{6.7}$$

If we assume that the population of insects is initially homogeneously distributed over space, then the initial density is constant

$$u(r, t = 0) = u_0. \tag{6.8}$$

The population size as a function of time $N(t)$ can be computed as

$$N(t) = 2\pi \int_a^R r u(r, t) dr, \tag{6.9}$$

with initial population size $N_0 = N(t = 0) = \pi(R^2 - a^2)$.

Figure 6.1 illustrates snapshots of the population density, which is zero inside the trap. Notice the perturbation that develops around the vicinity of the trap as the density decreases closer to the trap boundary due to trapping.

Trap counts $J(t)$ can be computed as the total diffusive flux through the circular trap boundary over time t [21, 22], as

$$J(t) = 2\pi D \int_0^t \left.\frac{\partial u}{\partial r}\right|_{r=a} dt'. \tag{6.10}$$

On assuming that there is no insect mortality or reproduction, the total number of individuals that remain on the arena and the trap counts must always sum to the total population size N_0, leading to the relation

$$N(t) + J(t) = N_0. \tag{6.11}$$

Similarly, one could apply these concepts to a more complicated trap shapes, for example, square shaped [13].

6.2.2 Individual Based Modelling Using Random Walks

The 2D curvilinear movement path of a single insect in the field $\mathbf{x}(t) = (x(t), y(t))$ can be modelled using a discrete time and continuous space random walk (RW). The path is mapped as a series of discrete steps linking an insects location $\mathbf{x}_i = \mathbf{x}(t_i)$, recorded at discrete times $t_i = \{t_0, t_1, t_2, \ldots\}$, and distance between any two successive steps as step lengths $l_i = |\mathbf{x}_i - \mathbf{x}_{i-1}| = \{l_1, l_2, l_3, \ldots\}$ with average velocity $\mathbf{v}_i = \frac{\mathbf{x}_i - \mathbf{x}_{i-1}}{\Delta t}$ and speed $v_i = |\mathbf{v}_i|$ [23]. Discrete time analyses of insect telemetry data often work with regular time steps, and therefore, we assume $t_i = i\Delta t$, with constant time increment $\Delta t = \frac{\mathbb{E}[l]}{v}$ independent of i, where $\mathbb{E}[l]$ is the mean step length and v is the mean speed. The total duration in an n step RW is simply given as $T = n\Delta t$.

If we consider an insect situated at location $\mathbf{x}_{i-1} = (x_{i-1}, y_{i-1})$ at time t_{i-1}, then the location at the next time step t_i can be expressed as

$$\mathbf{x}_i = \mathbf{x}_{i-1} + (\Delta\mathbf{x})_i, \quad i = 1, 2, 3, \ldots \tag{6.12}$$

where $(\Delta\mathbf{x})_i = (\Delta x_i, \Delta y_i)$ is a step vector whose components are random variables, for the i^{th} step along the walk. The 2D RW can be described in polar co-ordinates, by expressing the step vector in terms of step lengths l and step orientations (or headings) θ, using the transformation

$$\Delta x = l\cos\theta, \ \ \Delta y = l\sin\theta, \quad l \in [0, \infty), \ \ \theta \in (-\pi, \pi] \tag{6.13}$$

with inverse transformation

$$l = \sqrt{(\Delta x)^2 + (\Delta y)^2}, \quad \theta = \operatorname{atan}_2(\Delta y, \Delta x). \tag{6.14}$$

The turning angle α_i can then be measured as the difference between the orientations of two successive steps

$$\alpha_i = \theta_i - \theta_{i-1}. \tag{6.15}$$

On assuming that step lengths and step orientations are neither autocorrelated nor cross-correlated [24], the individual movement can be simulated once the distributions of step lengths $\lambda(l)$ and turning angles $\psi(\alpha)$ are prescribed.

The mean cosine c_α and the mean sine s_α, defined as

$$c_\alpha = \mathbb{E}[\cos\alpha] = \int_{-\pi}^{\pi} \cos(\alpha)\psi(\alpha)d\alpha, \quad s_\alpha = \mathbb{E}[\sin\alpha] = \int_{-\pi}^{\pi} \sin(\alpha)\psi(\alpha)d\alpha, \tag{6.16}$$

both lie between 0 and 1, and are useful statistical parameters that characterize the turning angle distribution $\psi(\alpha)$. A null mean sine $s_\alpha = 0$ corresponds to a balanced RW (i.e. left and right turns are equiprobable), in which case $\psi(\alpha)$ is centrally symmetric. The mean cosine c_α represents the correlation between the orientations of successive steps. A null mean cosine $c_\alpha = 0$ corresponds to completely random movement (known as a simple RW), and at the other extreme end, $c_\alpha = 1$ corresponds to straight line (or ballistic) movement [25].

6.2.3 Simple Random Walk (SRW)

The simple random walk (SRW) is isotropic (unbiased) which means that the individual is equally likely to move in each possible direction [17, 26, 27]. In this case, the distribution of turning angles $\psi(\alpha)$ is uniform and defined over the interval from $-\pi$ to π with null mean sine and cosine, i.e. a balanced uncorrelated RW. In general, the SRW provides an oversimplified description for insect movement, and usually serves as a theoretical baseline model for more complicated movement behaviours [28, 29]. However, under certain scenarios, such a model can provide an accurate description, for example, in the case of infected ants that behave like ‘zombies’, since their movement behaviour has been observed to be completely random, rather than directional walking [30].

The earliest successful modelling attempts to model movements of insects were entirely based on SRWs whilst considering Gaussian increments, which is a discrete time model of Brownian motion [13, 19]. For the remainder of this chapter, we choose to rely on this, so that the distribution for the components of the step vector $\Delta\mathbf{x}$ is:

$$\phi(\Delta x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(\Delta x)^2}{2\sigma^2}\right), \quad \phi(\Delta y) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(\Delta y)^2}{2\sigma^2}\right) \tag{6.17}$$

with mean $\mathbb{E}[\Delta x] = 0$ and variance $\mathrm{Var}[\Delta x] = \sigma^2$, and exact same expressions for Δy. Due to isotropicity, the variances are equal and depend on a single parameter σ which represents the mobility of the insect. Also, note that in the more general case of non-Gaussian increments, the basic requirements are that the distribution of each increment is symmetrical and zero-centered with finite variance.[1]

The corresponding probability distribution functions for step lengths and turning angles (l, α) are given by

$$\lambda(l;\sigma) = \frac{l}{\sigma^2}\exp\left(-\frac{l^2}{2\sigma^2}\right), \quad \psi(\alpha) = \frac{1}{2\pi}, \tag{6.18}$$

where $\lambda(l)$ is the Rayleigh distribution with mean step length and mean squared step length

$$\mathbb{E}[l] = \frac{\sigma\sqrt{2\pi}}{2}, \quad \mathbb{E}[l^2] = 2\sigma^2 \tag{6.19}$$

and $\psi(\alpha)$ is the uniform distribution, see [14] for a derivation of the equations in (6.18) from the Gaussian step increments in (6.17). Note that, a finite variance is always ensured if the step length distribution $\lambda(l)$ decays sufficiently fast at large l, and this holds for the Rayleigh distribution whose end tail decays faster than exponential.

6.2.4 Simulating Trapping

Recall that in Sect. 6.2.1 we considered a confined population of N_0 individuals, initially homogenously distributed on an annular arena $(a < r < R)$. For an individual based description, the initial location $\mathbf{x}_0$ of each individual is simulated as

$$\mathbf{x}_0 \sim \left(\sqrt{\left(R^2 - a^2\right)U + a^2},\ 2\pi U\right) \tag{6.20}$$

where U is a random variable drawn from the uniform distribution between 0 and 1. Note that, for an arbitrary trap shape or outer boundary, the arena no longer has an infinite number of symmetry axes, and therefore, initial locations can be drawn at random over the whole space within the outer boundary, and remove those occurring within the trap. Once the initial distribution is prescribed, the movement of each individual can be modelled independently by a SRW with Gaussian increments, and therefore, by extension, the distribution of the population in space [33]. The population is confined due to the impenetrable arena boundary, and therefore, we impose the 'no-go' condition, so that if an individual attempts to overstep the boundary, then an alternative step with a completely new random direction is chosen at the previous location [34].

[1] This is to avoid any resulting global biases in the movement path (Biased RW [27], see later Sect. 6.4.4) or the case of heavy tails (Lévy walks or flights [31, 32]).

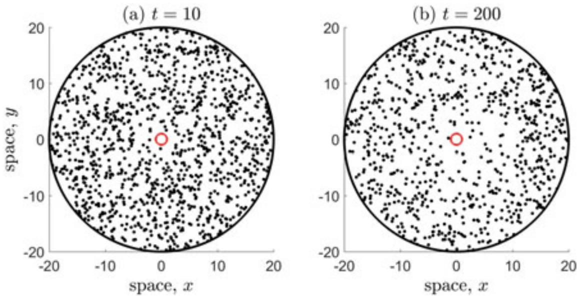


Fig. 6.2 Snapshots of the population spatial distribution with initial density $u_0 = 1$ (corresponding to a population size $N_0 \approx 1254$ individuals). Each individual performs a SRW with Gaussian increments (independently) with mobility parameter $\sigma = \sqrt{0.02} \approx 0.1414$. Trap and arena boundary radii are $a = 1$ and $R = 20$. Time is computed as $t = n\Delta t$ with time increment $\Delta t = 0.01$, where n is the maximum number of steps in the walk

We introduce the concept of trapping by stating that at each time t_i, those individuals whose location lies within the trap ($r < a$) are removed, and these counts are summed so that (accumulated) trap counts $J(t_i)$ can be computed [13, 20]. Trap counts can then be presented as a monotonously increasing stochastic trajectory in terms of discrete time, (see later, the dashed curve in Fig. 6.3). The effect of noise can be further decreased by averaging over many simulation runs, resembling a collection of multiple samples in the real field, or by increasing the population size N_0 (or equivalently the initial density u_0 given a fixed arena area).

Figure 6.2 illustrates snapshots of the spatial distribution of a confined population performing a SRW on an annulus with a circular trap installed at the centre. With time, the population decreases as a result of trapping. Compare this to Fig. 6.1, where the equivalent mean-field diffusion model was used.

6.2.5 Equivalent Trap Counts

It is well known that for a SRW, the governing mean-field equation is the diffusion equation, which can be explicitly derived from first principles [26, 27, 35]. A useful metric to analyze movement patterns is the mean squared displacement (MSD), which is defined as the expected value of the squared beeline distance between an individuals initial and final positions. For the SRW, the MSD is

$$\mathbb{E}\left[R_n^2\right] = n\mathbb{E}\left[l^2\right] \tag{6.21}$$

and can also be related to the diffusion coefficient D, as follows:

$$\mathbb{E}\left[R_n^2\right] = 4Dn\Delta t, \tag{6.22}$$

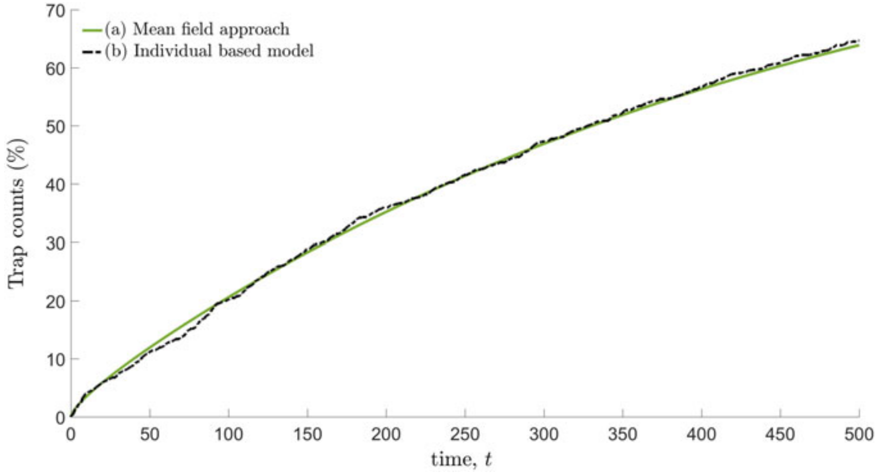


Fig. 6.3 Trap counts computed using: **a** mean-field approach as the total diffusive flux through the circular trap boundary, see Eq. (6.10), and **b** individual-based model where trap counts are simulated using the method described in Sect. 6.2.4. All details regarding arena dimensions, and movement parameters are the same as in the captions of Figs. 6.1 and 6.2. Note that the relation between the movement parameters σ and D satisfy Eq. (6.23), which links these different approaches

which illustrates that the MSD grows linearly with time [18, 27, 36]. On equating (6.21) and (6.22), and given that the mean squared step length for a SRW with Gaussian increments is $2\sigma^2$ from (6.19), one gets

$$\sigma^2 = 2D\Delta t. \tag{6.23}$$

This provides a direct relationship between the mobility parameter of individual movement σ and the diffusion coefficient D.

Figure 6.3 demonstrates that the trap counts are equivalent, whether we model insect movement at the individual level or using the corresponding mean-field description. This is a classical example of how these different modelling approaches are complementary.

6.3 Geometrical Considerations for Trap Counts Modelling

6.3.1 Simulation Artefacts Due to the RW Jump Process

The RW model described in Sect. 6.2.2 is essentially a position jump process, where only the location of an individual after each step is recorded, and any information regarding the movement path in between locations is lost [37, 38]. Under this con-

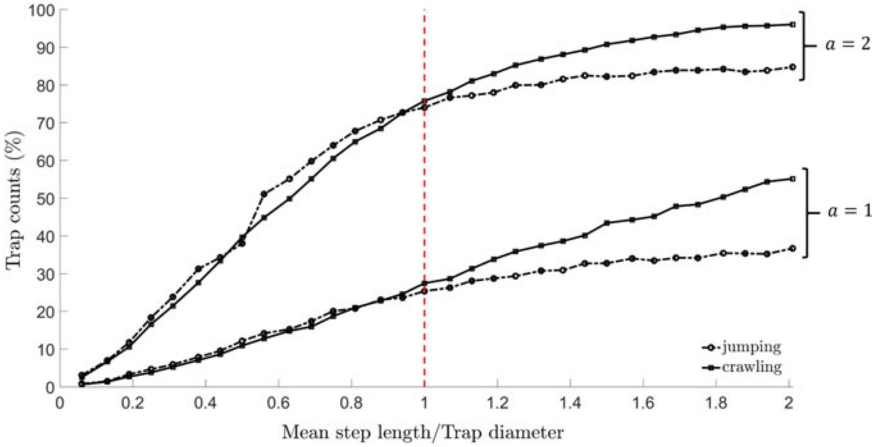


Fig. 6.4 Trap counts recorded at the end location ('jumping'), or using sub step lengths ('crawling') with a step length increment of $\Delta l = 0.1$. An annular arena is considered with outer boundary radius $R = 20$, and circular trap installed at the centre with radius $a = 1$ or $a = 2$. Each individual in the population performs a SRW with Gaussian increments, with σ varied, and executes a maximum number of $n = 200$ steps. The population density is chosen to be sufficiently large to reduce noise

struction, the line segment which links two subsequent locations may intersect with the trap boundary, and therefore, it is possible for an individual to 'jump' over the trap and go undetected—resulting in a null trap count. This is more likely to occur if an individual can execute longer steps, i.e. if the mobility parameter σ is large enough, as it is directly linked to the mean step length. Following this, one could expect a critical relation between σ and the characteristic scale length of the trap.

Suppose that an individual moves from location $\mathbf{x}_{i-1}$ to $\mathbf{x}_i$ with step length l_i. Rather than checking if the individual lies within the trap at each end location, we can check at intermediate points (sub step lengths) by considering step length increments of Δl

$$(\Delta l)l_i \quad \text{with} \quad 0 < \Delta l < 1. \tag{6.24}$$

This process could resemble an insect that persists to 'crawl' from $\mathbf{x}_{i-1}$ to $\mathbf{x}_i$, whilst maintaining the same direction of movement, but only execute a turn once it arrives at location $\mathbf{x}_i$.

Figure 6.4 demonstrates the critical value beyond which the trap count dynamics shift, precisely where the mean step length is equal to the circular trap diameter $\mathbb{E}(l) = 2a$, indicated by the vertical dashed line. If

$$\mathbb{E}[l] < 2a, \tag{6.25}$$

then the chances of the individual jumping over the trap are less likely, and therefore, the trap counts are similar. However, if $\mathbb{E}[l] \geq 2a$, the frequency of jumping over the trap is more likely, and as a result trap counts can be grossly underestimated. In the particular case of a SRW with Gaussian increments, we can derive an upper bound on the mobility parameter from Eq. (6.25), so that one gets

$$\sigma < \frac{4a}{\sqrt{2\pi}} \tag{6.26}$$

where the mean step length is given in Eq. (6.19).

6.3.2 Impact of the Arena Boundary Shape, Size and the Average Release Distance

The impact of the arena boundary shape and size on trap counts is negligible as long as it is sufficiently larger than the trap size. However, this raises the question: how much larger should it be so that the effect is not realized? To determine this, we consider the arena boundary to be of circular or square shape, which is usually the default choice in real field experiments, with a circular trap installed at the centre. Here, we consider the area of the arena to be equal (see Fig. 6.5), so that the insects browse over the same amount of space.

Figure 6.6 shows that if the arena area is greater than 5000 units^2, then the absolute trap count differences lie well below 1%, indicated by the horizontal dotted line. However, for much larger trap sizes these differences decay more slowly, and thus

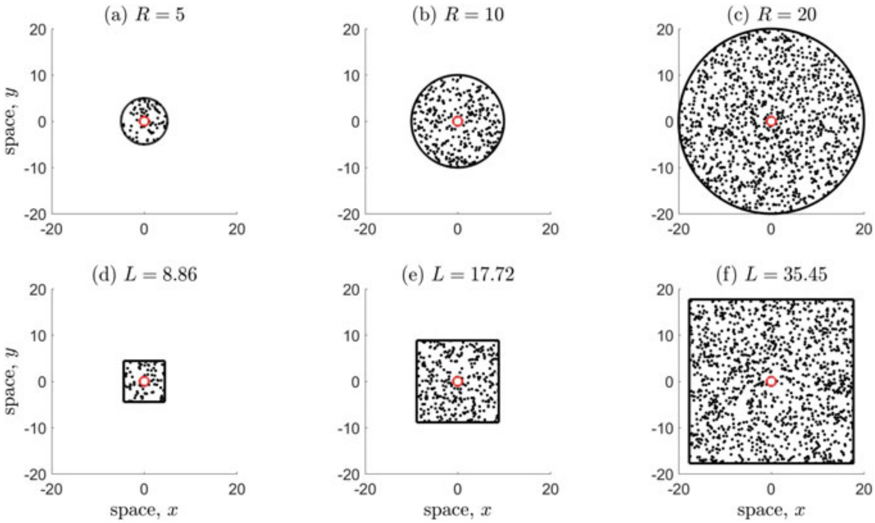


Fig. 6.5 Plots **a**–**c** Annular arena with increasing outer boundary radius $R = 5, 10, 20$. Plots **d**–**f** Square boundary with side lengths L. The arenas have the same area, and boundary dimensions are related by the equation $L = \sqrt{\pi}R$. In either case, a circular trap is placed at the centre with radius $a = 1$

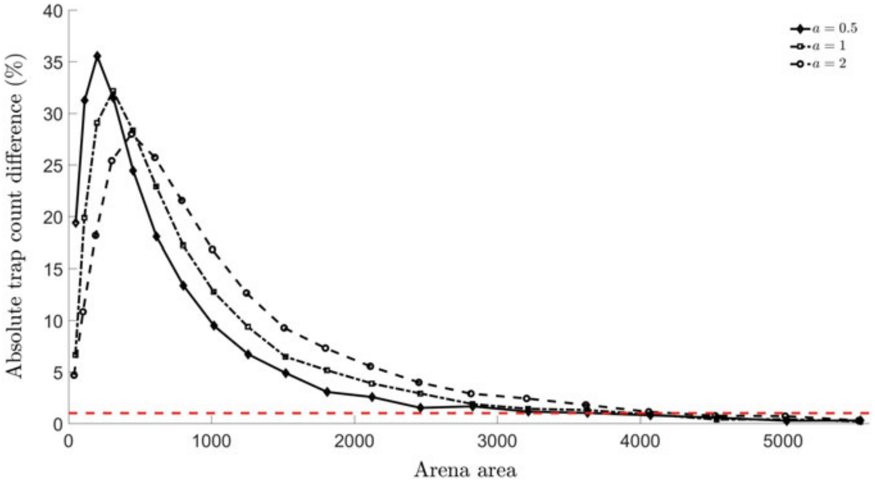


Fig. 6.6 Absolute trap count differences were computed between the set up described in Fig. 6.5. Circular traps of varying sizes were considered with radius $a = 0.5, 1, 2$. Each individual performed a SRW with Gaussian increments with $\sigma = 0.5$. Trap counts were computed at time $t = 20$, and the initial population density was fixed to ensure sufficient reduction of noise. Note that, for small arena sizes, trap counts accumulate much faster, even for a short amount of time, and therefore, trap count differences can vary significantly

requires a larger arena size to offset the impact on trap count differences. It follows that, in the case of an annular arena with a circular trap of unit radius, the outer boundary radius must be at least 40 fold, and under this condition, its shape is not that important.

This result also has some further implications, as the average initial release distance of the individuals (as measured from the centre of the trap), will be different depending on the shape of the outer boundary. One may naively assume that these differences can possibly explain a variation in trap counts. However, we have demonstrated that this depends on the relative size of the outer boundary, and if sufficiently larger than the trap size, any fundamental trap count differences that may emerge due to other factors (for example, different trap shapes or movement types, etc.), are actual and not a result of differences in the average initial release distance.

6.3.3 *Impact of Trap Shape*

To determine how trap counts may be impacted by the trap shape, we consider the circular, square and slot (rectangular) trap shapes with geometry $\mathfrak{D}$ defined as

(i) Circular trap with radius a

$$\mathfrak{D}_{\text{circle}} = \left\{(x, y) |\, x^2 + y^2 < a^2\right\} \tag{6.27}$$

(ii) Square trap with equal side lengths b

$$\mathfrak{D}_{\text{square}} = \left\{(x, y) \,\middle|\, |x| < \frac{b}{2},\ |y| < \frac{b}{2}\right\}, \tag{6.28}$$

(iii) Slot trap with base length b_1 and width b_2

$$\mathfrak{D}_{\text{slot}} = \left\{(x, y) \,\middle|\, |x| < \frac{b_1}{2},\ |y| < \frac{b_2}{2}\right\}. \tag{6.29}$$

For the slot trap, we can express the width in terms of base length as $b_2 = \epsilon b_1$, where ϵ is the aspect ratio, and the square trap can be considered as a particular case for $\epsilon = 1$, so that $b_1 = b_2 = b$. Square or slot traps are sometimes used in real field experiments [39].

We consider different scenarios where a single trap of either shape (6.27)–(6.29) is placed at the centre of an arena with a circular outer boundary of radius R. To fairly compare trap efficiencies in confined space, the trap dimensions should be related on a basis of equal perimeter lengths P (and not by trap area) [1, 8, 40]. This can be argued by the fact that trapping is fundamentally a phenomenon of interactions with the trap boundary. Following this, one gets

$$b_1 = \frac{\pi a}{1+\epsilon}, \quad b_2 = \epsilon b_1 \tag{6.30}$$

which reads $b = \frac{\pi a}{2}$ on relating the circular and square traps.

Figure 6.7 illustrates the trap shapes of equal perimeter lengths: (a) circle trap, (b)–(d) shows the transition from square to slot shape, which is characterized by the aspect ratio ϵ. As ϵ increases from 1, the base length decreases and width increases, forming a thinner rectangular slot with smaller area. Recall from Eq. (6.25) that if the mean step length is less than the trap diameter, then sub step lengths need not

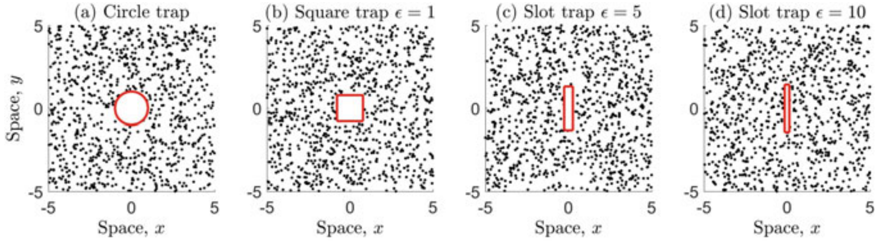


Fig. 6.7 Illustration of the trap shapes. **a** Circular trap with radius $a = 1$, **b** Square trap with side lengths $b = \frac{\pi}{2}$ ($\epsilon = 1$), **c** Slot trap with base length $b_1 = \frac{\pi}{6}$ and width $b_2 = \frac{5\pi}{6}$ ($\epsilon = 5$). **c** Thinner slot trap with $b_1 = \frac{\pi}{11}$ and $b_2 = \frac{10\pi}{11}$ ($\epsilon = 10$). All traps have the same perimeter length $P = \pi a$

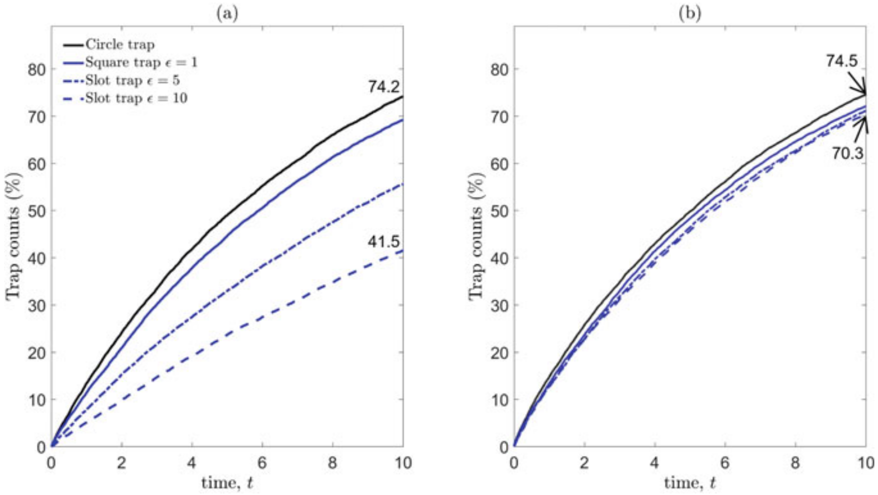


Fig. 6.8 Trap counts computed for different trap shapes whilst allowing for **a** individuals to 'jump' over the trap or **b** using sub step lengths with increment $\Delta l = 0.1$. Trap shapes are either circle with radius $a = 1$, square ($\epsilon = 1$) or rectangular slot ($\epsilon = 5, 10$) with equal perimeter lengths $P = 2\pi$. Dimensions of the square/slot traps can be determined from the equations in (6.30). The outer boundary is of circular shape with radius $R = 20$. Note that, the impact of different arena sizes (or equivalently average initial release distances) due to small changes in trap areas is negligible. The movement process is a SRW with Gaussian increments with $\sigma = 1.5$, so that $\mathbb{E}(l) < 2a$, and therefore, trap counts are similar for the circular trap in both plots, where it does not matter if individuals are allowed to jump over the trap or not. The initial population was uniformly distributed over the arena, and the density was fixed to ensure sufficient reduction of noise

be considered. However, for thinner slots, it is more likely for individuals to 'jump' over the trap, and therefore, one must take additional care and use sub steps.

Figure 6.8 compares trap counts across different scenarios with circle, square or slot shaped traps of equal perimeter lengths. In plot (a) we find that there is a large difference between trapping efficiencies, but this is a direct result of individuals 'jumping' over the trap, where trap counts were grossly underestimated for thinner slot traps. This methodology was used in [20]. In plot (b) we have the 'actual' corrected trap counts by using sub step lengths. In either case, we find that there is a hierarchy of trap shapes in terms of trapping efficiency, with the circle trap as the optimal shape, followed by the square and slot traps—with decreasing efficiency for thinner slots (i.e. as the aspect ratio ϵ increases). In plot (b), the trap count differences may seem minute, however, in a multiple trapping scenario, the differences in the number of captures can be exacerbated based on the choice of trap shape.

6.4 Anisotropic Models of Insect Movement

6.4.1 Correlated Random Walk (CRW)

In a more realistic scenario, an insect is more likely to keep moving in the same direction in the short term rather than to perform abrupt turns—allowing for correlation between the orientations of successive steps. As a result, there is a short-term localized directional bias in the movement path (forward persistence), and the corresponding movement process is anisotropic and known as the correlated random walk (CRW)' [41–44]. In this case, the turning angle distribution is zero-centered and centrally symmetric with null mean sine if we consider a balanced CRW, and peaked about the mean value. An example of such is the von Mises distribution

$$\psi(\alpha;\kappa) = \frac{e^{\kappa\cos\alpha}}{2\pi I_0(\kappa)}, \quad I_0(\kappa) = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{\kappa\cos\alpha} d\alpha \tag{6.31}$$

which ranges from $-\pi$ to π, and the concentration parameter $\kappa \in [0, \infty)$ measures the strength of the short-term directional persistence [45]. Here, $I_0(\kappa)$ denotes the zeroth order modified Bessel function of the first kind, defined through the integral indicated in Eq. (6.31).

For the von Mises distribution, the mean cosine c_α is computed as

$$c_\alpha = \frac{I_1(\kappa)}{I_0(\kappa)}. \tag{6.32}$$

With increasing c_α (or equivalently κ) there is an increase in the short-term directional persistence. However, in the case of a null mean cosine, the von Mises distribution reduces to the uniform distribution, in which case the SRW can be considered as a special case of the CRW. Note that other types of peaked circular distributions which are commonly used include the wrapped or truncated normal or the wrapped Cauchy distribution [46, 47].

6.4.2 MSD Formula for the CRW

The Mean Squared Displacement (MSD) of a 2D balanced CRW, is given by

$$\mathbb{E}\left[R_n^2\right] = n\mathbb{E}\left[l^2\right] + 2\mathbb{E}[l]^2 \frac{c_\alpha}{1-c_\alpha}\left(n - \frac{1-c_\alpha^n}{1-c_\alpha}\right), \tag{6.33}$$

which is expressed in terms of moments of step length l, mean cosine of the turning angle c_α, and step number n [27, 42]. As an aside note, the equation for the MSD of a non-balanced CRW is more complicated, and can be found in [43]. For a SRW, we have that $c_\alpha = 0$, and therefore, the MSD reduces to that previously seen in Eq. (6.21).

In case of a CRW, for a large number of steps n, the MSD approaches

$$\mathbb{E}\left[R_n^2\right]_{\mathsf{a}} = n\left(\mathbb{E}\left[l^2\right] + 2\mathbb{E}[l]^2\frac{c_\alpha}{1-c_\alpha}\right), \tag{6.34}$$

where the subscript 'a' is included here to represent the asymptotic value to which the MSD tends when n increases indefinitely.

It is readily seen from Eq. (6.34) that the actual MSD is asymptotically proportional to n, and therefore, the walk becomes isotropically diffusive in the long term. To demonstrate how the MSD behaves for a small number of steps, consider the case with a high directional persistence, so that $c_\alpha = 1 - \delta$ where $\delta \ll 1$. It follows that:

$$1 - c_\alpha^n = 1 - (1-\delta)^n = n\delta - \frac{1}{2}n(n-1)\delta^2 + O(\delta^3), \tag{6.35}$$

and on omitting terms of order δ^3 and higher, Eq. (6.33) becomes

$$\mathbb{E}\left[R_n^2\right]_{\mathsf{b}} = n\left(\mathbb{E}\left[l^2\right] + \mathbb{E}[l]^2(1-\delta)(n-1)\right), \tag{6.36}$$

where subscript 'b' is used for the MSD expression in case of small n. Therefore, in the general case $c_\alpha \neq 0$, the actual MSD in Eq. (6.33) describes the movement that in the course of time (measured here as the number of steps along the path) slows down from almost ballistic movement (i.e. the dominant term in $\mathbb{E}\left[R_n^2\right]_{\mathsf{b}}$ is n^2) to diffusion motion (i.e. $\mathbb{E}\left[R_n^2\right]_{\mathsf{a}}$ grows linearly with n). This is also valid, more generally, for any $c_\alpha > 0$, but is more prominent when c_α is close to 1.

6.4.3 Measuring Tortuosity

The amount of turning in an insect's movement path (tortuosity), can be quantified by the sinuosity index [24, 25, 48], defined as

$$S = \sqrt{\frac{v}{D}}, \tag{6.37}$$

where $v = \frac{\mathbb{E}[l]}{\Delta t}$ is the mean speed and D is the diffusion coefficient, which can also be written as

$$S = \sqrt{\frac{\mathbb{E}[l]}{D\Delta t}}. \tag{6.38}$$

For an isotropically diffusive 2D RW, the MSD is related to D through Eq. (6.22). It follows that an equivalent expression for the sinuosity index, written explicitly in terms of the asymptotic MSD in Eq. (6.34) and mean path length $L = n\mathbb{E}[l]$ is

$$S = 2\sqrt{\frac{L}{\mathbb{E}\left[R_n^2\right]_{\mathsf{a}}}}. \tag{6.39}$$

Note that, although the sinuosity index S is frequently used, there are various other methods of quantifying the tortuosity of discrete movement paths, see [49].

6.4.4 Biased Random Walk (BRW)

The Biased Random Walk (BRW) differs from the SRW and CRW by featuring a preference towards a certain direction at each time step and, unlike the CRW, retains no knowledge of the previous movement direction. Therefore, the BRW can be considered as a SRW with a latent preference to move towards a target and is often referred to as a random walk with drift. This bias in movement can be towards a specific point in space, in which case the orientation of the preferred direction will depend upon the spatial location of the individual at each step [27, 50, 51], or it can be towards a specific direction (often referred to as a 'point at infinity'), for example, magnetic North [24, 27, 52].

In the case of the bias being towards a constant global direction, the distribution of turning angles $\psi(\alpha)$ will remain zero-centred and centrally symmetric with mean cosine c'_α and null mean sine $s'_\alpha = 0$ (we include the prime to distinguish between this and the mean cosine/sine used for the CRW). However, the distribution of the global directions $\omega(\theta)$ which measures the angle of each location clockwise from the positive x-axis (equivalent to the angle θ used in polar form in Sect. 6.2.2), will not necessarily be zero-centred. Instead, $\omega(\theta)$ will be centred around the angle of the global direction bias measured from the positive x-axis. Mathematically this results in the mean sine s_θ, not necessarily being equal to 0. For example, the von Mises distribution would now have the form

$$\omega(\theta; \kappa', \mu) = \frac{e^{\kappa' \cos(\theta-\mu)}}{2\pi I_0(\kappa')}, \tag{6.40}$$

where κ' is the concentration parameter and μ is the angle towards the global direction of bias taken clockwise relative to the positive x-axis, c.f. Eq. (6.31).

The strength of the bias is now dependent on both the mean cosine c_θ and mean sine s_θ values of ω, or the mean cosine c'_α of the turning angle, and are related to κ' by the relation

$$c'_\alpha = c_\theta^2 + s_\theta^2 = \left(\frac{I_1(\kappa')}{I_0(\kappa')}\right)^2. \tag{6.41}$$

Increasing values of $c_\theta^2 + s_\theta^2$ (or equivalently κ') gives an increase in the strength of the bias, with $c_\theta^2 + s_\theta^2 \to 1$ resulting in movement that is close to 'straight-line' or ballistic. Whereas, $c_\theta^2 + s_\theta^2 = 0$ (or $\kappa' = 0$) reduces the von Mises distribution to the uniform distribution, and therefore, the BRW is equivalent to a SRW.

An important difference between the BRW and the CRW (including SRW as a special case) is that the expected location of an individual $\mathbb{E}[\mathbf{x}]$, is now not equal to the starting location of the individual. In the case for the bias being in a global direction, the expected location increases linearly with the number of steps n, towards the direction of the bias, and is given by

$$\mathbb{E}[\mathbf{x}] = (x_0, y_0) + n\mathbb{E}[l](c_\theta, s_\theta), \tag{6.42}$$

where (x_0, y_0) is the starting location and $\mathbb{E}[l](c_\theta, s_\theta)$ is the drift [53].

6.4.5 MSD Formula for the BRW

In the case of a BRW, where the bias is towards a constant global direction, the MSD is given by

$$\mathbb{E}[R_n^2] = n\mathbb{E}[l^2] + n(n-1)\mathbb{E}[l]^2 c'_\alpha = \underbrace{n\left(\mathbb{E}[l^2] - \mathbb{E}[l]^2 c'_\alpha\right)}_{\text{diffusion term}} + \underbrace{n^2\mathbb{E}[l]^2 c'_\alpha}_{\text{drift term}} \tag{6.43}$$

where c'_α is the mean cosine of the turning angles for the BRW. Equation (6.43) reflects the fact that the BRW is essentially a combination of the diffusive random walk and a drift, so that in the large time the corresponding MSD is dominated by the contribution from the drift [24]. In the case where there is no long-term bias, the mean cosine c'_α is null, and hence the MSD reduces to the MSD for the SRW as previously see in Eq. (6.21).

6.4.6 Equivalent RWs in Terms of Diffusion

In this section, we derive the conditions under which two balanced CRWs are 'equivalent', in the sense that they have the same MSD after n steps, given that n is sufficiently large. The MSD from Eq. (6.34) can be written as

$$\mathbb{E}\left[R_n^2\right] = L\mathbb{E}[l]\left(\frac{1+c_\alpha}{1-c_\alpha} + \gamma^2\right), \tag{6.44}$$

where $L = n\mathbb{E}[l]$ is the mean path length and $\gamma = \sqrt{\frac{\mathbb{E}[l^2]}{\mathbb{E}[l]^2} - 1}$ is the coefficient of variation. If we consider a second balanced RW with step length l^* and mean cosine

c_α^*, assuming the same coefficient of variation and mean path length L, we obtain the following 'condition of equivalence':

$$\frac{\mathbb{E}[l^*]}{\mathbb{E}[l]} = \left(\frac{1 - c_\alpha^*}{1 + \left(\frac{1-\gamma^2}{1+\gamma^2} \right) c_\alpha^*} \right) \left(\frac{1 + \left(\frac{1-\gamma^2}{1+\gamma^2} \right) c_\alpha}{1 - c_\alpha} \right), \tag{6.45}$$

so that both RWs have, asymptotically, the exact same MSD. If we consider the above relation, where one of the RWs is a SRW, say $c_\alpha^* = 0$, then the above reduces to

$$\frac{\mathbb{E}[l^*]}{\mathbb{E}[l]} = \frac{1 + \left(\frac{1-\gamma^2}{1+\gamma^2} \right) c_\alpha}{1 - c_\alpha}. \tag{6.46}$$

Now consider a CRW and a BRW with step lengths l, l' and mean cosines c_α, c'_α, respectively. To obtain a condition of equivalence between these RWs we assume the same path length L, same coefficient of variation γ and the same diffusivity, i.e. the same MSD part which arises from diffusion. In a CRW, the MSD is asymptotically due only to diffusion and is given by Eq. (6.44). In a BRW, the MSD given by Eq. (6.43), written as the sum of diffusion and drift terms, can be written as

$$\mathbb{E}\left[R_n^2\right] = \underbrace{L\mathbb{E}\left[l'\right](1 - c'_\alpha + \gamma^2)}_{\text{diffusion term}} + \underbrace{L^2 c'_\alpha}_{\text{drift term}}. \tag{6.47}$$

By Eq. (6.44) with the diffusive term in Eq. (6.47), and on re-arranging, one gets

$$\frac{\mathbb{E}[l]}{\mathbb{E}[l']} = \left(1 - \frac{c'_\alpha}{1 + \gamma^2} \right) \left(\frac{1 - c_\alpha}{1 + \left(\frac{1-\gamma^2}{1+\gamma^2} \right) c_\alpha} \right), \tag{6.48}$$

If we consider a SRW as a special case of the CRW with step length l^* and mean cosine $c_\alpha^* = 0$, then the above relation reduces to

$$\frac{\mathbb{E}[l^*]}{\mathbb{E}[l']} = 1 - \frac{c'_\alpha}{1 + \gamma^2} \tag{6.49}$$

where c'_α is given in Eq. (6.41) for a von Mises distribution of turning angles.

6.4.7 Drift Diffusion Equation

In Sect. 6.2.5, we mentioned that for a SRW the governing mean-field equation for the spatio-temporal population density $u = u(\mathbf{x}, t)$ is the standard diffusion equation. In the case of a BRW, which has a constant global preferred direction, the corresponding equation is the drift (or advection) diffusion equation [27], written as

$$\frac{\partial u}{\partial t} = -\mathbf{v} \cdot \nabla u + D\nabla^2 u, \tag{6.50}$$

where $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)$ is the gradient operator, $\mathbf{v} = (v_1, v_2)$ is the (constant) average drift velocity and D is the constant diffusion coefficient. The first term on the RHS in the above equation relates to the drift involved in the BRW whilst the second term describes the diffusion. Note that, in the case where there is no drift $\mathbf{v} = 0$, the drift diffusion equation returns to the standard diffusion equation previously seen in Eq. (6.1).

The drift diffusion equation provides a useful baseline case for modelling a population that responds to an attractant (or lure) in a unified global direction. However, for more complicated trapping scenarios, for example, with a baited trap in an annular arena, the bias is spatially dependent, and the governing PDE is much more complex (i.e. the Fokker Planck equation [27]). Analytical solutions are only available in very specific cases [27, 54], and therefore, for the most part must be solved numerically.

6.4.8 Biased and Correlated Random Walk (BCRW)

The BRW and CRW models can be extended if we consider a RW which allows for both short and long-term biases. These biased and correlated random walks (BCRW) give a more flexible approach to modelling insect movement, as they allow individuals to move with both some knowledge of their previous direction, as well as with a preference towards a specific direction or target [51, 55, 56]. BCRW can be expressed in various ways depending on how the biased and correlated components are to be combined. For example, they can be considered as a weighted vector sum where the change in location is governed by the equations [56, 57]

$$\begin{aligned}(\Delta x)_i &= l_i \left[w \cos\left(\Omega_{i-1}\right) + (1-w)\cos\theta_{i-1}\right],\\ (\Delta y)_i &= l_i \left[w \sin\left(\Omega_{i-1}\right) + (1-w)\sin\theta_{i-1}\right],\end{aligned} \tag{6.51}$$

where l_i is the step length, w is the weighting factor $w \in [0, 1]$, Ω_{i-1} is the local direction of the target and θ_{i-1} is the previous direction of movement, and all angles are measured with respect to the positive x-axis and modelled by circular distributions as in Sect. 6.4.4.

A similar model for a BCRW is formed by weighting the angles of movement [51, 58]

$$\theta_i = w\Omega_{i-1} + (1-w)\theta_{i-1}. \tag{6.52}$$

Note that for both these cases as $w \to 0$ the BCRW more closely resembles a pure CRW, whereas as $w \to 1$ the movement becomes more similar to a BRW.

Whilst these two models look similar, there is a subtle but important difference that can lead to qualitatively and quantitatively different movement behaviour. The vector weighted method in Eq. (6.51) uses the weighting parameter to decide how far in the direction of each component the individual moves, whereas, Eq. (6.52) gives a weighted average of the two directional angles dependent on the weighting. This difference can be highlighted by considering the extreme case where $\Omega_{i-1} = 0$, $\theta_{i-1} = \pi$ and $w = 0.5$. In the vector weighted approach, the individual would remain at its current location, whereas in the angular approach the individual moves in a direction with angle $\theta_i = \frac{\pi}{2}$. Both these models require the individual to constantly balance their forward persistence and navigation at each time step.

6.5 Effect of Movement on Trap Counts

6.5.1 *Effect of Movement Diffusion*

In Sect. 6.3 we focused on the impact of geometry on trap counts, and relied solely on a SRW model for individual movement. To understand how short-term directional persistence may effect trap counts, we consider two scenarios, where each individual in the population performs a CRW with either (a) increasing diffusion, or (b) with the same diffusion. For the latter, we have already presented the conditions of 'equivalence' in Sect. 6.4.6 to directly relate the CRW to a SRW, see Eq. (6.46).

Consider a CRW and a SRW with Gaussian increments, so that step lengths l, l^* are distributed according to the Rayleigh distribution given in (6.18) with scale parameters σ, σ^*, respectively. The turning angle distribution is given by the von Mises distribution in (6.31) for the CRW with mean cosine c_α, and the uniform distribution for the SRW with null mean cosine. Assuming that both these walks have the same mean path length and fixed step length coefficient of variation $\gamma = \sqrt{\frac{4}{\pi} - 1}$, and provided that the step number n is sufficiently large, we can relate scale parameters from Eq. (6.46), which reads

$$\frac{\sigma^*}{\sigma} = \frac{1 + \left(\frac{\pi}{2} - 1\right) c_\alpha}{1 - c_\alpha}. \tag{6.53}$$

Also, note that a CRW behaves as a SRW in the long term, and therefore, the sinuosity is the same with value

$$S = \left(\frac{2\pi}{\sigma^{*2}}\right)^{\frac{1}{4}}. \tag{6.54}$$

Figure 6.9 demonstrates that if all individuals traverse the same maximum path length, the trap counts tend to remain the same with an increase in diffusion, but can decrease significantly if there is a strong short-term directional persistence. This can be explained by the fact that a strong short-term bias leads to a movement path

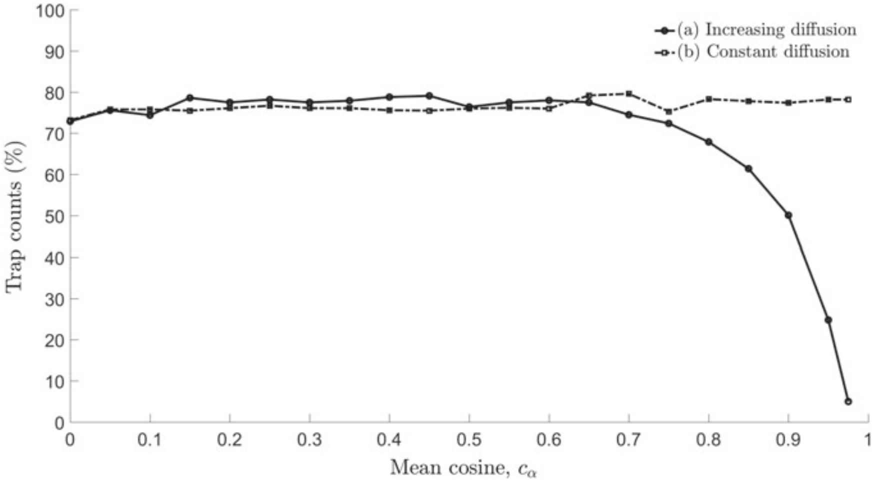


Fig. 6.9 Trap counts plotted as a function of mean cosines. Each individual in the population performs a CRW, with the SRW as a special case for $c_\alpha^* = 0$. In **a**, the diffusion increases with the mean cosine c_α, and the movement scale parameters are the same for all walks $\sigma^* = \sigma = 1.5$. Sinuosity values gradually decrease from $S = 1.29$ for $c_\alpha^* = 0$ ($\kappa^* = 0$) to $S = 0.16$ for $c_\alpha = 0.975$ ($\kappa = 20.26$). In this case, the step number is fixed as $n = 1000$ (corresponding to time $t = 10$ with time increment $\Delta t = 0.01$), and the same maximum path length $L = 1880$ is traversed. In **b**, the movement scale parameter σ decreases from $\sigma^* = 1.5$ with an increase in c_α, and the step number n is increased (corresponding to different times), to ensure that each RW has the exact same sinuosity value $S = 1.29$, see Eqs. (6.53) and (6.54), and the same maximum path length $L = 1880$. As a result, the asymptotic MSD remains the same. In terms of geometry, we considered a circular trap of unit radius with circular outer boundary of radius $R = 20$, with an initial homogeneous population distribution on the arena. To ensure the population is confined at all times, if an individual attempts to overstep the boundary, then an alternative step with a completely random direction is chosen at the previous location. Note that, for the SRW ($c_\alpha^* = 0$), the trap counts correspond with those seen in Fig. 6.8 for the circular trap

which is close to straight line movement, therefore, those individuals moving away from the trap and close to the outer boundary will tend to accumulate at the boundary edge. However, if diffusion is kept constant, then the trap counts remain the same, irrespective of the strength of the short-term directional bias.

6.5.2 Baited Trapping

Traps can be baited with the use of chemical or visual lures (for example, sex or aggregation pheromones, or light), to attract insects towards the trap location, and are commonly used to monitor insect pest populations [2, 59]. Clearly, attractant baited traps are more efficient in obtaining trap counts in comparison to passive pitfall traps. This can be particularly useful in capturing pest insect species at low

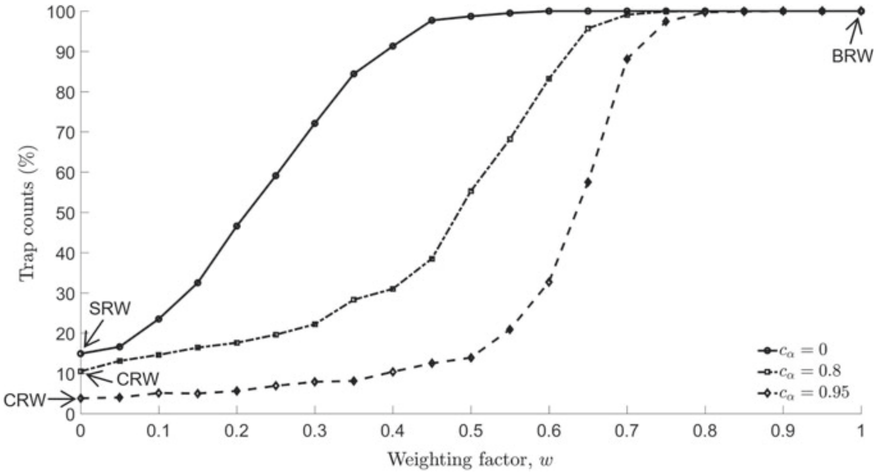


Fig. 6.10 Trap counts plotted as a function of the weighting factor w. Each individual in the population performs a BCRW with the centre of the circular trap as the global target direction. We considered Gaussian step increments with movement scale parameter $\sigma = 1.5$. The strength of the long-term bias was fixed, and measured through the mean cosine with value $c'_\alpha = 0.25$ (corresponding to concentration parameter $\kappa' = 1.16$), and c_α corresponds to the strength of the short-term directional persistence. Each individual in the population executes a maximum number of $n = 100$ steps, corresponding to time $t = 1$ with time increment $\Delta t = 0.01$. All other details are the same as in the caption of Fig. 6.9

densities [60], or to detect foreign or 'exotic' pests as soon as they enter an area [61]. Once a trap is baited, and the insect population is receptive to the lure, the movement is globally biased towards the trap location and can be modelled using a BCRW, see Sect. 6.4.8. Under such a description, the insects may balance short and long-term directional persistence mechanisms as they navigate through space.

Figure 6.10 shows that in the absence of long term bias ($w = 0$), trap counts decrease with increasing short-term directional persistence (or increasing diffusion), as previously seen in Sect. 6.5.1. However, with an increase in w, the long-term bias becomes more dominant, as more weight is given to the BRW component, and as a result each individual navigates in the long term towards the trap, resembling the type of movement behaviour one may expect if the trap is baited.

6.6 Concluding Remarks

Ground dwelling insects are almost exclusively sampled through the use of pitfall traps [3]. This is mainly due to the simplicity of the methods, where these sorts of traps are relatively easy to construct and deploy and, trap counts can be manipulated to obtain important information on patterns on insect distribution and abundance [1, 62]. Interpreting trap count data is challenging, partly due to the fact trapping methodologies can vary considerably, in relation to survey design and how trap counts are reported [63, 64]. Although there have been some recent attempts to address this (see [11]), more broadly, it can be difficult to draw meaningful comparisons across studies. In this context, mathematical models provide a useful and cost-effective alternative to real field experiments [13, 19, 65]. In particular, mean-field approaches and mechanistic models of individual movement can be used to study the details of trap counts dynamics, often providing complementary information or even information that may be difficult to obtain from field sampling [14]. In this chapter, we have provided an overview of the key mathematical approaches that have been developed for 2D insect trap counts modelling and have focused on important concepts related to trapping. In particular, we have highlighted the variation in trap counts due to trap geometries, differences between passive and baited traps and the effects of movement. This information has important implications for spatial ecology.

Spatial ecology necessitates a full understanding of movement. Understanding how habitats are connected by dispersal and movement is critical to determining the distribution, abundance and persistence of insect species. For instance, understanding the drivers of asynchrony in population fluctuations in metapopulation dynamics (e.g. [66]) now necessitates great attention to mechanistic detail. The mean squared displacement is an important metric for evaluating movement patterns and particularly for inferring random walks from data [43] (see Sect. 6.4.2). Utilizing these approaches allows the spatial distribution, and hence the likely metapopulation structures of a species to be determined (e.g. [67]). The approaches we review in this chapter highlight the importance of trap counts modelling to spatial dynamics (albeit in 2D). Extending these mechanistic explanations for movement further is critical for advancing spatial ecology and building the work we present here to 3D (i.e. for flying insects) is critical for a richer appreciation of mechanistic movement ecologies to metapopulation and spatial ecological dynamics [68, 69].

References

1. Work TT, Buddle CM, Korinus LM, Spence JR (2002) Pitfall trap size and capture of three taxa of litterdwelling arthropods: implications for biodiversity studies. Environ Entomol 31:438–448
2. Pimentel D (2009) Integrated pest management: innovation-development process, vol 1. Springer, Berlin, pp 83–88
3. Hohbein RR, Conway CJ (2018) Pitfall traps: a review of methods for estimating arthropod abundance. Wildl Soc Bull 42(4):597–606
4. Den Boer PJ (1981) On the survival of populations in a heterogeneous and variable environment. Oecologia 50:39–53
5. Niemelä J, Halme E, Haila Y (1990) Balancing sampling effort in pitfall trapping of carabid beetles. Entomol Fenn 1:233–238
6. Dye C (1983) Insect movement and fluctuations in insect population size. Antenna 7:174–178
7. Raworth DA, Choi WJ (2001) Determining numbers of active carabid beetles per unit area from pitfall-trap data. Ent. Exp Appl 98:95–108

8. Luff ML (1975) Some features influencing the efficiency of pitfall traps. Oecologia 19(4):345–357
9. Pekár S (2002) Differential effects of formaldehyde concentration and detergent on the catching efficiency of surface active arthropods by pitfall traps. Pedobiologia 46(6):539–547
10. Koivula MJ, Kotze DJ, Hiisivuori LKH, Rita HJ (2003) Pitfall trap efficiency: do trap size, collecting fluid and vegetation structure matter? Entomol Fenn 14:1–14
11. Brown GR, Matthews IM (2016) A review of extensive variation in the design of pitfall traps and a proposal for a standard pitfall trap design for monitoring ground-active arthropod biodiversity. Ecol Evol 6(12):3953–64
12. Cheli GH, Corley JC (2010) Efficient sampling of ground-dwelling arthropods using pitfall traps in arid steppes. Neotrop Entomol 39(6):912–917
13. Petrovskii S, Bearup D, Ahmed DA, Blackshaw R (2012) Estimating insect population density from trap counts. Ecol Complex 10:69–82
14. Petrovskii S, Petrovskya N, Bearup D (2014) Multiscale approach to pest insect monitoring: random walks, pattern formation, synchronization and networks. Phys Life Rev 11(3):467–525
15. Greenslade PJM (1964) Pitfall trapping as a method for studying populations of carabidae (Coleoptera). J Anim Ecol 33(2):301–310
16. Kareiva PM (1983) Local movement in herbivorous insects: applying a passive diffusion model to mark-recapture field experiments. Oecologia (Berlin) 56:234. https://doi.org/10.1007/BF00379695
17. Okubo A (1980) Diffusion and ecological problems: mathematical models. Springer, Berlin
18. Turchin P (1998) Quantitative analysis of movement. Measuring and modelling population redistribution in animals and plants. Sinauer Associates, Inc. Sunderland, Massachusetts
19. Bearup D, Benefer CM, Petrovskii S, Blackshaw RP (2016) Revisiting Brownian motion as a description of animal movement: a comparison to experimental movement data. Methods Ecol Evol 7(12):1525–37
20. Ahmed DA, Petrovskii S (2019) Analysing the impact of trap shape and movement behaviour of ground-dwelling arthropods on trap efficiency. Methods Ecol Evol 10(8):1246–64
21. Carslaw HS, Jaeger JC (1959) Conduction of heat in solids, 1st edn. Clarendon Press, Oxford
22. Crank J (1975) The mathematics of diffusion, 2nd edn. Oxford University Press, Oxford
23. Berg HC (1983) Random walks in biology. Princeton University Press, Princeton
24. Benhamou S (2006) Detecting an orientation component in animal paths when the preferred direction is individual-dependent. Ecology 87(2):518–528
25. Benhamou S (2004) How to reliably estimate the tortuosity of an animal's path: straightness, sinuosity, or fractal dimension? J Theor Biol 229(2):209–220
26. Lin CC, Segel LA (1974) Mathematics applied to deterministic problems in the natural sciences. Macmillan, New York, NY
27. Codling E, Plank MJ, Benhamou S (2008) Random walk models in biology. J R Soc Interface 5(25):813–834
28. Petrovskii S, Morozov A, Li BL (2008) On a possible origin of the fat-tailed dispersal in population dynamics. Ecol Complex 5:146–150
29. Holmes EE (1993) Are diffusion models too simple? a comparison with telegraph models of invasion. Am Nat 142(5):779–795
30. Hughes DP, Andersen SB, Hywel-Jones NL, Himaman W, Billen J, Boomsma JJ (2011) Behavioral mechanisms and morphological symptoms of zombie ants dying from fungal infection. BMC Ecol 11(13)
31. James A, Plank MJ, Edwards AM (2011) Assessing Lévy walks as models of animal foraging. J R Soc Interface 8(62):1233–1247. https://doi.org/10.1098/rsif.2011.0200
32. Ahmed DA, Petrovskii S, Tilles PFC (2018) The Lévy or diffusion controversy: how important is the movement pattern in the context of trapping? Mathematics 6(77)
33. Grimm V, Railsback SF (2005) Individual-based modeling and ecology. Princeton University Press, Princeton
34. Bearup D, Petrovskii S (2015) On time scale invariance of random walks in confined space. J Theor Biol 367:230–245

35. Okubo A, Levin SA (2001) Diffusion and ecological problems: modern perspectives. Springer, Berlin, Germany
36. Sornette D (2004) Critical phenomena in natural sciences, 2nd edn. Springer, Berlin
37. Othmer HG, Dunbar SR, Alt W (1988) Models of dispersal in biological systems. J Math Biol 26:263–198
38. Hillen T, Othmer HG (2000) The diffusion limit of transport equations derived from velocity jump processes. SIAM J Appl Math 61:751–775
39. Blackshaw RP, Vernon RS, Thiebaud F (2018) Large scale Agriotes spp. click beetle (Coleoptera: Elateridae) invasion of crop land from field margin reservoirs. Agric For Entomol 20(1):51–61
40. Miller JR, Adams CG, Weston PA, Schenker JH (2015) Trapping of small organisms moving randomly. Principles and applications to pest monitoring and management. Springer briefs in ecology. Springer, United States
41. Patlak CS (1953) Random walk with persistence and external bias. Bull Math Biophys 15:311–338
42. Hall RL (1977) Amoeboid movements as a correlated walk. J Math Biol 4:327–335
43. Kareiva PM, Shigesada N (1983) Analyzing insect movement as a correlated random walk. Oecologia 56(2–3):234–238
44. Bovet P, Benhamou S (1988) Spatial analysis of animals' movements using a correlated random walk model. J Theor Biol 131(4):419–433
45. Morales JM, Haydon DT, Frair J, Holsinger KE, Fryxell JM (2004) Extracting more out of relocation data: building movement models as mixtures of random walks. Ecology 85(9):2436–45
46. Mardia KV, Jupp PE (2000) Directional statistics. Wiley, Chichester
47. Codling EA, Bearon RN, Thorn GJ (2010) Diffusion about the mean drift location in a biased random walk. Ecology 91(10):3106–13
48. Benhamou S (2018) Mean squared displacement and sinuosity of three-dimensional random search movements. arXiv:1801.02435
49. Almeida PJAL, Vieira MV, Kajin M, Forero-Medina G, Rui. Cerqueira. (2010) Indices of movement behaviour: conceptual background, effects of scale and location errors. Zoologia (Curitiba) 27(5):674–680
50. Hill NA, Hader DP (1997) A biased random walk model for the trajectories of swimming micro-organisms. J Theor Biol 186:503–526
51. Schultz CB, Crone EE (2001) Edge-mediated dispersal behaviour in a prairie butterfly. Ecology 82:1879–1892
52. Benhamou S (2003) Bicoordinate navigation based on non-orthogonal gradient fields. J Theor Biol 225:235–239
53. Bailey J (2019) Modelling and analysis of individual animal movement. PhD thesis, University of Essex
54. Codling EA (2003) Biased random walks in biology. PhD thesis, University of Leeds
55. Nicosia A, Duchesne T, Rivest LP, Fortin D (2017) A general hidden state random walk model for animal movement. Comput Stat Data Anal 105:76–95
56. Bailey J, Wallis J, Codling EA (2018) Navigational efficiency in a biased and correlated random walk model of individual animal movement. Ecology 99(1):217–223
57. Benhamou S, Bovet P (1992) Distinguishing between elementary orientation mechanisms by means of path analysis. Anim Behav 43:371–377
58. Fortin D, Morales JM, Boyce MS (2005) Elk winter foraging at fine scale in yellowstone national park. Oecologia 145:335–343
59. Jacobson M, Beroza M (1963) Chemical insect attractants 140(3574):1367–73
60. Mccullough DG, Siegert NW, Poland TM, Pierce SJ, Ahn SZ (2011) Effects of trap type, placement and ash distribution on emerald ash borer captures in a low density site. Environ Entomol 40(5):1239–52
61. Rassati D, Faccoli M, Toffolo EP, Battisti A, Marini L (2015) Improving the early detection of alien woodboring beetles in ports and surrounding forests. J Appl Ecol 52(1):50–58

62. Epsky ND, Morrill WL, Mankin R (2004) Traps for capturing insects. Springer, Dordrecht. In: Encyclopedia of entomology
63. Adis J (1979) Problems of interpreting arthropod sampling with pitfall traps. Zoologischer Anzeiger 202:177–184
64. Woodcock BA (2005) Insect sampling in forest ecosystems. Blackwell Science ltd
65. Bearup D, Petrovskaya N, Petrovskii S (2015) Some analytical and numerical approaches to understanding trap counts resulting from pest insect immigration. Math Biosci 263:143–160
66. Hanski I (1999) Metapopulation ecology. Oxford series in ecology and evolution
67. Turchin P (1991) Translating foraging movements in heterogeneous environments into the spatial distribution of foragers. Ecology 72(4):1253–66
68. Moorcroft PR, Lewis MA, Crabtree RL (2006) Mechanistic home range models capture spatial patterns and dynamics of coyote territories in yellowstone. Proc Biol Sci 273(1594):1651–59
69. Ahmed DA, Benhamou S, Bonsall MB, Petrovskii SV (2020) Three-dimensional random walk models of individual animal movement and their application to trap counts modelling. Available at BioRxiv. https://doi.org/10.1101/2020.07.28.224766

Part II
Medical Image Analysis

Chapter 7
Artificial Intelligence in Dermatology: A Case Study for Facial Skin Diseases

Rola El-Saleh, Hazem Zein, Samer Chantaf, and Amine Nait-ali

Abstract The purpose of the first part of this chapter is to cover broadly the concept of using Artificial Intelligence (AI) in the field of dermatology. Afterward, it will mainly focus on facial skin diseases by covering some common AI-based approaches. The aim of this research application is the ability to identify certain pathologies by analyzing face images with present lesions through both AI and computer vision techniques. In particular, a special interest will be addressed to Machine-learning and Deep-learning approaches. In a case study, some key functionalities of a prototype software, developed by our research biometric group, will be presented.

7.1 Introduction

Analyzing human skin health using artificial intelligence (AI) is neither a new idea nor a new technology. For decades, researchers have been developing automatic techniques to analyze human skin images in order to identify diseases, to quantify and to track their evolution over time. General-purpose image processing and, in particular, pattern recognition algorithms were used for this purpose. After the processing phase, extracted features are then passed through classifiers in order to assign each skin lesion to the most likely disease. Based on this analysis, this type of processing may somehow be considered as AI-based. However, it is controversial in the scientific community because some may argue that the processing models are not obviously inspired from human perception. Nowadays, technologies dedicated to skin analysis are getting much and more advanced. Consequently, AI-based methods make more sense due to the following reasons: (1) easy access to high-performance computers

R. El-Saleh · A. Nait-ali (✉)
Univ Paris Est Creteil, LISSI, F-94400 Vitry, France
e-mail: naitali@u-pec.fr

R. El-Saleh
e-mail: rolaelsaleh@gmail.com

H. Zein · S. Chantaf
Lebanese University-Faculty of Technology-Branch 1 - Saida, Beirut, Lebanon

T. D. Pham et al. (eds.), *Advances in Artificial Intelligence, Computation, and Data Science*, Computational Biology 31,
https://doi.org/10.1007/978-3-030-69951-2_7

(2) recent advances in machine-learning and deep-learning, (3) increase of medical image databases in terms of quantity and quality. Considering dermatology applications, it is obvious that we reached a stage where dermatologists are currently using AI as a clinical assistant system. This statement doesn't exclude the fact that, one day, AI might compete with humans in this field. Within this context, many research groups are collaborating with companies to develop interesting and promising applications such as the analysis of skin cancer. This is achieved by training the AI models over thousands or even hundred of thousands of skin images. Such systems are able to classify lesion images, either as benign or malignant (e.g., melanoma). Moreover, they can be used to predict or prevent the occurrence of some skin diseases. By pushing further the functionalities, AI-based systems can also consider multiple classes in order to distinguish between numerous skin lesions. Recently, smartphone applications have been designed for users so that they can remotely analyze their skin. Some of the functionalities aim to prevent the onset of skin diseases, and others provide useful skincare recommendations. Technically speaking, these applications require an interactivity between the user and the AI model. In some other cases, it can involve dermatologists when critical decisions are to be made. In this chapter, it specifically focuses on facial skin diseases using AI models. The case study in this chapter highlights an AI-based system (still under development) which is capable of classifying numerous facial skin diseases such as acne, Actinic Keratosis, and Blepharitis. This chapter is organized as follows: in Sect. 7.2, it focuses on the state of the art of facial skin diseases. In Sect. 7.3, a case study is presented and some specific AI approaches which are embedded in a software we have developed in Sect. 7.4 are discussed. Finally, a conclusion is provided in Sect. 7.5.

7.2 State of the Art

The use of automated methods to detect facial skin diseases has started in the past few years and was mostly focused on the detection of acne. Many techniques have been proposed and researches have been carried out to increase the accuracy of identification. Khongsuwan et al. used UV fluorescence images to detect and count acne. UV images are cropped, resized, converted to a gray image and enhanced and finally, H-maxima transform is used to count acne. Despite its efficiency, this technique may hurt the skin if exposed to UV light for a long duration [1]. Chang et al. detected automatically face skin defects and classified them into spots, acne, and normal skin using support-vector-machine-based classifier. The classification is preceded by an automatic selection of the region of interest and then extraction of the skin defects. They used images in front and profile views. The experimental results show the proposed approach is adequate and procures good accuracy [2]. Chantharaphaichit et al. proposed an automatic method to detect acne. Their approach consists first on marking the round-shaped areas in the face skin using blob detection method. Then they extract the specific features from the selected regions and finally send them to a Bayesian classifier to decide whether it is acne or not. The accuracy of this technique is affected by the place, form, and lighting conditions of acne in the image [3]. Kittigul worked on the detection and quantification of acne using Speeded Up Robust Features. She extracted 6 main characteristics to recognize acne. The classification is achieved by K-Nearest Neighbors algorithm [4]. Xiao lei Shen et al. detected facial acne vulgaris using CNN networks. Their method is based on two parts. Firstly, they detect skin to select the region of interest using a binary-classifier deciding skin or non-skin. Secondly, acne vulgaris is classified into one of seven classes including a healthy skin class using a seven-classifier [5]. Amini et al. developed a mobile application to detect and classify acne. The face image is captured by the camera of the mobile and then undergoes a series of image processing to extract the region of interest then identify and classify acne into two types: papules and pastules. The application could recognize and classify acne with high accuracy [6]. Binol et al proposed an aided diagnostic system to identify rosacea using deep learning. They used Inception and ResNet networks to extract features from facial images and then identify rosacea lesions [7]. Yang et al. developed a model to diagnose benign pigmented facial skin lesions using two CNN networks. They could identify six pigmentary facial skin pathologies. They extract lesions from images to train the networks. This model achieved good performance [8].

7.3 Study Case

We developed an automated system based on deep learning approach to help dermatologists diagnosing many facial skin diseases with high accuracy. Our system could recognize 8 face skin disorders, healthy skin, and no skin. For this purpose, we proposed a fine-tuned version of VGG-16 adapted to facial skin diseases recognition that we called Facial Skin Diseases Network (FSDNet). We created a dataset constituted of 20000 images gathered from online sources representing the ten classes classified by our network. The dataset is divided into training set to learn our network and fit the parameters and validation set to evaluate the efficiency and robustness of the system. The images are first labeled and then resized to fit our model. Five divide modes are used 90:10, 80:20, 70:30, 60:40, and 50:50 for training versus validation sets. Finally, to assess the effectiveness of the model, we give the system face skin images not included in the dataset to classify them.

7.3.1 Considered Skin Diseases

In our application, we considered eight most common facial skin diseases affecting all ages, genders, and races. These diseases are Acne, Actinic Keratosis, Angioedema, Blepharitis, Rosacea, Vitiligo, Melasma, and Eczema.

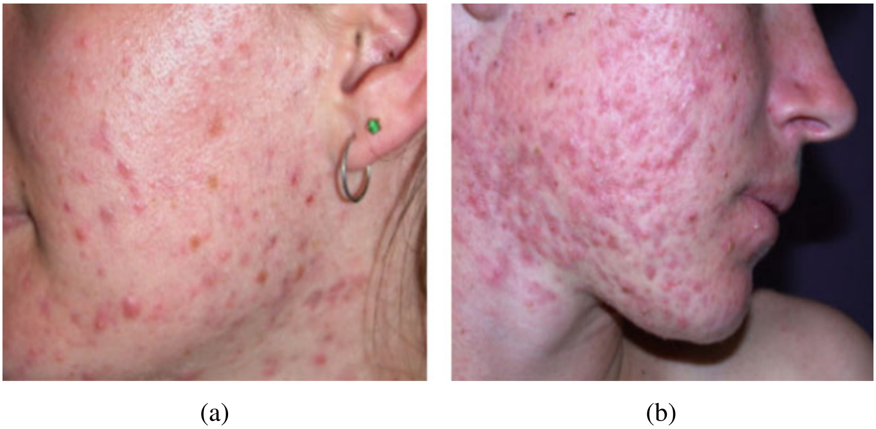

(a) (b)

Fig. 7.1 Acne disease [23]. **a** Acne in female, **b** Acne in male

7.3.1.1 Acne

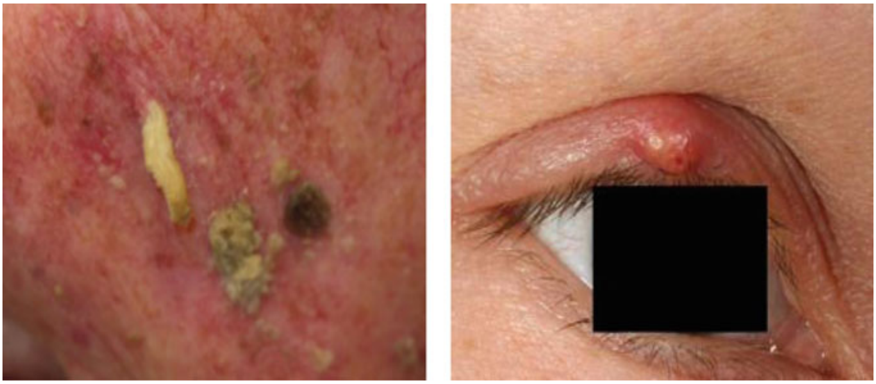
Acne is one of the most common facial skin diseases that affect especially teenagers. It is due to many factors such as hormonal changes, sebum production, bacteria, and inflammation [9]. It can be mild, moderate, or severe and is commonly located on the face (see Fig. 7.1), neck, shoulders, chest, and upper back; it is characterized by blackheads, whiteheads, pustules, papules, cysts, and nodules. This disease can scar or darken the skin if left untreated.

7.3.1.2 Actinic Keratosis

It is also known as Solar keratosis. Actinic keratosis or solar keratosis is a precancer characterized by scaly patches on the skin (see Fig. 7.2a). It is caused by recurrent exposure to ultraviolet radiation from the sun or tanning beds [10]. It can be found on the face, lips, ears, the backs of your hands, forearms, shoulders, scalp, or neck. It appears as small, scaly, thick spots on the skin of different colors.

7.3.1.3 Blepharitis

Blepharitis is an inflammation affecting the eyelids caused by bacteria, dandruff, rosacea, or even allergy to substances used in contact with lids [11]. This disorder is mostly characterized by redness, sticking, and crusting of the eyelids, itch, blurring sight (see Fig. 7.2b).

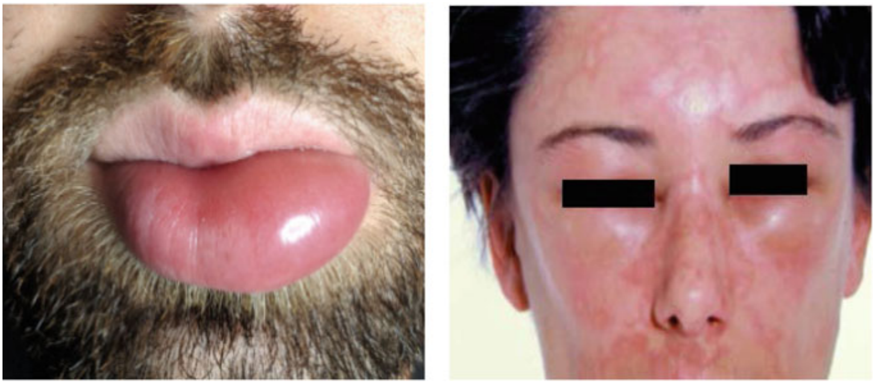


(a) (b)

Fig. 7.2 **a** Actinic Keratosis, **b** Blepharitis [23]

7.3.1.4 Angioedema

Angioedema is a swell of short life in the layer just under the skin. It occurs in many parts of the body but mostly around the eyes and lips (see Fig. 7.3). The swell can cause ache and warmth in the affected area [12]. Angioedema can result from an allergic reaction to food, animal, plant, or drug and also can be hereditary.



(a) (b)

Fig. 7.3 Angioedema disease [23]. **a** Angioedema around the lips, **b** Angioedema around the eyes

7.3.1.5 Vitiligo

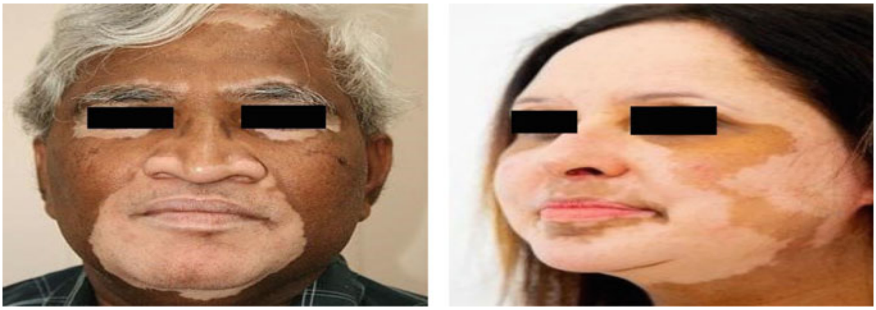
Vitiligo is a long-term skin disease characterized by white spots (depigmentation) on the skin, of variable size, appearance, and location, which tend to enlarge (see Fig. 7.4). It is more observable in dark-skinned persons [13]. Vitiligo is caused by the loss of melanin due to an autoimmune disorder or heredity. Persons suffering from vitiligo are more exposed to sunburns, loss of hearing, stress, and sight problems.

7.3.1.6 Rosacea

Rosacea is a skin disorder affecting mostly Caucasians persons of middle age in the face, mainly the nose, cheeks, forehead, and chin [14]. It occurs due to inflammatory, environmental, or hereditary factors. It has many symptoms such as red papules, red enlarged nose, red face bumps, swelling, eyelids redness, and dryness of facial skin (see Fig. 7.5a).

7.3.1.7 Melasma

Melasma is a skin pigmentation disease causing brown patches mainly on the face (see Fig. 7.5b). It affects women more than men and even persons with brown skin [15]. Sun exposure, hormone changes, pregnancy, and thyroid diseases may cause melasma. It doesn't always necessitate treatment and also may come back even if treated.



(a) (b)

Fig. 7.4 Vitiligo disease [23]. **a** Vitiligo in male, **b** Vitiligo in female

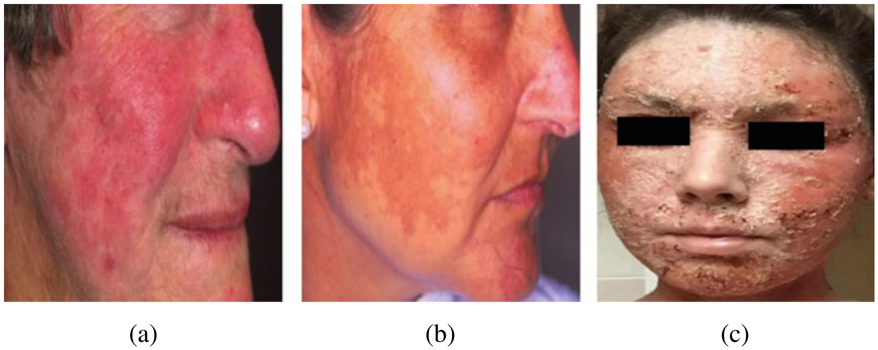


Fig. 7.5 **a** Rosacea, **b** Melasma, **c** Eczema [23]

7.3.1.8 Eczema

Eczema is a skin condition characterized by inflamed red and itchy skin. There are many types of eczema and it could affect all ages [16]. It is provoked foods, environmental conditions, soaps, detergents, sweating, and stress. Eczema is always accompanied by itch. Affected areas may be red, irritating, greasy, or oily (see Fig. 7.5c). Hair loss can occur in the area with the rash.

7.3.2 Machine-Learning/Deep-Learning Approaches

Machine learning is a branch of artificial intelligence. It is described as the brain that permits systems to learn automatically, analyze data, and thus provide an adequate decision. Machine learning approaches are grouped depending on their learning type: supervised, unsupervised, and semi-supervised learning. It exists many machine learning techniques: regression algorithms including linear regression, and logistic regression, Decision Tree algorithm, Support Vector Machine, and Artificial Neural Networks forming the basis of deep learning.

Deep Learning is a sub-area of machine learning defined as neural network architectures with multiple layers and parameters. It is useful for recognition, information processing, and classification purpose and its use has broadened in industries such as cars, aerospace, electronics, and also in defense and medical fields. To obtain high system performance, deep learning requires large data to train the model and significant computing power to speed up data processing. There are many types of deep learning models:

1. Autoencoders which are unsupervised networks mainly used for dimensionality minimization.
2. Convolutional Neural Networks (CNN) used mainly in image recognition purposes.

3. Recurrent Neural networks designed for sequence identification such as speech, text, or signal.
4. Recursive Neural Networks used in natural language processing.

In our application, we are concerned by CNN models. CNN are the most prevalent type of deep learning models due to their architecture [17]. They are very efficient for image classification for many reasons. They use 2D convolution layers which makes them suitable to process 2D data such as images. CNN extract features automatically from images and learn them during the training process. This automatic extraction makes the model very accurate especially for computer vision purposes such as object classification and image classification. Even more, CNN can reduce the number of parameters in the network without losing any information which is very important in image recognition tasks. CNN architecture includes an input layer, convolution layers responsible for extracting the features, pooling layers reducing the number of parameters to learn while retaining information, and fully connected layers flattening outputs coming from preceding layers and predicting the class of the image (see Fig. 7.6) [18]. It exists in many CNN architectures differing by their structure, number of layers, and their design. The most important ones are LeNet [19], AlexNet [20], Googlenet (also called Inception) [21], and VGGNet [22] that we used in our application mainly VGG-16 model. VGG-16 is a deep network consisting of 16 weight layers: 13 convolutional layers and 3 fully connected layers followed by a Softmax classifier. The convolutional layers are divided into five groups each followed by a Maxpooling layer. We proposed a fine-tuned version of the VGG-16 model adapted to face skin disorders classification. We modify the structure of the fully connected layers (see Fig. 7.7).

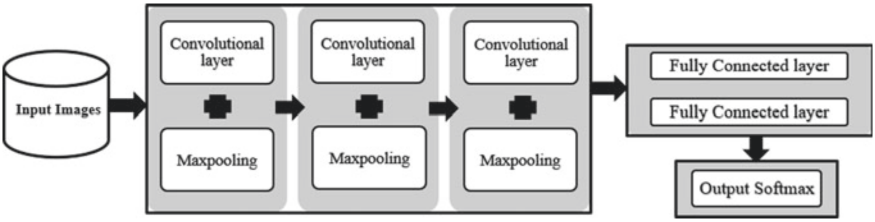


Fig. 7.6 General architecture of CNN

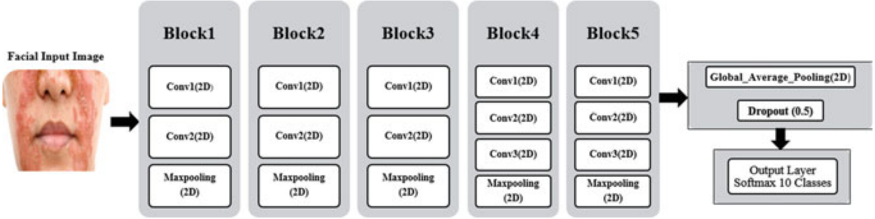


Fig. 7.7 FSDNet architecture

7.3.3 Preliminary Results

In this method, 20000 images were used and partitioned randomly into training and validation purposes. Five divide modes were considered to study the efficiency of the network. FSDNet gave promising results in facial skin diseases recognition with accuracies of up to 97%. The accuracy and loss in each case are shown in Table 7.1 The accuracy varies between 94.9% and 97% and the loss between 0.2 and 0.09 for split cases 50:50 and 90:10, respectively, and a batch size of 16 and epoch size of 10.

To study how well the diseases are identified, we calculated the confusion matrices presented in Fig. 7.8. These matrices show that few images were wrongly classified. Compared to related works in number and variety of used diseases, our model reveals uniqueness in the performance and implementation. It reached a precision comparable and in some cases higher than that obtained previously. Noting that the majority of works in the state of the art deals with acne and its subtypes.

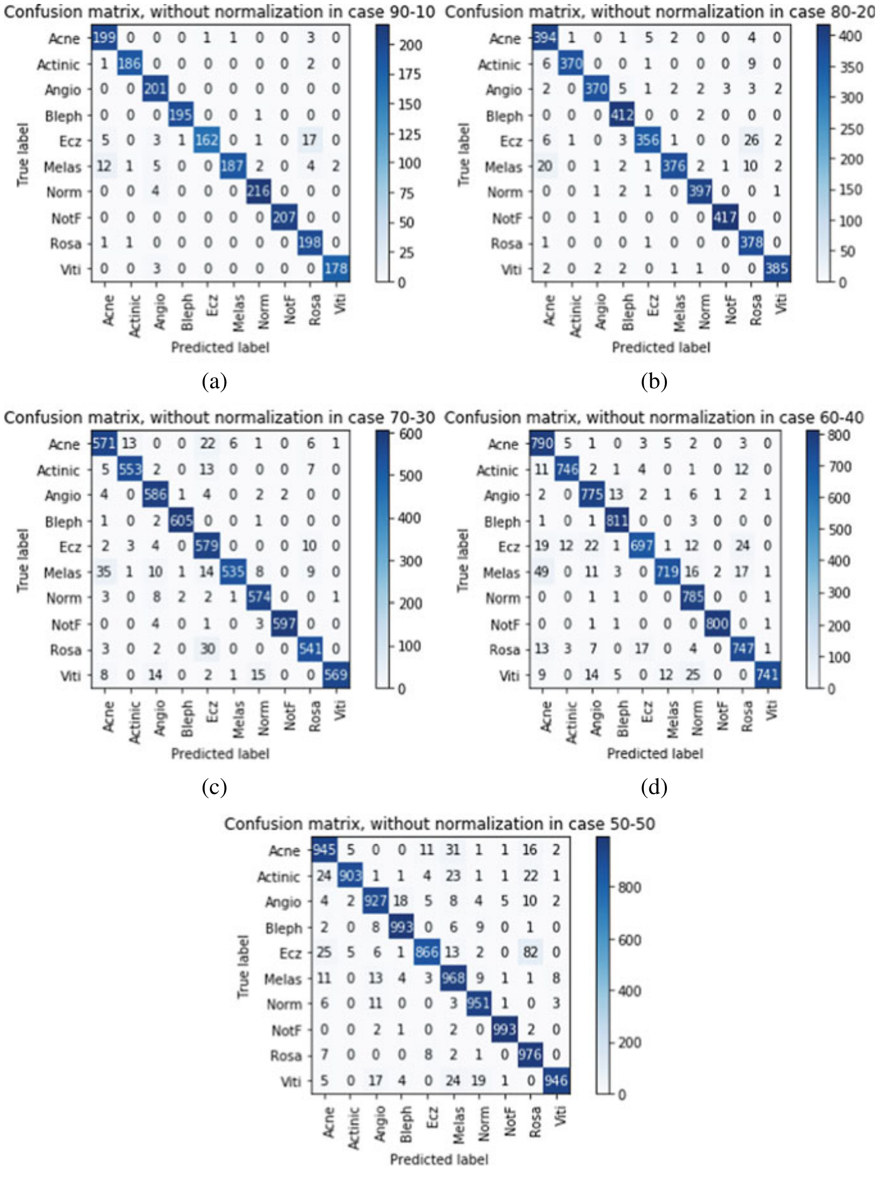
The model performance was tested with images from outside our dataset. They were all well classified accuracies up to 100% (see Fig. 7.9).

7.4 Developed Software

When the training and testing phase is finished, the trained deep learning model is ready to be deployed. Therefore, our biometrics group developed a software to become an assistant to the doctor in his work. This software can assist the doctor in many tasks including, organizing patients profiles, remote communication and consultations with the patients, notification system for easily notifying patients of new changes, classifying skin diseases, issuing treatments and tracking patient condition and treatment progress. The prototype software includes two parts: patient side and doctor side.

Table 7.1 Accuracy of facial skin diseases network

Split of images training (%)/validation (%)	Accuracy (%)	Loss
90/10	97	0.0996
80/20	96.7	0.1452
70/30	95.5	0.2153
60/40	95.4	0.2566
50/50	94.9	0.2791


Confusion matrix, without normalization in case 90-10

(a)

Confusion matrix, without normalization in case 80-20

(b)

Confusion matrix, without normalization in case 70-30

(c)

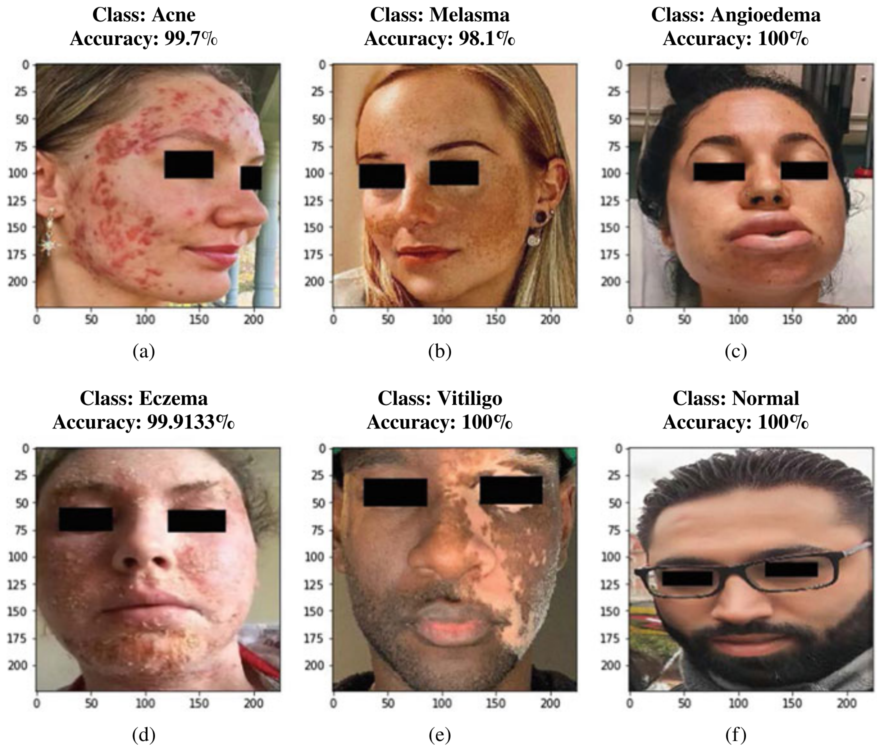
Confusion matrix, without normalization in case 60-40

(d)

Confusion matrix, without normalization in case 50-50

(e)

Fig. 7.8 Confusion matrix in five data split cases for training versus validation sets. **a** 90:10, **b** 80:20, **c** 70:30, **d** 60:40, **e** 50:50


Class: Acne
Accuracy: 99.7%

(a)

Class: Melasma
Accuracy: 98.1%

(b)

Class: Angioedema
Accuracy: 100%

(c)

Class: Eczema
Accuracy: 99.9133%

(d)

Class: Vitiligo
Accuracy: 100%

(e)

Class: Normal
Accuracy: 100%

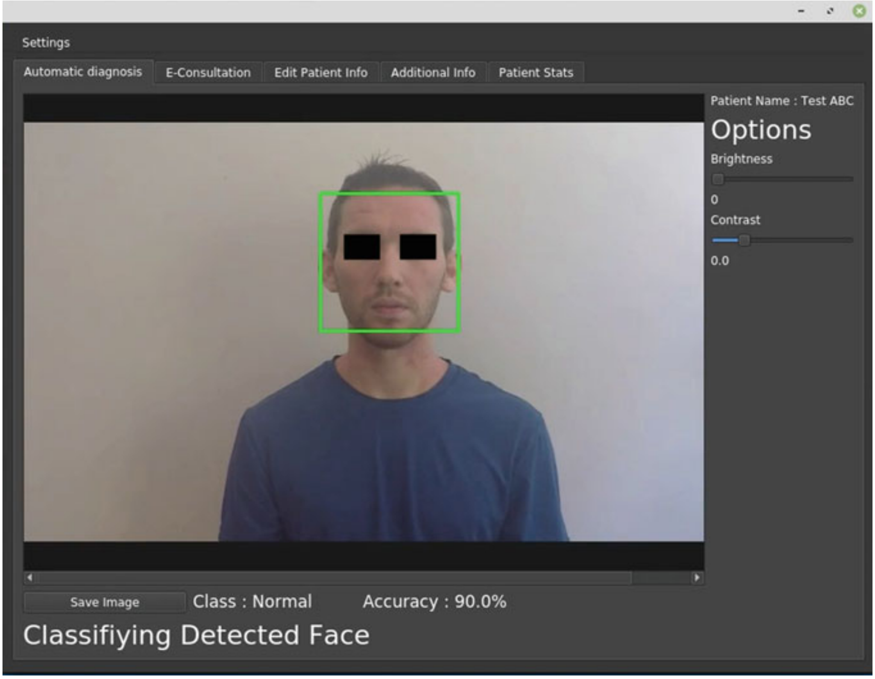
(f)

Fig. 7.9 Test of images out of the database. **a** Acne classified with accuracy of 99.7%, **b** Melasma classified with accuracy of 98.1%, **c** Angioedema classified with accuracy of 100%, **d** Eczema classified with accuracy of 99.9%, **e** Vitiligo classified with accuracy of 100%, **f** normal skin classified with accuracy of 100%

7.4.1 Patient Actions

After creating and submitting the registering form and creating an account, the patient can proceed to start having consultations and treatments from the assigned doctor while using many tools that will help him and the doctor build a patient profile with all the needed data. In the automatic diagnosis tab, the patient has the ability to take images that automatically crops the patient's face, adjust image conditions (brightness and contrast) and save it to the database to be sent to the doctor for evaluation. In addition, the patient can use the trained Deep learning model to get a real-time prediction on the video feed with automatic face detection to identify which class the facial skin disease belongs to (see Fig. 7.10). This AI can be useful as an assistant to the doctor in identifying the skin disease type.

The patient can review the taken images in the E-Consultation tab, delete the selected images, upload new images from a local directory, and use the trained AI model to get class predictions on these locally loaded images. The doctor's diagnosis



Patient Name : Test ABC

Brightness 0

Contrast 0.0

Class : Normal Accuracy : 90.0%

Classifiying Detected Face

Fig. 7.10 Real-time facial skin disease detection

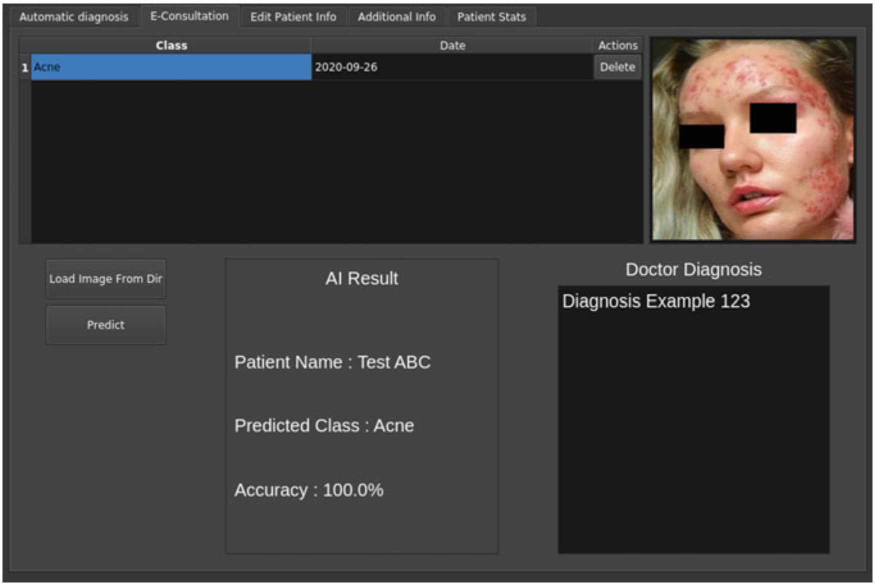
area contains the final doctor's decision about patient condition evaluation (see Fig. 7.11).

A metric will be defined and calculated from several factors in the future to show the progress of the patient's condition. This tab will include several visualizations of the metric starting from the start date of consultation and will include also visualization of treatments. These visualizations will help to track if the progress is becoming worse to do the necessary changes specified by the doctor or when the treatment becomes better then that means the treatment is going well. The chat box feature lets the patients chat with the assigned doctor for discussing the patient's condition or any other matter related to this subject.

7.4.2 Doctor Actions

Patient data tab contains the images that the patient has uploaded using the real-time face capture feed or images uploaded locally. The doctor can use the trained deep learning model as an assistant to his classifying of the facial skin disease.

The doctor can also use the history table to keep track of the patient's progress by adding new records containing the doctor notes taken in the evaluation date and



	Class	Date	Actions
1	Acne	2020-09-26	Delete

AI Result

Patient Name : Test ABC

Predicted Class : Acne

Accuracy : 100.0%

Doctor Diagnosis

Diagnosis Example 123

Fig. 7.11 E-Consultation Tab

entering the metric used to visualize the patient progress. Finally, the doctor can input his final evaluation for the patient to be shown in the doctor's diagnosis area. The face analysis tab provides the doctor with many useful features to evaluate the facial skin disease. This tab contains the following features:

- Automatic selection of face regions
- Manual selection of face regions

This feature allows the doctor to divide the face into the following parts:

- Forehead
- Left Eye
- Right Eye
- Left Cheek
- Right Cheek
- Mouth
- Nose

After applying a transparent mask to visually divide the face to the mentioned parts, the doctor can select each part to zoom in to have a better view of the selected part for a better evaluation. This feature also allows us to gather data to train future deep learning models to classify individual parts of the face. In Fig. 7.12, we can see examples of automatic region selection.

Manual selection of face Regions tool gives the doctor the ability to draw a polygon over the desired area to crop it and visualize it in a new window for better

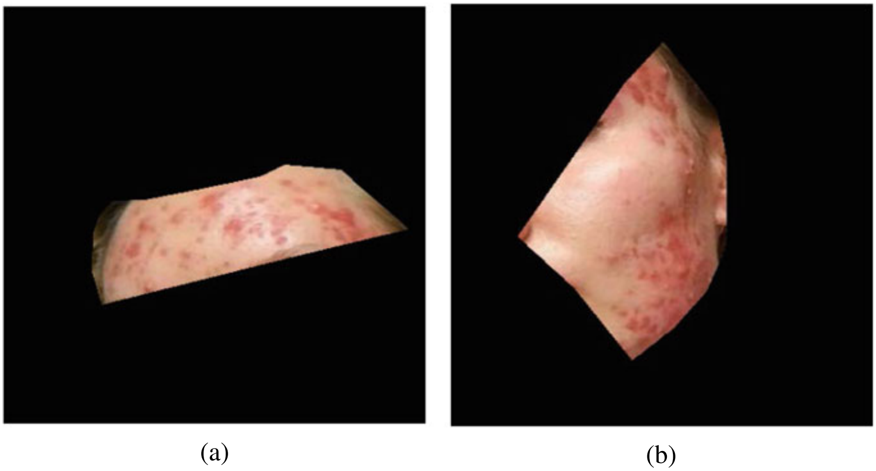
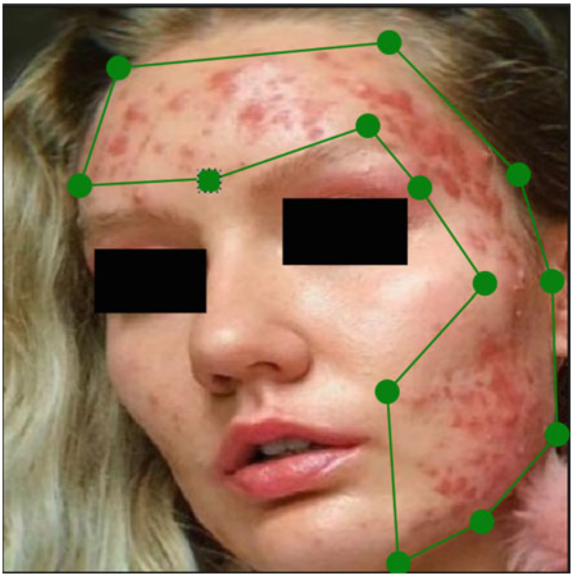


Fig. 7.12 Automatic region selection tool. **a** Forehead region, **b** Right cheek region

Fig. 7.13 Manual face region selection tool



condition evaluation. It can be used as one of the factors defining the metric. Saving the cropped area can also help gather a dataset to train a model for individual face parts classification. Figure 7.13 shows an example of the manual region selection tool.

Allowing the patient to use the software freely without any doctor restrictions can be dangerous for the patients because they might stop consulting the doctor and use it for doing self-evaluations. Therefore, the doctor has access to the a permissions tab, which he can change depending on the access doctors think patients can have.

When the patient submits the information form, the doctor can visualize this information and use them to evaluate the skin disease. It will provide help with ways of treatment. These forms can be useful in gathering datasets that can be used in training a machine learning models to study patterns to detect the cause of some skin diseases or to be able to suggest treatment with the approval of the doctor based on input data of the model. A notification system will be created to allow the doctor to receive a notification about being assigned a new patient and any new changes done by the patient.

In this part, we reviewed the different functionalities and tools in the desktop application developed to help the doctor in the diagnosis process using the trained deep learning model, managing patients' profile quickly, and making remote consultations easier and efficient for the patient. There are plenty of additions that could be done to improve the doctor–patient experience, such as expanding to new popular platforms(web application, mobile application).

7.5 Conclusion

In this chapter, an overview of artificial intelligence as an advanced technology in the field of dermatology is provided through a case study. The focus is on facial skin diseases using deep-learning based identification of skin diseases. One can point out that the whole facial features are considered as input for the AI-system rather than injecting separate lesions. Preliminary results highlight an accuracy of more than 94%. This is, in fact, a promising result when considering the technology as an "assistance-system". At this stage, the role of dermatologists is either to confirm, or correct AI-system estimation. Such actions highly contribute to the increase of accuracy and, consequently, improve the performances. If the system is used by more than one dermatologist, more data is collected and the system will naturally be updated more often. Consequently, AI-systems will be smarter, in the sense that this will narrow down the differences between human-evaluation and machine-evaluation. In the future, it is technically not impossible for AI to provide better diagnosis than humans do.

References

1. Khongsuwan M, Kiattisin S, Wongseree W, Leelasantitham A (2011) Counting number of points for acne vulgaris using UV fluorescence and image processing. IEEE BMEICON Conference, pp 142–146
2. Chang CY, Liao HY (2013) Automatic facial spots and acnes detection system. J. Cosmet. Dermatol. Sci. Appl. 3:28–35

3. Chantharaphaichit T, Uyyanonvara B, Sinthanayothin C, Nishihara A (2015) Automatic acne detection for medical treatment. IEEE IC-ICTES, Conference
4. Kittigul N (2017) Automatic acne detection and quantification for medical treatment through image processing. thesis
5. Shen X, Zhang J, Yan C, Zhou H (2018) An automatic diagnosis method of facial acne vulgaris based on convolutional neural network. Sci Reports 8:5839
6. Amini M, Vasefi F, Valdebran M, Huang K, Zhang H, Kemp W, MacKinnon N (2018) Automated facial acne assessment from smartphone images. In: Proceedings of the SPIE 10497, Imaging, Manipulation, and Analysis of Biomolecules, Cells, and Tissues XVI
7. Binol H, Plotner A, Sopkovich J, Kaffenberger B, Khalil khan Niazi M, Gurcan MN (2020) A deep convolutional neural network for automatic identification of rosacea lesions. Skin Res Technol J 26(3):413–421
8. Yang Y, Ge Y, Guo L, Wu Q, Peng L, Zhang E, XIie J, Li Y, Lin T (2020) Development and validation of two artificial intelligence models for diagnosing benign, pigmented facial skin lesions. Skin Res Technol J. https://doi.org/10.1111/srt.12911
9. Aslam I, Fleischer A, Feldman S (2015) Emerging drugs for the treatment of acne. Expert Opin Emerg Drugs Rev 20(1):91–101. https://doi.org/10.1517/14728214.2015.990373
10. Prajapati V, Barankin B (2008) Dermacase. Actinic keratosis. Can Fam Phys 54(5):691–699. PMC 2377206. PMID 18474700
11. Inceboz T, Yaman A, Over L, Ozturk AT, Akisu C (2009) Diagnosis and treatment of demodectic blepharitis. Turkiye Parazitolojii Dergisi 33(1):32–36. PMID 19367544
12. Bernstein JA, Cremonesi P, Hoffmann TK, Hollingsworth J (2017) Angioedema in the emergency department a practical guide to differential diagnosis and management. Int J Emerg Med. 10(1):15. https://doi.org/10.1186/s12245-017-0141-z
13. Ezzedine K, Eleftheriadou V, Whitton M, van Geel N (2015) Vitiligo Lancet 386(9988):7484. https://doi.org/10.1016/s0140-6736(14)60763-7
14. Tuzun Y, Wolf R, Kutlubay Z, Karakus O, Engin B (2014) Rosacea and rhinophyma. Clin Dermatol 32(1):35–46. https://doi.org/10.1016/j.clindermatol.2013.05.024
15. James W, Berger T, Elston D (2005) andrews diseases of the skin: clinical dermatology, 10th ed. Saunders. ISBN 0-7216-2921-0
16. Nedorost ST (2012) Generalized dermatitis in clinical practice. Springer Science Business Media, Berlin, pp 13–14. ISBN 9781447128977
17. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge, p 326
18. CS231n Convolutional Neural Networks for Visual Recognition. cs231n.github.io. Retrieved 2018-12-13
19. Krizhevsky A, Sutskever I, Hinton GE (2017) ImageNet classification with deep convolution neural networks. Commun ACM 60(6):84–90. https://doi.org/10.1145/3065386
20. LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD (1989) Backpropagation applied to handwritten zip code recognition. Neural Comput 1(4):541–551
21. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
22. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
23. https://dermnetnz.org/topics/facial-skin-problems/

Chapter 8
Medical Imaging Based Diagnosis Through Machine Learning and Data Analysis

Jianjia Zhang, Yan Wang, Chen Zu, Biting Yu, Lei Wang, and Luping Zhou

Abstract Machine learning techniques have played an essential role in computer-assisted medical image analysis. In this chapter, we will introduce several of our recent achievements with machine learning methods for feature extraction and representation, classification, dense prediction (segmentation and synthesis), and multi-modality analysis, across the pipeline of computer-assisted diagnosis (CAD). These methods consist of both traditional machine learning techniques and state-of-the-art deep learning based approaches. They were proposed to address pain points in the techniques, for example, similarity metric learning for better classification, 3D and sample-adaptive dense prediction models for segmentation and synthesis, and effective multi-modal imaging data fusion. These methods have been employed in different levels of medical imaging applications, such as medical image synthesis

Jianjia Zhang, Yan Wang, Chen Zu, Biting Yu—Equal contribution.

J. Zhang
School of Biomedical Engineering, Sun yat-sen University, Guangzhou, China
e-mail: seuzjj@gmail.com

Y. Wang
School of Computer Science, Sichuan University, Chengdu, China
e-mail: wangyanscu@hotmail.com

C. Zu
JD.com, Chengdu, China
e-mail: chenzu@outlook.com

B. Yu · L. Wang
School of Computing and Information Technology, University of Wollongong,
Wollongong, NSW, Australia
e-mail: by354@uowmail.edu.au

L. Wang
e-mail: leiw@uow.edu.au

L. Zhou (✉)
School of Electrical and Information Engineering, University of Sydney,
Camperdown, NSW, Australia
e-mail: luping.zhou@sydney.edu.au

T. D. Pham et al. (eds.), *Advances in Artificial Intelligence, Computation, and Data Science*, Computational Biology 31,
https://doi.org/10.1007/978-3-030-69951-2_8

within and across imaging modalities, brain tumor segmentation, and mental disease classification. Common approaches used for related research topics are also briefly reviewed.

8.1 Introduction

With the pervasive use of medical imaging devices, a large amount of medical imaging data are acquired, stored, processed and analyzed in clinics everyday. Therefore, systems and methods that can automatically analyze medical imaging data are in high demand to reduce doctors' workloads and provide diagnostic measurements that are precise, objective, and reproducible. Traditional medical image analysis methods focused on processing a single image with signal processing techniques. However, evidence from a single image is often insufficient to well represent the complexity of abnormalities and deal with the large variations among patients' imaging data. Consequently, researchers have turned to learning inferences from existing samples using machine learning techniques. Nowadays, machine learning has been widely utilized in the medical imaging field, spreading from low-level pixel processing, such as super-resolution [93], through middle-level ROI (Region of Interest) detection, such as tumor segmentation [21, 54], to high-level diagnostic decision-making, such as disease classification [48, 79, 141] and survival prediction [128], which cover the whole computer-assisted diagnosis pipeline. In this chapter, we are going to briefly review the machine learning techniques used in different levels of medical imaging applications, and focus on introducing several advanced machine learning models from our achievements, which cover two major types of medical image analysis tasks, i.e., medical image classification and dense prediction (e.g., medical image segmentation and synthesis), and extend from single image modalities to multiple imaging modalities.

Specifically, this chapter is organized as follows. First, we introduce classification methods for diagnosis in Sect. 8.2. Rather than relying on intensively studied classifiers, we focus on designing and learning sample similarity measures for better classification. In addition to image-level predictions in Sect. 8.2, in Sect. 8.3 we also address dense pixel-level prediction, which involves both medical image segmentation and synthesis. For segmentation, a deep segmentation model jointly trained with sample-adaptive intensity lookup tables is used to improve brain tumor segmentation. For synthesis, two deep models based on Generative Adversarial Networks (GANs) are introduced for within- and cross-modality image synthesis. In Sect. 8.4, we extend medical image analysis to multi-modal imaging data, and explain two pieces of our work in this field. The first is based on traditional machine learning methods, which jointly learn a similarity measure and select features from multiple modalities. The second is a deep learning based approach, which effectively integrates multiple imaging modalities to synthesize full-dose Positron Emission Tomography (PET) images. This chapter is concluded in Sect. 8.5.

8.2 Classification

A major task of medical image analysis is to assist in diagnostic decisions by labeling (or classifying) the images into different population categories, e.g., diseased or healthy. In machine learning, many linear and nonlinear classifiers have been proposed based on different assumptions about the underlying distribution of the data or the forms of the discriminant functions. Deep learning based classifiers integrate feature learning and classification into a single model. Reviews are in [75, 104], and we will comment but will focus on similarity measures. In most classification algorithms, the decision is based on the distance/similarity between samples, so how this is measured will affect the performance. Common similarity measures are based on Euclidean distances among samples; however, high-dimensional feature vectors of medical imaging data often reside on a low-dimensional subspace/manifold. Considering this structure could provide a more precise similarity measure. Therefore, different types of similarity metrics that could improve classification and how to automatically learn a similarity measure from data are discussed below in this section.

8.2.1 Classifiers

To model a classifier is to estimate a mapping function that takes a data point (e.g., a feature vector) and predicts a discrete variable as a category label. Classification algorithms in medical diagnosis tasks, include traditional non-deep-learning based classifiers such as linear regression [8] and Support Vector Machines [113] (SVM), and deep learning based classifiers.

Traditional classification methods either make assumptions about the underlying data distribution and optimize the distribution parameters by either maximum a posteriori estimation (MAP) or maximum likelihood estimation (MLE) as in Naive Bayesian Classifier and Gaussian Mixture Model or they make assumptions about the form of the discriminant/decision function rather than the data, for example, linear regression and SVM. Linear regression predicts a scalar response by fitting a linear combination of variables of a data point, and a map function, such as a logistic function, can further map predictions to discrete class labels. SVM constructs a hyperplane which admits the largest margin between the two classes to maximize the distance between the nearest data points on each side of the hyperplane which increases the error margin for an unseen point and the generality of the classifier. To address two classes that are not linearly separable in a finite-dimensional space, a nonlinear kernel function can map the original space into a higher dimensional or infinite-dimensional space, making separation easier. A kernel function calculates dot products of pairs of input data vectors in the mapped space and can be considered as a similarity measure. Selection or design of an appropriate kernel function for a specific application is a critical issue. A binary SVM classifier can easily be extended to multi-class cases with one-vs-all or one-vs-one strategies.

Deep learning models take the raw pixels in medical images and do not require feature engineering. Of their various architectures, such as Multilayer Perceptrons (MLP), deep belief networks, and recurrent neural networks, convolutional Neural Networks (CNN) [69] have been widely used for medical image classification [42, 108], segmentation [81], retrieval [92], and reconstruction [100]. A typical CNN model consists of multiple convolution-pooling blocks followed by one (or several) fully connected layer(s) as a classifier. A convolutional layer filters input from the previous layer with a set of fixed-size convolutional kernels and outputs feature maps to the next layer. A pooling layer reduces the dimensions of the feature maps from the previous layer by aggregating them (e.g., by computing the max or average). By stacking the convolutional and pooling layers, a CNN can effectively extract hierarchical patterns corresponding to different levels of semantics. On top of the convolution-pooling blocks, there are usually one or more fully connected layers which flatten the feature maps to output predicted class labels. A CNN is more suitable for image-based analysis, since it significantly reduces the number of model parameters to learn, relative to a classical MLP, by utilizing a local receptive field, weight sharing, sparse connectivity, and preserving spatial information contained in images. Thus "deep" models can extract complex and high-level patterns while reducing the risk of overfitting data. Typical CNN models include

- **AlexNet** [68] which has an eight-layer architecture, utilizes a ReLu (Rectified Linear Units) activation function, dropout, and data augmentation, and was trained on a GPU to improve performance. It achieved a considerable accuracy improvement on ILSVRC2012 [98].
- **VGGNet** [103] increases model depth by using an architecture with very small (3×3) convolution filters and showed a significant performance improvement.
- **GoogLeNet** [107] consists of multiple inception modules that apply parallel filters corresponding to multiple receptive fields to extract features at different levels of detail simultaneously. It increases the depth and width of the network while keeping the computational budget constant by using a carefully crafted design. It also removes the fully connected layers at the end, which significantly reduces the number of model parameters.
- **ResNet** [46] learns residuals rather than the mapping function directly. It makes use of shortcut/skip connections, allowing simple copying of activations from layer to layer to preserve information as data goes through the layers. This mitigates the vanishing gradient problem in training very deep CNN models.

These CNN models originated from generic image recognition tasks and have been applied to effective analysis of medical imaging data. AlexNet [14] and Inception-v3 (GoogLeNet) [43] were used to classify retinal images to diagnose diabetic retinopathy; VGG19 classified skin images based on tumor thickness to diagnose melanoma [57]; and ResNet was used to detect COVID-19 patients based on Chest X-rays [36]. To deal with 3D volumetric medical imaging data, such as CT scans and MRI, 3D CNN models have been developed [104].

Despite the success of deep learning models, non-deep-learning based machine learning techniques still have a role in medical image analysis. Non-deep-learning methods are usually formed in a principled way and are relatively easier to interpret than black-box like deep learning models. This can be critical in medical imaging applications where understanding the results is important. Non-deep-learning methods require less data and may work when only limited training data and computing resources are available.

However, most classifiers are based, explicitly or implicitly, on similarity measures of data points. In the following, two examples, on designing and learning similarity measures in the framework of an SVM to improve classification, are presented.

8.2.2 Example 1: Similarity Metric

Constructing and analyzing functional brain networks based on resting-state fMRI (rs-fMRI) holds great promise for the diagnosis of brain diseases [114], e.g., Attention Deficit Hyperactivity Disorder (ADHD) and Alzheimer's disease (AD). Recent studies have identified ADHD-related brain connectivity differences relative to normal subjects, e.g., increased connectivity between the right pulvinar and bilateral occipital regions and reduced connectivity between the bilateral pulvinar and right prefrontal regions [13, 32, 72]. With these disease-related connectivity differences, diagnosis of brain diseases could be achieved by modeling and classifying brain connectivity networks.

A large body of research work models the functional connectivity based on co-varying patterns of rs-fMRI time series across brain regions via correlation-based statistics. Among them, sparse inverse covariance estimation (SICE) is a principled method for partial correlation estimation, which factors out the effects of other regions while measuring pair-wise regional correlations. In SICE, each individual brain network is represented by an inverse covariance matrix (referred to as a SICE matrix in this chapter) whose off-diagonal entries correspond to the partial correlations between two brain regions. Therefore, how similarity between SICE matrices is measured is fundamental for accurate diagnosis. This section introduces an example [140] that explores the properties of SICE matrices, derives a compact representation method with specifically designed similarity metrics, and demonstrates that this representation will lead to improved diagnosistic performance.

As an inverse covariance matrix, a SICE matrix is symmetric positive definite (SPD), which restricts SICE matrices to a lower dimensional Riemannian manifold rather than the full $d \times d$-dimensional Euclidean space. Brain connectivity patterns are specific and generally similar across different subjects, so SICE matrices representing brain connectivity should be concentrated on an even smaller subset of this manifold. Therefore, the intrinsic degree of freedom of these SICE matrices will be much lower than the apparent dimensions of $d \times d$. These two factors motivate an exploration of a compact representation that better characterizes the underlying distribution of SICE matrices [140].

Linear principal component analysis (PCA), while commonly used for unsupervised dimensionality reduction, is not expected to work well due to the manifold structure of SICE matrices. Several similarity measures have been proposed for SPD matrices that consider the underlying manifold structure. Multiple SPD kernels, such as the Stein [105] and Log-Euclidean [3] kernels, have been proposed and implicitly embed the Riemannian manifold of SPD matrices to a kernel-induced feature space $\mathcal{F}$. They offer a better measure than their counterparts in Euclidean spaces and require less computation than a Riemannian metric [105]. Reference [140] uses these kernels to conduct an SPD-kernel-based PCA and derive a compact representation that can mitigate the curse of dimensionality, as elaborated below.

8.2.2.1 SICE Representation Using SPD-Kernel Based PCA

The SICE method was applied to N subjects to obtain a training set $\{\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_N\} \subset \mathrm{Sym}_d^+$, where $\mathbf{S}_i$ is the SICE matrix for the i-th subject. The kernel mapping is defined as $\Phi(\cdot)$: $\mathrm{Sym}_d^+ \mapsto \mathcal{F}$, which is implicitly induced by a given SPD kernel. Kernel PCA which generalizes linear PCA to a kernel-induced feature space $\mathcal{F}$ is described briefly below (also see [101]). Without loss of generality, assume that $\Phi(\mathbf{S}_i)$ is centered, i.e., $\sum_{i=1}^{N} \Phi(\mathbf{S}_i) = \mathbf{0}$. This can be achieved by computation with a kernel matrix [101]. Then an $N \times N$ kernel matrix $\mathbf{K}$ can be obtained with each entry $\mathbf{K}_{ij} = \langle \Phi(\mathbf{S}_i), \Phi(\mathbf{S}_j) \rangle = k(\mathbf{S}_i, \mathbf{S}_j)$. Kernel PCA first performs the eigen-decomposition on the kernel matrix: $\mathbf{K} = \mathbf{U}\Lambda\mathbf{U}^\top$. The i-th column of $\mathbf{U}$, denoted by $\mathbf{u}_i$, corresponds to the i-th eigenvector, and Λ = diag($\lambda_1, \lambda_2, \ldots, \lambda_N$), where λ_i corresponds to the i-th eigenvalue in descending order. Let $\mathbf{\Sigma}_\Phi$ denote the covariance matrix computed by $\{\Phi(\mathbf{S}_i)\}_{i=1}^N$ in $\mathcal{F}$. The i-th eigenvector of $\mathbf{\Sigma}_\Phi$ can be expressed as

$$\mathbf{v}_i = \frac{1}{\sqrt{\lambda_i}} \mathbf{\Phi} \mathbf{u}_i, \tag{8.1}$$

where $\mathbf{\Phi} = [\Phi(\mathbf{S}_1), \Phi(\mathbf{S}_2), \ldots, \Phi(\mathbf{S}_N)]$.

Analogous to linear PCA, for a given SICE matrix $\mathbf{S}$, $\Phi(\mathbf{S})$ can be projected onto the top m eigenvectors to obtain an m-dimensional principal component vector:

$$\boldsymbol{\alpha} = \mathbf{V}_m^\top \Phi(\mathbf{S}),$$

where $\mathbf{V}_m = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m]$. Note that the i-th component of $\boldsymbol{\alpha}$, denoted by α_i, is $\mathbf{v}_i^\top \Phi(\mathbf{S})$. Using a kernel, it can be computed as

$$\alpha_i = \mathbf{v}_i^\top \Phi(\mathbf{S}) = \frac{1}{\sqrt{\lambda_i}} \mathbf{u}_i^\top \mathbf{\Phi}^\top \Phi(\mathbf{S}) = \frac{1}{\sqrt{\lambda_i}} \mathbf{u}_i^\top \mathbf{k}_\mathbf{S}, \tag{8.2}$$

where $\mathbf{k}_\mathbf{S} = [k(\mathbf{S}, \mathbf{S}_1), k(\mathbf{S}, \mathbf{S}_2), \ldots, k(\mathbf{S}, \mathbf{S}_N)]^\top$. Once $\boldsymbol{\alpha}$ is obtained as a new representation for each SICE matrix, an SVM or k-NN classifier can be trained on $\boldsymbol{\alpha}$ with class labels.

This section studies four commonly used SPD kernels, Cholesky kernel (CHK) [58], Power Euclidean kernel (PEK) [58], Log-Euclidean kernel (LEK) [3], and Stein kernel (SK) [105]. The four kernels are all in the form of

$$k(\mathbf{S}_i, \mathbf{S}_j) = \exp\big(-\theta \cdot d^2(\mathbf{S}_i, \mathbf{S}_j)\big), \tag{8.3}$$

where $d(\cdot,\cdot)$ is a distance between two SPD matrices. Different definitions of $d(\cdot,\cdot)$ lead to different kernels, and the distance functions in the four kernels are Cholesky distance [35], Power Euclidean distance [35], Log-Euclidean distance [3], and root Stein divergence [105], respectively.

Cholesky distance measures the difference between $\mathbf{S}_i$ and $\mathbf{S}_j$ by

$$d(\mathbf{S}_i, \mathbf{S}_j) = ||\,\mathrm{chol}(\mathbf{S}_i) - \mathrm{chol}(\mathbf{S}_j)||_F, \tag{8.4}$$

where $\mathrm{chol}(\mathbf{S})$ is a lower triangular matrix with positive diagonal entries obtained by the Cholesky decomposition of $\mathbf{S}$, that is, $\mathbf{S} = \mathrm{chol}(\mathbf{S})\,\mathrm{chol}(\mathbf{S})^\top$ and $||\cdot||_F$ denotes the Frobenius matrix norm.

Power Euclidean distance between $\mathbf{S}_i$ and $\mathbf{S}_j$ is given by

$$d(\mathbf{S}_i, \mathbf{S}_j) = \frac{1}{p}||\mathbf{S}_i^p - \mathbf{S}_j^p||_F \tag{8.5}$$

where $p \in \mathbb{R}$. Note that $\mathbf{S}$, as a SPD matrix, can be eigen-decomposed as $\mathbf{S} = \mathbf{U}\Lambda\mathbf{U}^\top$, and $\mathbf{S}^p$ can be computed by $\mathbf{S}^p = \mathbf{U}\Lambda^p\mathbf{U}^\top$. In this section, it is set as $p = 0.5$, which gave the best result in the literature [35, 58] and the experiments in this section.

Log-Euclidean distance is defined as

$$d(\mathbf{S}_i, \mathbf{S}_j) = ||\log(\mathbf{S}_i) - \log(\mathbf{S}_j)||_F, \tag{8.6}$$

where $\log(\mathbf{S}) = \mathbf{U}\log(\Lambda)\mathbf{U}^\top$ and $\log(\Lambda)$ takes the logarithm of each diagonal element of Λ.

Root Stein divergence is the square root of Stein divergence, which is defined as

$$d(\mathbf{S}_i, \mathbf{S}_j) = \left[\log\left(\det\left(\frac{\mathbf{S}_i + \mathbf{S}_j}{2}\right)\right) - \frac{1}{2}\log\big(\det(\mathbf{S}_i\mathbf{S}_j)\big)\right]^{\frac{1}{2}}. \tag{8.7}$$

With root Stein divergence as the distance function, the θ in $k(\mathbf{S}_i, \mathbf{S}_j) = \exp\big(-\theta \cdot d^2(\mathbf{S}_i, \mathbf{S}_j)\big)$ is a positive scalar within the range of $\{\frac{1}{2}, \frac{2}{2}, \frac{3}{2}, \ldots, \frac{(d-1)}{2}\} \cup (\frac{(d-1)}{2}, +\infty)$ to guarantee the Stein kernel to be a Mercer kernel [105].

The four distance functions and their kernels are applied to SPD-kernel PCA to produce the principal component vector $\boldsymbol{\alpha}$.

8.2.2.2 Experimental Study

Rs-fMRI data of 44 MCI and 38 NC subjects were downloaded from the ADNI website[1] and used in this study. The data were acquired on a 3 Tesla (Philips) scanner with TR/TE set as 3000/30 ms and flip angle of 80°. Each series has 140 volumes, and each volume consists of 48 slices of image matrices with dimensions 64×64 with voxel size of $3.31 \times 3.31 \times 3.31$ mm^3. Preprocessing was carried out using SPM8[2] and DPARSFA [15]. The first 10 volumes of each series were discarded for signal equilibrium. Slice timing, head motion correction, and MNI space normalization were performed. Participants with too much head motion were excluded. The normalized brain images were warped into the automatic anatomical labeling (AAL) [111] atlas to obtain 90 ROIs as nodes. The ROI mean time series were extracted and band-pass filtered to obtain multiple sub-bands by following the work in [124].

The functional connectivity networks of 82 participants were obtained by the SICE method using SLEP [78], with sparsity levels of $\lambda = [0.1 : 0.1 : 0.9]$. For comparison, constrained sparse linear regression (SLR) [124] was also used to learn functional connectivity networks with the same settings. Functional connectivity networks constructed by SICE and SLR were called "SICE matrices" and "SLR matrices," respectively. To make full use of the limited number of subjects, a leave-one-out procedure was used for training and testing. Both SVM and k-NN were used as the classifier to compare the classification accuracy of different methods. The parameters, including the sparsity level λ, the sub-band of the time series, the number of eigenvectors m, and the regularization parameter of SVM, were tuned by cross-validation on the training set. θ in all the four SPD kernels was empirically set as 0.5, and the k of k-NN was set as 7.

In this experiment, the comparison involved the performance of classification by the compact representation obtained by the proposed SPD-kernel PCA, linear PCA, and the method computing the local clustering coefficient (LCC). LCC, as a measure of local neighborhood connectivity for a node, is defined as the ratio of the number of edges existing between the neighbors of the node and the number of potential connections between these neighbors [63]. In this case, LCC can map a network, represented by a $d \times d$ Adjacency matrix, to a d-dimensional vector, where d is the number of nodes in the network.

Table 8.1 shows the classification results when using the compact representation of SICE or SLR matrices using k-NN with Euclidean distance and a linear kernel SVM. LCC achieved 65.9% for both SICE and SLR matrices with k-NN as the classifier. When linear PCA was applied to the vectorized SICE or SLR matrices to extract the top m principal components as features, the classification accuracy

[1] http://adni.loni.usc.edu.

[2] http://www.fil.ion.ucl.ac.uk/spm/software/.

Table 8.1 Classification accuracy (in %) of compact representation on SICE/SLR matrices

	LCC		Linear PCA		LEK PCA		SK PCA		CHK PCA		PEK PCA	
	k-NN	SVM	k-NN	SVM	k-NN	SVM	k-NN	SVM	k-NN	SVM	k-NN	SVM
SLR [124]	65.9	64.6	67.1	65.9	N.A. Because SLR matrices are not necessarily SPD							
SICE	65.9	63.4	67.1	68.3	69.5	69.5	72	**73.2**	68.3	70.7	72	**73.2**

increased to 67.1% for both SICE and SLR matrices. This indicated the power of compact representation and preliminarily justifies our idea of exploring the lower intrinsic dimensions of the SICE matrices. By taking the SPD property into account and using the proposed SPD-kernel PCA to extract the compact representation, the classification accuracy was significantly boosted to 73.2% for both SK-PCA and PEK-PCA, with SVM as the classifier. This achieved an improvement of 4.9% (73.2% vs. 68.3%) over linear PCA and 7.3% (73.2% vs. 65.9%) over LCC. These results demonstrate that (i) The obtained compact representation can effectively improve the generalization of the classifier in the case of limited training samples. (ii) It is important to consider the manifold property of SICE matrices to obtain a better compact representation.

8.2.3 Example 2: Similarity Learning

Section 8.2.2 demonstrated the importance of selecting an appropriate similarity metric by considering the properties of data. However, existing metrics may not be optimal for certain tasks. In this section, another example [139] is introduced to demonstrate that a similarity metric can be adaptively learned to better align with specific applications.

The Stein kernel (Sect. 8.2.2) that evaluates the similarity between two SPD matrices through their eigenvalues has shown promising performance on classifying images represented by SPD matrices. However, directly using the original eigenvalues cause problems because (i) Eigenvalue estimation becomes biased when the number of samples is inadequate, which may lead to unreliable kernel evaluation; (ii) More importantly, eigenvalues only reflect the property of an individual SPD matrix and are not necessarily optimal for computing a Stein kernel when the goal is to discriminate different classes of SPD matrices. To address these issues, a discriminative Stein kernel was proposed in [139] by introducing an extra parameter vector to adjust the eigenvalues of input SPD matrices. The optimal parameter values were sought by optimizing a proxy of classification performance. To show the generality of the proposed method, three kernel learning criteria that are commonly used in the literature were employed as proxies. The results demonstrated that the discriminative Stein kernel can attain greater discrimination and better align with classification tasks by altering the eigenvalues.

8.2.3.1 The Proposed Method

Let $\lambda_i(\boldsymbol{X})$ denote the ith eigenvalue of a SPD matrix $\boldsymbol{X}$, where $\lambda_i(\boldsymbol{X})$ is always positive due to the SPD property. Throughout this section, it is assumed that the d eigenvalues have been sorted in descending order. Noting that the determinant of $\boldsymbol{X}$ equals $\prod_{i=1}^{d} \lambda_i(\boldsymbol{X})$, the Stein Divergence in Eq. (8.7) can be rewritten as

$$\begin{aligned} S(\boldsymbol{X}, \boldsymbol{Y}) &= \log\left(\det\left(\frac{\boldsymbol{X}+\boldsymbol{Y}}{2}\right)\right) - \frac{1}{2}\log\left(\det(\boldsymbol{X}\boldsymbol{Y})\right) \\ &= \sum_{i=1}^{d} \log \lambda_i \left(\frac{\boldsymbol{X}+\boldsymbol{Y}}{2}\right) - \frac{1}{2}\sum_{i=1}^{d} \left(\log \lambda_i(\boldsymbol{X}) + \log \lambda_i(\boldsymbol{Y})\right). \end{aligned} \tag{8.8}$$

The important role of eigenvalues in computing $S(\boldsymbol{X}, \boldsymbol{Y})$ can be observed. Inappropriate eigenvalues will affect the precision of Stein Divergence and in turn the Stein kernel.

On Issue-I. Eigenvalues of a sample-based covariance matrix are biased estimates of true eigenvalues [102], especially when the number of samples is small.

On Issue-II. Even if true eigenvalues could be obtained, a more important issue exists when the goal is to classify different sets of SPD matrices. A SPD matrix can be expressed as

$$\boldsymbol{X} = \lambda_1 \boldsymbol{u}_1 \boldsymbol{u}_1^\top + \lambda_2 \boldsymbol{u}_2 \boldsymbol{u}_2^\top + \cdots + \lambda_d \boldsymbol{u}_d \boldsymbol{u}_d^\top,$$

where λ_i and $\boldsymbol{u}_i$ denote the ith eigenvalue and the corresponding eigenvector. The magnitude of λ_i only reflects the property of this specific SPD matrix, for example, the data variance along the direction of $\boldsymbol{u}_i$. It does not characterize this matrix from the perspective of discriminating among different sets of SPD matrices. We know that, by fixing the d eigenvectors, varying the eigenvalues changes the matrix $\boldsymbol{X}$. Geometrically, a SPD matrix corresponds to a hyper-ellipsoid in a d-dimensional Euclidean space. This change is analogous to varying the lengths of the axes of the hyper-ellipsoid while maintaining their directions. To make the Stein kernel better prepared for class discrimination, *can we adjust the eigenvalues to make the SPD matrices in the same class as similar to each other as possible while maintaining the SPD matrices across classes to be sufficiently different?* "Similar" and "different" are defined in the sense of the Stein kernel. This idea can be considered as "can an ideal similarity measure be more sensitive to inter-class differences and less affected by intra-class variation?" Without exception, this should also apply to the Stein kernel.

8.2.3.2 Proposed Discriminative Stein Kernel (DSK)

Let $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_d]^\top$ be a vector of adjustment parameters. Let $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top$ denote the eigen-decomposition of a SPD matrix, where the columns of $\boldsymbol{U}$ correspond to the eigenvectors and $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_d)$. $\boldsymbol{\alpha}$ is used for eigenvalue adjustment and to define the adjusted $\boldsymbol{X}$ as

$$\tilde{\boldsymbol{X}}_p = \boldsymbol{U} \begin{pmatrix} \lambda_1^{\alpha_1} & & & \\ & \lambda_2^{\alpha_2} & & \\ & & \ddots & \\ & & & \lambda_d^{\alpha_d} \end{pmatrix} \boldsymbol{U}^\top \tag{8.9}$$

$$\text{and } \tilde{\boldsymbol{X}}_c = \boldsymbol{U} \begin{pmatrix} \alpha_1\lambda_1 & & & \\ & \alpha_2\lambda_2 & & \\ & & \ddots & \\ & & & \alpha_d\lambda_d \end{pmatrix} \boldsymbol{U}^\top. \tag{8.10}$$

In the first case, $\boldsymbol{\alpha}$ is used as the *power* of eigenvalues. It can naturally maintain the SPD property because $\lambda_i^{\alpha_i}$ is always positive. In the second case, $\boldsymbol{\alpha}$ is used as the *coefficient* of eigenvalues. It is mathematically simpler but needs to impose the constraint $\alpha_i > 0$ $(i = 1, \ldots, d)$ to maintain the SPD property. The two adjusted matrices are denoted by $\tilde{\boldsymbol{X}}_p$ and $\tilde{\boldsymbol{X}}_c$, where p and c refer to "power" and "coefficient". Both cases are investigated in this section.

Given two SPD matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$, the $\boldsymbol{\alpha}$-adjusted Stein Divergence is defined as

$$S_{\boldsymbol{\alpha}}(\boldsymbol{X}, \boldsymbol{Y}) \triangleq S(\tilde{\boldsymbol{X}}, \tilde{\boldsymbol{Y}}). \tag{8.11}$$

For the two ways of using $\boldsymbol{\alpha}$, the term $S(\tilde{\boldsymbol{X}}, \tilde{\boldsymbol{Y}})$ can be expressed as

$$\begin{aligned} S(\tilde{\boldsymbol{X}}_p, \tilde{\boldsymbol{Y}}_p) &= \sum_{i=1}^{d} \log \lambda_i \left(\frac{\tilde{\boldsymbol{X}}_p + \tilde{\boldsymbol{Y}}_p}{2} \right) \\ &\quad - \frac{1}{2} \sum_{i=1}^{d} \alpha_i \left(\log \lambda_i(\boldsymbol{X}) + \log \lambda_i(\boldsymbol{Y}) \right) \end{aligned}$$

$$\begin{aligned} \text{and } S(\tilde{\boldsymbol{X}}_c, \tilde{\boldsymbol{Y}}_c) &= \sum_{i=1}^{d} \log \lambda_i \left(\frac{\tilde{\boldsymbol{X}}_c + \tilde{\boldsymbol{Y}}_c}{2} \right) \\ &\quad - \frac{1}{2} \sum_{i=1}^{d} \left(2 \log \alpha_i + \log \lambda_i(\boldsymbol{X}) + \log \lambda_i(\boldsymbol{Y}) \right). \end{aligned}$$

Based on the above definition, the discriminative Stein kernel (DSK) is proposed as

$$k_{\boldsymbol{\alpha}}(\boldsymbol{X}, \boldsymbol{Y}) = \exp\left(-\theta \cdot S_{\boldsymbol{\alpha}} \left(\boldsymbol{X}, \boldsymbol{Y} \right) \right). \tag{8.12}$$

Note that the DSK will remain a Mercer kernel as long as θ varies in the range of $\boldsymbol{\Theta}$ defined in Root Stein divergence in Sect. 8.2.2.1, because $k_{\boldsymbol{\alpha}}(\boldsymbol{X}, \boldsymbol{Y})$ can always be viewed as $k(\tilde{\boldsymbol{X}}, \tilde{\boldsymbol{Y}})$, the original Stein kernel applied to two adjusted SPD matrices $\tilde{\boldsymbol{X}}$ and $\tilde{\boldsymbol{Y}}$.

Treating $\boldsymbol{\alpha}$ as the kernel parameter of $k_{\boldsymbol{\alpha}}(\boldsymbol{X}, \boldsymbol{Y})$, kernel learning techniques are applied to find its optimal value. Many learning criteria such as kernel alignment [30], kernel class separability [115], and radius margin bound [16] have been proposed. In this work, to investigate the generality of the proposed DSK, all three criteria are employed to solve the kernel parameters $\boldsymbol{\alpha}$.

Let $\boldsymbol{\Omega} = \{(\boldsymbol{X}_i, t_i)\}_{i=1}^{n}$ be a set of n training SPD matrices, each of which represents a sample, e.g., an image to be classified. t_i denotes the class label of the ith sample, where $t_i \in \{1, \ldots, M\}$ with M denoting the number of classes. $\boldsymbol{K}$ denotes the kernel matrix computed with DSK on $\boldsymbol{\Omega}$, with $\boldsymbol{K}_{ij} = k_{\boldsymbol{\alpha}}(\boldsymbol{X}_i, \boldsymbol{X}_j)$. In the following part, three frameworks are developed to learn the optimal value of $\boldsymbol{\alpha}$.

8.2.3.3 Kernel Alignment Based Framework

Kernel alignment measures the similarity of two kernel functions and can be used to quantify the degree of agreement between a kernel and a given classification task [30]. With kernel alignment, the optimal $\boldsymbol{\alpha}$ can be obtained through optimization as

$$\boldsymbol{\alpha}^* = \arg\max_{\boldsymbol{\alpha}\in\mathcal{A}} J(\boldsymbol{K}, \boldsymbol{T}) - \lambda\|\boldsymbol{\alpha} - \boldsymbol{\alpha}_0\|_2^2, \tag{8.13}$$

where $\boldsymbol{T}$ is an $n \times n$ matrix with $\boldsymbol{T}_{ij} = 1$ if $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ are from the same class and $\boldsymbol{T}_{ij} = -1$ otherwise. This definition of $\boldsymbol{T}$ naturally handles multi-class classification. $J(\boldsymbol{K}, \boldsymbol{T})$ is defined as the kernel alignment criterion:

$$J(\boldsymbol{K}, \boldsymbol{T}) = \frac{\langle \boldsymbol{T}, \boldsymbol{K}\rangle_F}{\sqrt{\langle \boldsymbol{T}, \boldsymbol{T}\rangle_F \langle \boldsymbol{K}, \boldsymbol{K}\rangle_F}}, \tag{8.14}$$

where $\langle\cdot, \cdot\rangle_F$ denotes the Frobenius inner product between two matrices. $J(\boldsymbol{K}, \boldsymbol{T})$ measures the degree of agreement between $\boldsymbol{K}$ and $\boldsymbol{T}$, where $\boldsymbol{T}$ is regarded as the ideal kernel of a learning task. The $\boldsymbol{\alpha}_0$ is a priori estimate of $\boldsymbol{\alpha}$, and $\|\boldsymbol{\alpha} - \boldsymbol{\alpha}_0\|_2^2$ is the regularizer which constrains $\boldsymbol{\alpha}$ to be around $\boldsymbol{\alpha}_0$ to avoid overfitting. We can simply set $\boldsymbol{\alpha}_0 = [1, \ldots, 1]^\top$, which corresponds to the original Stein kernel. λ is the regularization parameter to be selected via cross-validation. $\mathcal{A}$ denotes the domain of $\boldsymbol{\alpha}$: when $\boldsymbol{\alpha}$ is used as a power, $\mathcal{A}$ denotes a Euclidean space $\mathbb{R}^d$; when $\boldsymbol{\alpha}$ is used as a coefficient, $\mathcal{A}$ is constrained to $\mathbb{R}_+^d$.

As seen in Eq. (8.12), there is a kernel parameter θ inherited from the original Stein kernel. θ and $\boldsymbol{\alpha}$ play different roles in the proposed kernel and cannot be replaced with each other. The value of θ needs to be appropriately chosen because it impacts the kernel value and in turn the optimization of $\boldsymbol{\alpha}$. A commonly used way to tune θ is k-fold cross-validation. In this section, to better align with the kernel alignment criterion, θ is tuned by maximizing the kernel alignment before adjusting $\boldsymbol{\alpha}$,

$$\theta^* = \arg\max_{\theta\in\boldsymbol{\Theta}} J(\boldsymbol{K}|_{\boldsymbol{\alpha}=\mathbf{1}}, \boldsymbol{T}), \tag{8.15}$$

where $\mathbf{1}$ is a d-dimensional vector with all entries equal to 1 and $\boldsymbol{K}|_{\boldsymbol{\alpha}=\mathbf{1}}$ denotes the kernel matrix computed by the original Stein kernel without $\boldsymbol{\alpha}$-adjustment. Through this optimization, a reasonably good θ is found and then $\boldsymbol{\alpha}$ is optimized. The maximization problem in Eq. (8.15) can be conveniently solved by choosing θ in the range of $\boldsymbol{\Theta} = \left\{\frac{1}{2}, \frac{2}{2}, \frac{3}{2}, \ldots, \frac{d-1}{2}\right\} \cup \left(\frac{d-1}{2}, +\infty\right)$. θ is not optimized jointly with $\boldsymbol{\alpha}$ since the noncontinuous range of θ could complicate gradient-based optimization. As will be shown in the experimental study, optimizing θ and $\boldsymbol{\alpha}$ sequentially can lead to promising results.

After obtaining θ^* and $\boldsymbol{\alpha}^*$, the proposed DSK will be applied to both training and test data for classification, with classifiers such as k-nearest neighbor (k-NN) or SVM. For a given classification task, the optimization of θ and $\boldsymbol{\alpha}$ only needs to be conducted *once* with training data. After that, they are used as fixed parameters to compute the Stein kernel for each pair of SPD matrices.

8.2.3.4 Class Separability Based Framework

Class separability is another commonly used criterion for model and feature selection [115, 116, 130]. For a training sample set defined as $\boldsymbol{\Omega} = \{(\boldsymbol{X}_i, t_i)\}_{i=1}^n$, where $t_i \in \{1, \ldots, M\}$, let $\boldsymbol{\Omega}_i$ be the set of training samples from the ith class, with n_i denoting the size of $\boldsymbol{\Omega}_i$. $\boldsymbol{K}_{\boldsymbol{\Omega}',\boldsymbol{\Omega}''}$ denotes a kernel matrix computed over two training subsets $\boldsymbol{\Omega}'$ and $\boldsymbol{\Omega}''$, where $\{\boldsymbol{K}_{\boldsymbol{\Omega}',\boldsymbol{\Omega}''}\}_{ij} = k(\boldsymbol{X}_i, \boldsymbol{X}_j) = \langle\phi(\boldsymbol{X}_i), \phi(\boldsymbol{X}_j)\rangle$ with $\boldsymbol{X}_i \in \boldsymbol{\Omega}'$ and $\boldsymbol{X}_j \in \boldsymbol{\Omega}''$. The class separability in the feature space $\mathcal{F}$ induced by a kernel k can be defined as

$$J = \frac{\mathrm{tr}(\boldsymbol{S}_B)}{\mathrm{tr}(\boldsymbol{S}_W)}, \tag{8.16}$$

where $\mathrm{tr}(\cdot)$ is the trace of a matrix, and $\boldsymbol{S}_B$ and $\boldsymbol{S}_W$ are the *between-class scatter matrix* and the *within-class scatter matrix*, respectively. Let $\boldsymbol{m}$ and $\boldsymbol{m}_i$ denote the total sample mean and the ith class mean. They can be expressed as $\boldsymbol{m} = \frac{1}{n}\sum_{\boldsymbol{X}_i \in \boldsymbol{\Omega}} \phi(\boldsymbol{X}_i)$ and $\boldsymbol{m}_i = \frac{1}{n_i}\sum_{\boldsymbol{X}_j \in \boldsymbol{\Omega}_i} \phi(\boldsymbol{X}_j)$.

The terms $\mathrm{tr}(\boldsymbol{S}_B)$ and $\mathrm{tr}(\boldsymbol{S}_W)$ can be expressed as

$$\begin{aligned} \mathrm{tr}(\boldsymbol{S}_B) &= \mathrm{tr}\left[\sum_{i=1}^{M} n_i\,(\boldsymbol{m}_i - \boldsymbol{m})\,(\boldsymbol{m}_i - \boldsymbol{m})^\top\right] \\ &= \sum_{i=1}^{M} \frac{\mathbf{1}^\top \boldsymbol{K}_{\boldsymbol{\Omega}_i,\boldsymbol{\Omega}_i}\mathbf{1}}{n_i} - \frac{\mathbf{1}^\top \boldsymbol{K}_{\boldsymbol{\Omega},\boldsymbol{\Omega}}\mathbf{1}}{n}, \end{aligned} \tag{8.17}$$

and

$$\begin{aligned}\operatorname{tr}(\boldsymbol{S}_W) &= \operatorname{tr}\left[\sum_{i=1}^{M}\sum_{j=1}^{n_i}\Big(\phi\left(\boldsymbol{X}_{ij}\right)-\boldsymbol{m}_i\Big)\Big(\phi\left(\boldsymbol{X}_{ij}\right)-\boldsymbol{m}_i\Big)^{\top}\right]\\ &= \operatorname{tr}\left(\boldsymbol{K}_{\boldsymbol{\Omega},\boldsymbol{\Omega}}\right)-\sum_{i=1}^{M}\frac{\mathbf{1}^{\top}\boldsymbol{K}_{\boldsymbol{\Omega}_i,\boldsymbol{\Omega}_i}\mathbf{1}}{n_i},\end{aligned} \tag{8.18}$$

where $\mathbf{1} = [1, 1, \ldots, 1]^{\top}$.

The class separability can reflect the goodness of a kernel function with respect to a given task.

8.2.3.5 Radius Margin Bound Based Framework

The radius margin bound is an upper bound on the number of classification errors in a leave-one-out (LOO) procedure of a hard margin binary SVM [16, 67]. This bound can be extended to an L_2-norm soft margin SVM with a slightly modified kernel. It has been widely used for parameter tuning [16] and model selection [116]. We first consider a binary classification task and then extend the result to the multi-class case. Let $\boldsymbol{\Omega}' \cup \boldsymbol{\Omega}''$ be a training set of l samples, and without loss of generality, the samples are labeled by $t \in \{-1, 1\}$. With a given kernel function k, the optimization problem of a SVM with an L_2-norm soft margin can be expressed as

$$\begin{aligned}\frac{1}{2}\|\boldsymbol{w}\|^2 = \quad & \max_{\eta\in\mathbb{R}^l}\left[\sum_{i=1}^{l}\eta_i-\frac{1}{2}\sum_{i,j=1}^{l}\eta_i\eta_j t_i t_j\tilde{k}(\boldsymbol{X}_i,\boldsymbol{X}_j)\right]\\ \text{subject to} \quad & \sum_{i=1}^{l}\eta_i t_i = 0;\;\; \eta_i \ge 0\;(i=1,2,\ldots,l),\end{aligned} \tag{8.19}$$

where $\tilde{k}(\boldsymbol{X}_i, \boldsymbol{X}_j) = k(\boldsymbol{X}_i, \boldsymbol{X}_j) + \frac{1}{C}\delta_{ij}$; C is the regularization parameter; $\delta_{ij} = 1$ if $i = j$, and 0 otherwise; and $\boldsymbol{w}$ is the normal vector of the optimal separating hyperplane of the SVM. Tuning of the parameters in $\tilde{k}$ can be achieved by minimizing an estimate of the LOO errors. The following *radius margin bound* holds [113]:

$$E(\text{LOO}) \le 4\cdot\frac{R^2}{\gamma^2} = 4R^2\|\boldsymbol{w}\|^2, \tag{8.20}$$

where $E(\text{LOO})$ denotes the number of LOO errors performed on the l training samples in $\boldsymbol{\Omega}' \cup \boldsymbol{\Omega}''$; R is the radius of the smallest sphere enclosing all the l training samples; and γ denotes the margin with respect to the optimal separating hyperplane and equals $1/\|\boldsymbol{w}\|$. R^2 can be obtained by the following optimization problem,

$$\begin{aligned} R^2 &= \max_{\beta \in \mathbb{R}^l} \Big[\sum_{i=1}^{l} \beta_i \tilde{k}(\boldsymbol{X}_i, \boldsymbol{X}_i) - \sum_{i,j=1}^{l} \beta_i \beta_j \tilde{k}(\boldsymbol{X}_i, \boldsymbol{X}_j) \Big] \\ &\text{subject to: } \sum_{i=1}^{l} \beta_i = 1; \quad \beta_i \geq 0 \ \ (i = 1, 2, \ldots, l). \end{aligned} \tag{8.21}$$

Both R and $\boldsymbol{w}$ are functions of the kernel $\tilde{k}$. The kernel function k is set as $k_{\boldsymbol{\alpha}}$ defined in Eq. (8.12). The model parameters in $\tilde{k}$, i.e., $\{\theta, \boldsymbol{\alpha}, C\}$, can be optimized by minimizing $R^2\|\boldsymbol{w}\|^2$ on the training set. A reasonably good θ^* can be chosen by optimizing Eq. (8.22) with respect to θ and C while fixing $\boldsymbol{\alpha}$ as $\mathbf{1}$.

$$\{\theta^*, C^*\} = \arg \min_{\boldsymbol{\theta} \in \boldsymbol{\Theta}, C>0 | \boldsymbol{\alpha}=\mathbf{1}} R^2\|\boldsymbol{w}\|^2. \tag{8.22}$$

Once θ^* is obtained, $\{\boldsymbol{\alpha}, C\}$, denoted by $\boldsymbol{\upsilon}$, can then be jointly optimized as follows:

$$\boldsymbol{\upsilon}^* = \arg \min_{\boldsymbol{\upsilon} \in \boldsymbol{\Upsilon} | \theta=\theta^*} R^2\|\boldsymbol{w}\|^2, \tag{8.23}$$

where $\boldsymbol{\Upsilon} = \{\boldsymbol{\alpha}, C | \boldsymbol{\alpha} \in \mathcal{A}; C > 0\}$. Let υ_z be the zth parameter of $\boldsymbol{\upsilon}$.

For multi-class classification tasks, $\boldsymbol{\upsilon}$ can be optimized by a pair-wise combination of the radius margin bounds of binary SVM classifiers [116].

8.2.3.6 Experimental Result

In this experiment, the proposed discriminative Stein kernel (DSK) was compared with the original Stein kernel (SK) on the resting-state functional Magnetic Resonance Imaging (rs-fMRI) dataset from the ADNI benchmark database. We employed both k-NN and SVM as the classifiers. For the kernel alignment and class separability frameworks, k-NN was used with the DSK as the similarity measure, since it does not involve any other (except k) algorithmic parameter. This allowed the comparison to directly reflect the change from SK to DSK. For the radius margin bound framework, a SVM classifier was used since it is inherently related to this bound.

In this experiment, the DSK obtained by the kernel alignment and class separability were called DSK-KA and DSK-CS. Also, DSK-RM indicates the DSK obtained by the radius margin bound, while DSK-TM denotes the DSK obtained by the trace margin criterion. Subscripts p or c are used to indicate whether $\boldsymbol{\alpha}$ acts as the power or the coefficient of eigenvalues. The names are summarized in Table 8.2. All parameters, including the k of k-NN, the regularization parameter of the SVM, λ in Eq. (8.13), θ in all the kernels, and the power order ζ in the Power Euclidean metric are chosen via multi-fold cross-validation on the training set.

DSK was tested on brain imaging analysis using a correlation matrix, a SPD matrix in which each off-diagonal element denotes the correlation coefficient between a pair of variables. It is commonly used in neuroimaging analysis to model functional

Table 8.2 The name of DSK under different learning criteria

$\boldsymbol{\alpha}$ as	Kernel alignment	Class separability	Radius margin bound	Trace margin criterion
Power	DSK-KA$_p$	DSK-CS$_p$	DSK-RM$_p$	DSK-TM$_p$
Coefficient	DSK-KA$_c$	DSK-CS$_c$	DSK-RM$_c$	DSK-TM$_c$

Table 8.3 Comparison of classification accuracy (in percentage) on fMRI data

k-NN					SVM				
Competing methods			DSK (proposed)		Competing methods			DSK (proposed)	
AIRM	CHK	EUK	DSK-KA$_p$	DSK-KA$_c$	AIRM	CHK	EUK	DSK-RM$_p$	DSK-RM$_c$
56.10	52.44	50.00	60.98	59.76	N.A.	53.66	54.88	**59.76**	54.88
SK	LEK	PEK	DSK-CS$_p$	DSK-CS$_c$	SK	LEK	PEK	DSK-TM$_p$	DSK-TM$_c$
56.10	54.88	51.22	60.98	**62.20**	54.88	53.66	57.32	**59.76**	53.66

brain networks to discriminate patients with Alzheimer's Disease (AD) from healthy controls [123]. In this task, a correlation matrix was extracted from each rs-fMRI image and used to represent the brain network of the corresponding subject. This is also a classification task involving SPD matrices.

The rs-fMRI dataset from ADNI consists of 44 patients with mild cognitive impairment (MCI, an early warning stage of Alzheimer's disease), and 38 healthy controls. The rs-fMRI images of these subjects were pre-processed by a standard pipeline using SPM8 (http://www.fil.ion.ucl.ac.uk/spm) for rs-fMRI. All the images were spatially normalized into a common space and parcellated into 116 regions of interest (ROI) based on a predefined brain atlas. 42 ROIs that are known to be related to AD were selected [51] in our experiment and the mean rs-fMRI signal within each ROI was extracted as the features. A 42×42 correlation matrix was then constructed for each subject [123].

In this experiment, DSK was compared with the other methods to classify the correlation matrices. The classification was conducted in the LOO manner due to the limited number of samples. This process was repeated for each of the samples. DSK achieved the best classification performance (Table 8.3). DSK-CS$_c$ increased the classification accuracy of SK from 56.1% to 62.2% with k-NN as the classifier. DSK-RM$_p$ obtained an improvement of 4.9 percentage points over SK when SVM was used as the classifier. This experimental result indicated that DSK holds promise for the classification of SPD matrices.

8.3 Dense Prediction

For classification tasks, prediction is made at the image-level, i.e., the class label is assigned to the whole image. For some medical imaging applications prediction is required at the pixel-level. This is known as dense prediction, i.e., predicting the value at every single pixel position in an image. Two typical dense prediction problems are medical image segmentation and synthesis. Medical image segmentation corresponds to per-pixel classification as it categories every image pixel into Regions of Interest (ROI) or background, while medical image synthesis corresponds to per-pixel regression as it estimates the intensity value at each pixel in a transformed image. Two dense prediction problems are elaborated in detail.

8.3.1 Segmentation

8.3.1.1 Overview

Medical image segmentation identifies regions of interest and delineates their contours giving shape and volume related clinical parameters to assist computer-aided diagnosis and treatment planning. Deep convolutional neural networks (CNNs) have been applied to segmentation tasks, such as organ, substructure, or lesion segmentation [74]. This use of deep CNNs in segmentation applications started from introducing classification models into pixel-/voxel-wise labeling [9, 24, 38, 71, 83, 84, 95, 109, 146]. These models conducted patch-to-pixel/voxel segmentation where each cropped patch from a given image was taken as input. After processing by convolutional and fully connected layers, CNNs estimate the segmentation label of its center pixel/voxel. By moving sliding windows over the entire image, the final segmentation result can be predicted. This approach was first applied by 2D CNNs to axial slices of 3D medical images [24, 38, 71, 83, 84, 109]. Then, 2.5D CNNs were proposed to extract features from all three views of medical images [9, 95, 146]. The axial, coronal, and sagittal patch slices that have the same central voxel are separately put into the three input pathways of these CNNs, and then their deep feature maps are fused to estimate the final label. With this strategy, the label annotation of each voxel can see the three views of its surroundings, so that more volumetric information will be captured, and the segmentation accuracy will be improved.Some effective approaches from generic image segmentation, like multi-scale feature extraction, residual learning, and cascade structure, were integrated into these CNNs to promote their segmentation performance [9, 24, 83, 84]. Direct application of classification models to infer segmentation has low efficacy. Insufficient acquisition of global-level information in classification models lowers their performance. Fully convolutional networks (FCNs) [80] that generated same-size labels on generic images were explored for medical image segmentation and extended to 2D, 2.5D, and 3D variants [34, 65, 97, 110, 136, 144]. A bounding box was introduced to better delineate small objects, like pancreas, in larger images, and deep supervision and residual learning strategies were used to build more back-propagation paths during training [34, 136]. CNNs with Unet-like structures that consist of contracting and expanding paths with skip connections [94] were used for global-level segmentation to mitigate the issue of vanishing gradients [26, 54, 55, 59, 64, 73, 86, 87, 91, 145]. Some exploited the inter-slice spatial dependence on 3D medical images [91] and connections among neighboring and distant feature maps [81] to increase segmentation accuracy. Dense blocks and dilated convolutional layers were incorporated into the architectures [33, 85], respectively, to enhance learning of hierarchical visual information.

8.3.1.2 Example: Learning Sample-Adaptive Intensity Lookup Table for Brain Tumor Segmentation [135]

Learning sample-adaptive intensity lookup tables (LuTs) were used to explicitly handle variation in MR intensity in a brain tumor segmentation framework [135]. An intensity LuT corresponds to a nonlinear mapping function that could be used to adjust the intensity levels of MR images from one set to another. A sample-specific mapping function may be needed and could be guided by the performance of the segmentation. Figure 8.1 illustrates the SA-LuT-Net framework [135], which incorporates a LuT module and a subsequent segmentation module. The LuT module generates the sample-adaptive parameters to determine mapping functions that are applied to change the intensity contrast of the MR image and applied to the segmentation module. Through end-to-end training, the LuT and segmentation modules are jointly trained and negotiated with each other to achieve the optimal segmentation result.

Diverse nonlinear functions could be used to model the intensity transformation mappings of LuTs. For simplicity and flexibility, a family of nonlinear functions, i.e., piece-wise linear functions, is used in this work. A three-segment piece-wise linear function is plotted in Fig. 8.2a as an example, and is mathematically formulated as

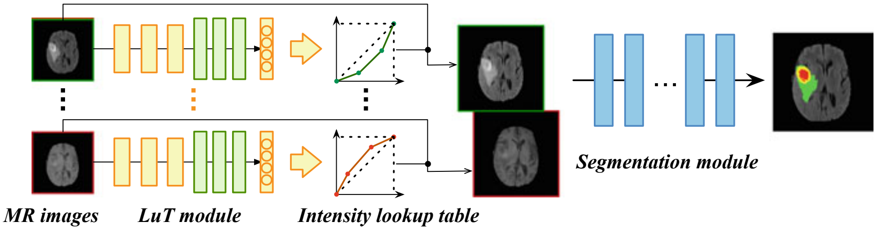


Fig. 8.1 Overview of SA-LuT-Net framework under a multi-modality scenario

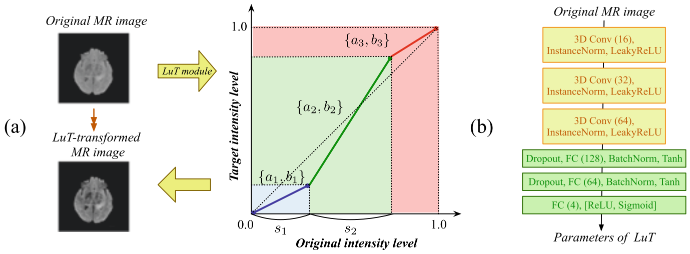


Fig. 8.2 Two intensity lookup tables using a piece-wise linear function, to transform the input intensity levels in an MR image to target levels by estimated sample-specific parameters of mapping functions

$$\hat{\mathbf{x}} = \begin{cases} a_1\mathbf{x} + b_1, & 0 \le \mathbf{x} < s_1, \\ a_2\mathbf{x} + b_2, & s_1 \le \mathbf{x} \le (s_1 + s_2), \\ a_3\mathbf{x} + b_3, & (s_1 + s_2) < \mathbf{x} \le 1, \end{cases} \tag{8.24}$$

where the intensity levels in the given MR image and its LuT-transformed image are indicated by $\mathbf{x}$ and $\hat{\mathbf{x}}$, respectively. To estimate the parameters of mapping functions, the LuT module consists of convolutional blocks and fully connected (FC) blocks, as shown in Fig. 8.2b.

Different segmentation backbones could be employed as the segmentation module of SA-LuT-Net. DMFNet [21] and a modified 3D Unet model [54], two advanced brain tumor segmentation models, were applied as the backbones in this work. The first backbone network, DMFNet [21], has 69 convolutional layers built as a 3D Unet-like structure with skip connections. It replaces the ordinary convolutional layers in its first six encoding residual units with more efficient adaptive dilated multi-fiber layers to capture multi-scale feature representations from brain tumor images. It attained comparable results on the BRATS2018 validation set [4] to the challenge first-placed model, NVDLMED [86], but requires about 1/10 of the learning parameters of NVDLMED. The modified 3D Unet model [54], contains 26 convolutional layers. It incorporates residual connections in its encoding convolutional blocks to combine neighboring shallow and deep features, and integrates different-depth segmentation outputs in an element-wise summation way to apply deep supervision for the final model output and achieved third place in BRATS2017 [4].

The SA-LuT-Net framework was evaluated on the FLAIR images from the training set of the BRATS2018 [4] dataset by fivefold cross-validation, and compared with its baselines, DMFNet [21] and modified 3D Unet model [54]. Their Dice scores are presented in Table 8.4. SA-LuT-Nets achieved significant improvements over the baselines. Improvement on segmenting the TC region was the largest on both backbone models.

Table 8.4 Dice scores of FLAIR segmentation results on the BRATS2018 training set, reported by mean(std)

Methods	WT	TC	ET
Modified 3D Unet [54] (baseline)	0.8437(0.1520)	0.5912(0.2104)	0.3520(0.2342)
SA-LuT-Net (3D Unet based)	**0.8621(0.1258)**	**0.6450(0.1968)**	**0.3959(0.2647)**
DMFNet [21] (baseline)	0.8549(0.1031)	0.5499(0.2408)	0.3696(0.3055)
SA-LuT-Net (DMFNet based)	**0.8746(0.0864)**	**0.6459(0.2353)**	**0.3776(0.2944)**

8.3.2 Synthesis

8.3.2.1 Overview

Medical image synthesis is defined as a mapping between images of an unknown target-modality and source-modalities. Current approaches can be roughly grouped into two categories, atlas- and learning-based methods. Atlas-based methods utilize paired image atlases of source- and target-modalities to calculate atlas-to-image transformations in the source-modality, and then explore this transformation to synthesize target-modality-like images from their corresponding target-modality atlases [10, 25, 49, 50, 96]. Since most atlases are built from healthy subjects, these methods perform less satisfactorily on images with pronounced abnormalities. The second category, learning-based methods, can mitigate this as these methods directly learn a mapping from the source- to target-modality. Once a training set appropriately contains a pathology, that information could be captured by the learned model, so that abnormalities, such as brain tumors, can also be synthesized in target-modality images.

A large category of learning-based synthesis methods train a nonlinear model that maps each small source-modality patch to the voxel at the center of the patch at the same location in the target-modality [53, 119, 132]. All these patch-based methods are limited by ignoring spatial relationships among the small patches in the image and that can lead to contrast inconsistency in the synthesized image. Global spatial information captured by multi-resolution patch regression [60] can alleviate this for cross-modality image synthesis. However, handcrafted features [53, 60, 119, 132] limited the descriptive power to represent complicated contextual details in images and affected synthesis quality. Patch-based estimation is usually applied to each individual voxel and the final estimation of a whole image is determined by a large number of highly overlapped patches resulting in over-smoothed synthesized images and heavy computational costs.

Deep learning based models, especially CNNs, have been used to automatically learn features with better descriptive power [46, 80]. This section mainly focuses on discussing CNN-based synthesis of three important medical images, CT, PET, and MRI. The applications of medical image synthesis can be roughly categorized into two classes, within- and cross-modality synthesis. Within-modality synthesis targets prediction of higher quality images from source images with lower quality. It includes synthesis of full-dose images from low-dose CT images [22, 23, 66], 7T MR images from 3T images [20, 89, 137], and full-dose PET images from low-dose images [129]. Cross-modality synthesis aims to transform visual information from the input source-modality to generate the target-modality images. It usually consists of a synthesis between MR and CT images [17, 37, 45, 47, 70, 77, 88, 89, 142], CT and PET images [2, 5, 6], MR and PET images [27, 125], and various MR modalities (T1, T2, FLAIR, and MRA) [18, 31, 90, 112, 126, 131]. Despite their different applications, they share the same technical basis to build a mapping from the source to the target image.

8.3.2.2 Example: 3D Conditional Generative Adversarial Networks for High-Quality PET Image Estimation at Low Dose

Medical images are usually stored in a 3D form. Most GAN methods split the 3D source images into axial slices and map these slices to 2D target slices [29, 44, 99, 127, 133], which are concatenated to form a 3D image. This leads to a loss of contextual information along the sagittal and coronal directions and strong discontinuities in the final image. A 3D-based architecture was proposed to mitigate this problem. [120] will be used as an example to describe a 3D cGAN method targeted at the synthesis of high-quality positron emission tomography (PET). PET is widely used to visualize metabolic processes in clinical diagnosis and research, and clear images are needed. A full dose of a radioactive tracer attached to a biologically active molecule is injected into a patient and pairs of gamma rays emitted from the tracer can be detected by the scanner and constructed into 3D PET images. The risk of radiation exposure from the tracer is accumulative and raises concerns about potential hazards for, especially pediatric, patients who need to take multiple PET scans. Lowering the injected dose to a half dose, causes poor image quality. Comparison of full- (F-PET) and low-dose (L-PET) PET images shows less noise in F-PET images (Fig. 8.3). Maintaining the high quality of PET images while reducing radiation doses is important.

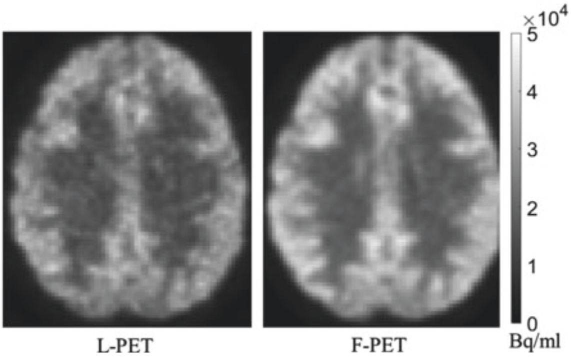


Fig. 8.3 Visual comparison between a low-dose PET (L-PET) image and the corresponding full-dose PET (F-PET) image. Image courtesy of [121]

The overall framework of the 3D cGAN proposed in [120] is illustrated in Fig. 8.4. It is comprised of two sub-modules: a 3D U-net-like generator G and a 3D CNN-based discriminator D. The entire network of G consists of 12 convolutional layers, 6 down- and 6 up-convolutional layers. The generator, G, drops pooling layers as they are not suitable for voxel-wise image quality enhancement. G learns underlying information from an L-PET image and generates an F-PET image to resemble the real target F-PET modality. The discriminator D differentiates the synthetic F-PET-like

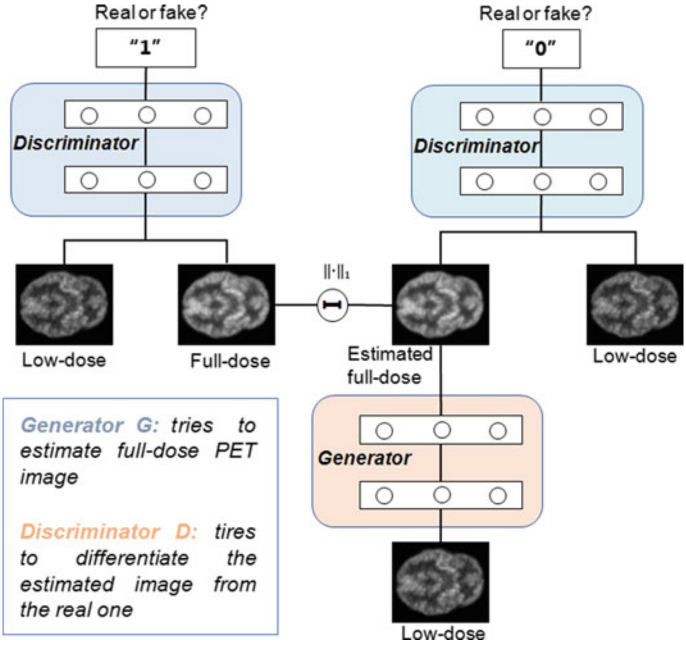

Real or fake?
"1"
Discriminator
Real or fake?
"0"
Discriminator
Low-dose
Full-dose
‖·‖₁
Estimated full-dose
Low-dose
Generator G: tries to estimate full-dose PET image
Discriminator D: tires to differentiate the estimated image from the real one
Generator
Low-dose

Fig. 8.4 Framework for training a 3D cGAN to estimate a full-dose PET image from a low-dose counterpart. Image courtesy of [120]

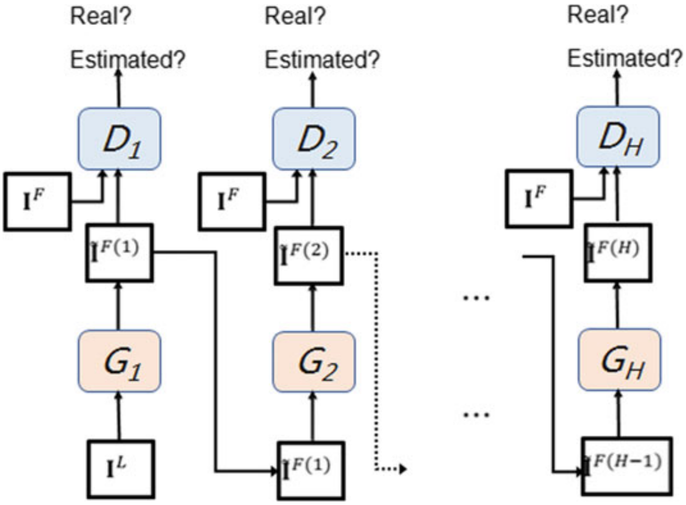


Fig. 8.5 Progressive refinement based on the concatenated 3D c-GANs. Image courtesy of [120]

image pair from the real one. G and D are trained alternatively with the optimization of adversarial and estimation error (L1) loss.

Due to possible large differences in feature distributions, a concatenated progressive refinement scheme is used. Denote I^L as L-PET images, I^F as real F-PET images, and $\tilde{I}^{F(i)}$ as the synthetic F-PET-like images from the ith layer refinement cGAN (Fig. 8.5). Estimation results from the previous 3D cGAN will be fed into the next 3D cGAN architecture to produce a new set of estimations.

The proposed 3D cGAN was evaluated on a real human brain dataset of 8 normal subjects and 8 subjects diagnosed with mild cognitive impairment (MCI) [120]. All the experiments were conducted with a “Leave-One-Subject-Out cross-validation” strategy. Considering the limited number of training images, large image patches of size 64 × 64 × 64 were extracted from each PET image of size 128 × 128 × 128 with a stride of 16, producing 125 patches for each training image. Peak signal-to-noise ratio (PSNR) and normalized mean squared error (NMSE) were utilized to measure the PET synthesis performance. Three 2D cGANs were separately trained with 2D slices from the corresponding axial, coronal, and sagittal views and were compared with our proposed 3D cGAN model (Fig. 8.6). The images estimated by all the cGANs show better quality than low-dose images, with reduced noise and enhanced image quality. Images generated by the proposed 3D cGAN (blue block, Fig. 8.6) exhibit better visual quality in all three views. Results from the 2D cGANs show good performance in the trained views indicated by red circles, but were distorted in the other two views. 2D cGANs cause discontinuous estimation and lose 3D structural information by considering the image slice by slice. The 3D cGAN model yielded the best PSNR and NMSE values on both normal and MCI subjects (Fig. 8.7). In [120], a concatenated architecture with two 3D cGANs was also constructed to compare

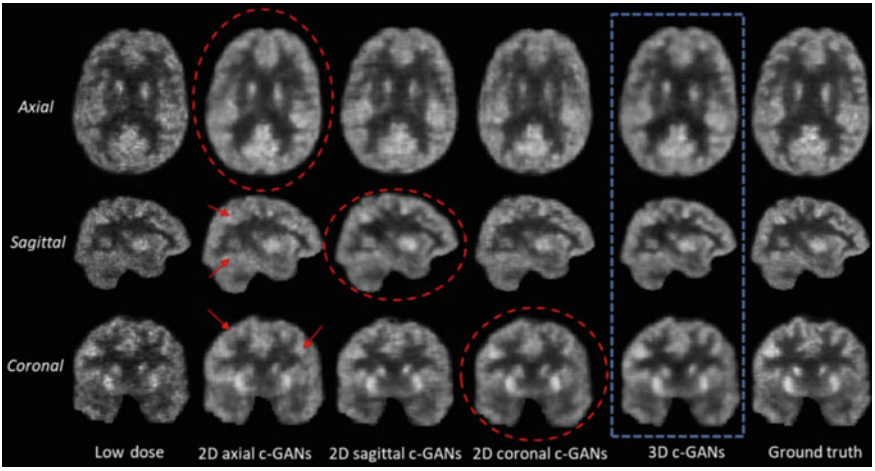


Fig. 8.6 Qualitative comparison between 2D and 3D cGANs. In the axial and coronal images, the left side of the image is the right side of the brain, and the right side of the image is the left side of the brain. Image courtesy of [120]

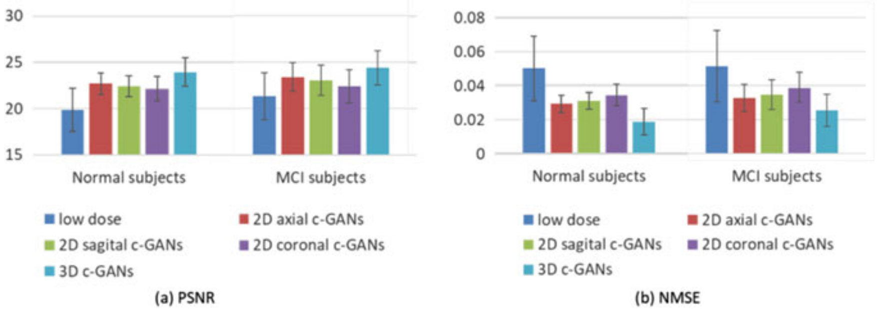


Fig. 8.7 Quantitative comparison between 2D and 3D cGAN, in terms of PSNR and NMSE. Error bars indicate the standard deviation. Image courtesy of [120]

progressive refinement against a single 3D cGAN architecture. It shows that the concatenated 3D cGANs could achieve a higher PSNR for both normal and MCI subjects.

Three state-of-the-art PET estimation methods were compared with the proposed 3D cGAN model: (1) mapping-based sparse representation (m-SR) [118], (2) semi-supervised tripled dictionary learning method (t-DL) [117], and (3) common CNN-based method [129]. 3D cGAN performed best among all methods as judged by PSNR and NMSE (Fig. 8.8).

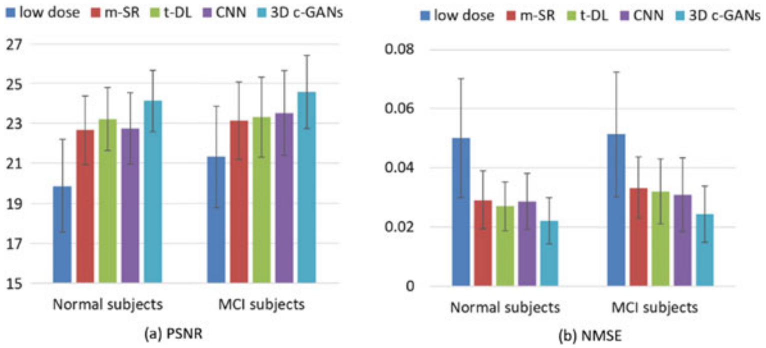


Fig. 8.8 Quantitative comparison between existing PET estimation methods and the proposed method, in terms of PSNR and NMSE. Error bars indicate the standard deviation. Image courtesy of [120]

8.3.2.3 Example: Edge-Aware Generative Adversarial Networks for Cross-Modality MR Image Synthesis [134]

Most cGAN models, like 3D cGAN [120], focus on pixel-to-pixel/voxel-to-voxel image synthesis and usually enforce pixel/voxel-wise intensity similarity between the synthesized and real images. However, they ignore structural details, such as texture in a MR image [11]. Since edges reflect local intensity changes and show the boundaries between different tissues in a MR image, maintaining edges could capture the textural structure and help sharpen the synthesized MR images. When lesions are contained in MR images, the edge information helps differentiate lesion and normal tissues, contributing to better depicting the contours of abnormal regions, e.g., glioma tumors in brain MR images [106]. To enforce edge preservation during MR image synthesis, an extra constraint based on the similarity of the edge maps from synthesized and real images can be added [134]. Edge maps were computed using the Sobel operator due to its simplicity, and its derivative can easily be computed for back-propagation.

In this work, three Sobel filters, F_i, F_j, and F_k, were used to convolve an image A to generate three edge maps corresponding to the intensity gradients along the i, j, and k directions of images. These three edge maps were merged into a final edge map $S(A)$ by

$$S(A) = \sqrt{(F_i * A)^2 + (F_j * A)^2 + (F_k * A)^2}, \tag{8.25}$$

where $*$ means the convolution operation.

The proposed MR image synthesis framework dEa-GAN (Fig. 8.9) consists of three modules, a generator G, a discriminator D, and a Sobel edge detector S. For cross-modality MR image synthesis, a source-modality image $\mathbf{x} \sim p_{data}(\mathbf{x})$ and a target-modality image $\mathbf{y} \sim p_{data}(\mathbf{y})$ were scanned on the same subject with different

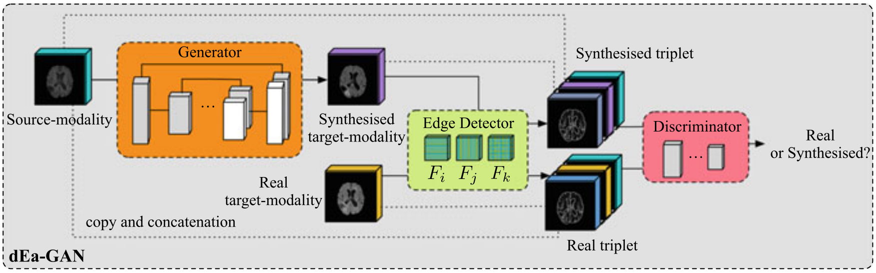


Fig. 8.9 The framework of dEa-GAN includes a generator G, a discriminator D, and a Sobel edge detector S

contrasts. The generator G of the proposed dEa-GAN aims to synthesize target-modality-like images $G(\mathbf{x})$ that can pass its discriminator D by training with the adversarial loss. L1-norm penalties were applied through G to down-weight the dissimilarity between the real and synthesized images, and between their edge maps extracted by the Sobel edge detector S. The constraint of edge map similarity is ignored in the cGAN, as in 3D cGAN. In this way, both the voxel-wise intensity similarity and the edge similarity are enforced during the synthesis. dEa-GAN incorporates the edge maps so both the generator and discriminator benefit through adversarial learning from the synthesized image and its edge map. Thus, the discriminator utilizes the edge details to differentiate real and synthesized images, and this enforces the generator to produce better edge details for synthesis.

The generator G in the dEa-GAN model was trained using the adversarial loss, the voxel-wise intensity difference loss, and the edge difference loss for synthesis, according to

$$\begin{aligned}\mathcal{L}^{G}_{dEa-GAN} = \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[\log(1 - D(\mathbf{x}, G(\mathbf{x}), S(G(\mathbf{x})))) + \\ \lambda_{l1}\mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_{data}(\mathbf{x},\mathbf{y})}[\|\mathbf{y} - G(\mathbf{x})\|_1] + \\ \lambda_{edge}\mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_{data}(\mathbf{x},\mathbf{y})}[\|S(\mathbf{y}) - S(G(\mathbf{x}))\|_1].\end{aligned} \tag{8.26}$$

The edge map $S(G(\mathbf{x}))$ implicitly appears in the first term of Eq. (8.26) through the output of the discriminator D.

The objective of the discriminator D becomes

$$\begin{aligned}\mathcal{L}^{D}_{dEa-GAN} = -\mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_{data}(\mathbf{x},\mathbf{y})}[\log D(\mathbf{x}, \mathbf{y}, S(\mathbf{y}))] - \\ \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[\log(1 - D(\mathbf{x}, G(\mathbf{x}), S(G(\mathbf{x})))].\end{aligned} \tag{8.27}$$

The discriminator takes a triplet as its input by adding the edge map $S(G(\mathbf{x}))$ or $S(\mathbf{y})$. For a synthesized triplet composed of $\mathbf{x}$, $G(\mathbf{x})$, and $S(G(\mathbf{x}))$, the label is zero; for a real triplet composed of $\mathbf{x}$, $\mathbf{y}$, $S(\mathbf{y})$, the label is one. The discriminator tries to differentiate these triplets.

Table 8.5 Quantitative evaluation results of the synthesised T1-to-FLAIR from the BRATS2015 dataset (mean ± standard deviation). A paired t-test was conducted between dEa-GAN and each of the other methods with a significance level of 0.05. When the improvement of dEa-GAN over a given method was statistically significant, the result for that method was underlined

Methods	PSNR	NMSE	SSIM
Replica [60]	27.17 ± 2.60	0.171 ± 0.267	0.939 ± 0.013
Multimodal [18]	27.26 ± 2.82	0.184 ± 0.284	0.950 ± 0.014
Pix2pix [56]	27.46 ± 2.55	0.144 ± 0.189	0.940 ± 0.015
3D cGAN	29.26 ± 3.21	0.119 ± 0.205	0.958 ± 0.016
gradient cGAN	29.38 ± 3.25	0.116 ± 0.204	0.960 ± 0.017
dEa-GAN	**30.11±3.22**	**0.105±0.174**	**0.963±0.016**

The final objective of the dEa-GAN model is

$$\mathcal{L}_{dEa-GAN} = \mathcal{L}^{G}_{dEa-GAN} + \mathcal{L}^{D}_{dEa-GAN}. \tag{8.28}$$

To validate the effectiveness of dEa-GAN, it was compared with five methods: (1) handcrafted feature used Replica [60], (2) common CNN-based Multimodal [18], (3) 2D cGAN based pix2pix [56], (4) 3D cGAN, and (5) gradient loss utilized gradient cGAN. They are all evaluated for synthesis of T1-to-FLAIR on the BRATS2015 dataset by fivefold cross-validation. The evaluation metrics, PSNR, NMSE, and SSIM [122] were applied separately on the synthesized whole images. dEa-GAN achieved the best performance at FLAIR synthesis among all the methods (Table 8.5). dEa-GAN produced higher quality images than its baseline 3D cGAN with statistically significant improvements on all evaluation metrics. dEa-GAN synthesizes MR images better than the compared methods by using edge maps in cGAN models.

8.4 Multi-modality Analysis

Medical imaging devices use imaging modalities, such as CT, PET, and differential contrast MRI, which may contain complementary information as they reflect the body from different perspectives. Integrating multiple modalities for analysis should demonstrate better diagnostic power than a single imaging modality. However, effective fusion of multiple imaging modalities for analysis is not straightforward. The relationships among different imaging modalities and the increase in features or model parameters, given the limited training samples, need to be considered. We introduce two multi-modality based methods, i.e., a non-deep-learning based approach that simultaneously learns a similarity metric and selects features from multi-modality imaging data for classification, and a deep learning model that applies a locality-adaptive strategy to effectively fuse multiple imaging modalities for PET image synthesis while keeping the increase of the number of model parameters low.

8.4.1 Example: A Non-deep-Learning Based Approach for Multi-modal Feature Selection

8.4.1.1 Overview

Machine learning and pattern classification methods have been widely applied for early diagnosis of AD based on a single modality of biomarkers. Examples are a novel landmark-based feature extraction method on MRI images that does not require nonlinear registration and tissue segmentation [141], an inherent structure-based multi-view learning method which utilizes structural information of MRI data [79], and fluorodeoxyglucose positron emission tomography (FDG- PET) to classify AD or MCI [28, 40, 48].

When analyzing neural images for disease diagnosis and therapy, features correspond to brain regions and effective feature selection can detect regions with brain atrophy, pathological amyloid depositions or metabolic alterations. Promising performance has been achieved by methods using multimodal data; however, training subjects are insufficient for the large feature dimensions and limit further improvement. High-dimensional feature vectors usually contain some irrelevant and redundant features and lead to overfitting. Therefore, feature selection is an important topic for multi-modal analysis of medical images.

Feature selection has been addressed by multi-task feature selection (MTFS) that selects a common subset of relevant features from each modality [138], inter-modality multi-task feature selection (IMTFS) that preserves complementary inter-modality information that can maintain the geometric structure of different modalities from the same subject [76], and a manifold regularized multi-task feature learning method (M2TFS) that preserves the information in each modality separately [7]. MTFS focuses on feature selection, without considering the underlying data structure while IMTFS and M2TFS also take account of the training subjects' relationships.

The neighbors and similarity of the original high-dimensional data are obtained separately from each individual modality. However, due to noisy and redundant features, the relationship of subjects in a high-dimensional space may not necessarily reveal the underlying data structure in the low-dimensional space after feature selection.

Therefore, similarity should be adaptive to changes in low-dimensional representations after feature selection. Similarity is a variable which can be optimized when selecting multi-modality features. A large amount of real-world high-dimensional data lies on low-dimensional manifolds embedded within a high-dimensional space [39]. Provided there is sufficient data (the manifold is well-sampled), each data point and its neighbors should lie on or close to a locally linear patch of the manifold. The local geometry of these patches can be characterized by linear coefficients that reconstruct each data point from its neighbors. Since neighborhood similarity is more reliable than similarity from farther samples, preserving local neighborhood structure can help to construct an accurate similarity matrix.

In light of this, we proposed a novel learning method, Adaptive Similarity Multimodality Feature Selection (ASMFS), which is able to capture the intrinsic similarity across data from different modalities and select the most informative features [147]. This method includes two major steps: (1) adaptive similarity learning with multimodality feature selection and (2) multimodal classification. In step one, simultaneously the sparse weight coefficient is optimized and the similarity matrix is updated. The manifold hypothesis is introduced in adaptive similarity learning and K local neighborhood similarities of every subject will be updated at one time. To better depict the inherent correlation among multiple modalities, the similarity matrix is shared by different modality data collected from the same subject. The objective function is optimized in an alternating manner. Experiments using the Alzheimer's Disease Neuroimaging Initiative (ADNI) database illustrated that the proposed method discovers disease sensitive biomarkers.

8.4.1.2 Method

The details of the ASMFS method in [147] are elaborated. First, a similarity measure is learned from both single and multi-modality data through adaptive similarity learning. Then, this similarity learning is embedded into the multi-modality feature selection framework.

Suppose that in a d-dimensional space, the data matrix of n subjects is denoted as $X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{d\times n}$. The subjects can be divided into c classes and the corresponding label vector is given as $y = [y_1, y_2, \ldots, y_n]$. The similarity matrix S of the data pairs can be constructed based on two principles: (1) if the distance $\|x_i - x_j\|_2^2$ between x_i and x_j is quite small, the similarity s_{ij} should be large, (2) if x_i and x_j belong to different classes, the similarity s_{ij} should be zero. The objective function to determine the similarities s_{ij} is

$$\begin{aligned} \min_{s_i} \quad & \sum_{j=1}^{n} \|x_i - x_j\|_2^2 s_{ij} \\ \text{s.t.} \quad & s_i^T \mathbf{1} = 1, 0 \le s_i \le 1 \\ & s_{ij} = 0, \text{if } y_i \ne y_j, \end{aligned} \tag{8.29}$$

where $s_i \in \mathbb{R}^n$ is a vector of which the j-th entry is s_{ij} and $\mathbf{1}$ denotes a column vector with all the elements as one. By solving problem (8.35), it can be found that only one subject which is the closest neighbor to x_i has the similarity $s_{ij} = 1$, while the others are 0. In other words, it is a trivial solution. Now, suppose the distance information is unavailable between subjects and the following problem is solved to estimate the similarities:

$$\begin{aligned} \min_{s_i} \quad & \sum_{j=1}^{n} s_{ij}^2 \\ \text{s.t.} \quad & s_i^T \mathbf{1} = 1, 0 \le s_i \le 1 \\ & s_{ij} = 0, \text{if } y_i \neq y_j \end{aligned} \tag{8.30}$$

The solution of $s_{ij} = 1/N$ reveals that all the subjects will become the nearest neighbors of x_i with $1/N$ probability. The problem (8.30) can be actually regarded as the prior of the nearest neighbor probability when the subject distance is unknown. Considering problems (8.35) and (8.30) jointly, we solve the following objective to obtain the similarities s_{ij}:

$$\begin{aligned} \min_{s_i} \quad & \sum_{j=1}^{n} (\|x_i - x_j\|_2^2 s_{ij} + \alpha s_{ij}^2) \\ \text{s.t.} \quad & s_i^T \mathbf{1} = 1, 0 \le s_i \le 1 \\ & s_{ij} = 0, \text{if } y_i \neq y_j. \end{aligned} \tag{8.31}$$

The second term s_{ij}^2 can be regarded as a regularization term to avoid a trivial solution and α is the regularization parameter. The problem (8.31) can be applied to calculate the similarities for each subject x_i. Consequently, we estimate the similarities for all subjects by solving the following problem:

$$\begin{aligned} \min_{s_i} \quad & \sum_{i=1}^{n} \sum_{j=1}^{n} (\|x_i - x_j\|_2^2 s_{ij} + \alpha s_{ij}^2) \\ \text{s.t.} \quad & s_i^T \mathbf{1} = 1, 0 \le s_i \le 1 \\ & s_{ij} = 0, \text{if } y_i \neq y_j. \end{aligned} \tag{8.32}$$

And the matrix $S = [s_1, s_2, \ldots, s_n]^T \in \mathbb{R}^{n\times n}$ can be treated as a similarity matrix of n data subjects. Now, we extend the above adaptive similarity learning to the multi-modality case. The multi-modality data are denoted as $X_1, X_2, \ldots, X_M$, where M is the number of modalities. The data matrix of the m-th modality is defined as $X_m = [x_1^{(m)}, x_2^{(m)}, \ldots, x_n^{(m)}]$. For all the multi-modality data, we solve the following problem to obtain the similarity matrix S:

$$\begin{aligned} \min_{s_i} \quad & \sum_{i=1}^{n} \sum_{j=1}^{n} (\sum_{m=1}^{M} \|x_i - x_j\|_2^2 s_{ij} + \alpha s_{ij}^2) \\ \text{s.t.} \quad & s_i^T \mathbf{1} = 1, 0 \le s_i \le 1 \\ & s_{ij} = 0, \text{if } y_i \neq y_j. \end{aligned} \tag{8.33}$$

Please note that different from traditional multi-modality methods which calculate the similarity for each modality separately, the similarity matrix S obtained in (8.33) is shared by different modality data. By doing so, we assume that the different modality data collected from the same subject should be generated via the same intrinsic distribution, thus the similarities of these data in diverse modalities would be identical.

Then, we embed the adaptive similarity learning into multi-modality feature selection in order to learn the optimal neighborhood similarity, thereby improving the performance of multi-modality classification by utilizing the more discriminative information.

To integrate the similarity learning problem (8.33) with multi-modality feature selection, the objective function of our proposed method is defined as

$$\begin{aligned}\min_{W,S}\quad & \sum_{m=1}^{M}\|y-w_m^T X_m\|_2^2+\lambda\|W\|_{2,1}\\ & +\mu\sum_{i=1}^{n}\sum_{j=1}^{n}(\sum_{m=1}^{M}\|x_i-x_j\|_2^2 s_{ij}+\gamma s_{ij}^2)\\ \text{s.t.}\quad & s_i^T\mathbf{1}=1, 0\le s_i\le 1\\ & s_{ij}=0, \text{if } y_i\neq y_j,\end{aligned} \tag{8.34}$$

where $W=[w_1,w_2,\ldots,w_M]\in\mathbb{R}^{d\times M}$ is the coefficient matrix, $w_m\in\mathbb{R}^d$ is the coefficient of the m-th modality. The $l_{2,1}$ norm of W is defined as $\|W\|_{2,1}=\sum_i^d\sqrt{\sum_j^M w_{ij}^2}$, which can result in a sparse row of W to achieve feature selection. ASMFS considers different modalities of subject into similarity construction. λ, μ and γ are regularization parameters which are capable of balancing the weights in (8.34). From (8.34), we can not only seize the inherent similarity shared across different modality data, but also select the most informative features.

8.4.1.3 Experimental Results

The data used in [147] are obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (www.loni.usc.edu). The ADNI was launched by a wide range of academic institutions and private corporations and the subjects were collected from approximately 200 cognitively normal older individuals to be followed for 3 years, 400 MCI patients to be followed for 3 years, and 200 early AD patients to be followed for 2 years across the United States and Canada. We use imaging data from 202 ADNI participants with corresponding baseline MRI and PET data. In particular, it includes 51 AD patients, 99 MCI patients, and 52 NC. The MCI patients were divided into 43 MCI converters (MCI-C) who have progressed to AD with 18 months and 56 MCI non-converters (MCI-NC) whose diagnoses have remained stable over 18 months.

The most discriminative brain regions are defined as those that are ranked by the cumulative regression coefficient W. Tables 8.6, 8.7, and 8.8 show the top 10

Table 8.6 Top 10 ROIs selected by the proposed method for AD versus NC

Number of ROIs	Selected ROIs
69	Hippocampal formation left
41	Precuneus left
80	Middle temporal gyrus right
84	Inferior temporal gyrus right
18	Angular gyrus right
87	Angular gyrus left
26	Precuneus right
46	Uncus left
83	Amygdala right
90	Lateral occipitotemporal gyrus left

Table 8.7 Top 10 ROIs selected by the proposed method for MCI versus NC

Number of ROIs	Selected ROIs
87	Angular gyrus left
69	Hippocampal formation left
64	Entorhinal cortex left
40	Cuneus left
83	Amygdala right
41	Precuneus left
63	Temporal pole left
92	Occipital pole left
30	Hippocampal formation right
17	Parahippocampal gyrus left

selected brain regions in the classification of AD versus NC, MCI versus NC, and MCI-C versus MCI-NC, respectively. For AD versus NC, brain regions such as the hippocampus, amygdala, precuneus, uncus, and temporal pole were found to be sensitive indicators of AD by our proposed method. [41, 82] have proved that the hippocampus is responsible for short-term memory, and in the early stage of Alzheimer's disease also known as MCI, the hippocampus begins to be destroyed, which directly results in the decline of short-term memory and disorientation. The amygdala is the part of brain that is responsible for managing basic emotions such as fear and anger. Damage to the amygdala caused by AD can result in paranoia and anxiety. This finding is consistent with many studies in the literature. The proposed method can also help researchers to focus on the brain regions selected in this experiment but neglected before, to help identify more brain regions related to AD and assist in the diagnosis of AD.

Table 8.8 Top 10 ROIs selected by the proposed method for MCI-C versus MCI-NC

Number of ROIs	Selected ROIs
41	Precuneus left
61	Perirhinal cortex left
35	Anterior limb of internal capsule left
48	Middle temporal gyrus left
10	Superior frontal gyrus right
83	Amygdala right
49	Lingual gyrus left
86	Middle occipital gyrus left
30	Hippocampal formation right
24	Fornix left

8.4.2 Example: A Deep Learning Based Approach for Multi-modality Fusion

8.4.2.1 Overview

Multi-modality fusion refers to the process of merging multiple images from single or multiple imaging modalities. Generally speaking, different modalities exhibit unique pathological features. The addition of other modalities can provide complementary information to the current one during the synthesis, theoretically boosting the quality of the synthesized result. Recent works also report that utilization of various modalities with functional or morphological information, like PET/CT or PET/MRI, enhances the quality of medical images [12, 52]. Researchers [62, 90, 143] have made efforts to synthesize the desired target modality by taking into account multiple modalities with regard to the same anatomical structure. A random forest regression based method, REPLICA [61], was proposed for multi-modal MRI synthesis as a representative of machine learning methods. Deep learning, CNN-based architectures, like MILR [19], have emerged due to their automatic learning of complex features. GAN-based approaches [6, 99], which can preserve high-frequency details, improve the performance of image synthesis with the help of multi-modality input. However, a critical problem of multi-modality fusion is how to combine these modality images effectively. A majority of the methods mentioned before directly stack different modalities or fuse them by employing a maximum or average operation, which inevitably causes the loss of some important information. To tackle this issue, a 3D locality adaptive multi-modality GAN, named as LA-GAN, was constructed in [121]. In the next section, we will take LA-GAN as an example and describe its performance at multi-modality fusion.

Acquiring a high-quality PET image needs a full dose of radioactive tracer injected into the human body, raising the risk of radiation exposure. Scanning MRI is radiation-free for patients. Thus, it is reasonable and beneficial to employ medical images of T1-weighted (T1-MRI), fractional anisotropy diffusion tensor imaging (FA-DTI), and mean diffusivity DTI (MD-DTI) to assist the synthesis of high-quality F-PET from L-PET images. Traditionally, for convenient processing, the same convolution filter was applied to all image locations in each input modality. Multiple input images were regarded as a multi-channel image in a global manner, which neglected the varying local contributions from the different modalities. To this end, the LA-GAN was proposed for the synthesis of F-PET from L-PET with accompanying T1-MRI, FA-DTI, and MD-DTI.

8.4.2.2 Framework

The proposed LA-GAN is composed of three modules: (1) the locality-adaptive fusion (LAF) network, (2) the generator network, and (3) the discriminator network, as presented in Fig. 8.10. The structures of the generator and the discriminator are similar to those of the 3D cGAN in Sect. 4.2. The newly incorporated LAF network takes an L-PET image and corresponding T1-MRI image, an FA-DTI image and an MD-DTI image as input, and processes them as a multi-channel image to produce a fused modality by applying different convolution kernels across different image locations. The received four modality images are divided into N non-overlapping small patches in the same way, each of which is represented by a different color in Fig. 8.10. The patches from the four input modalities are denoted in turn as

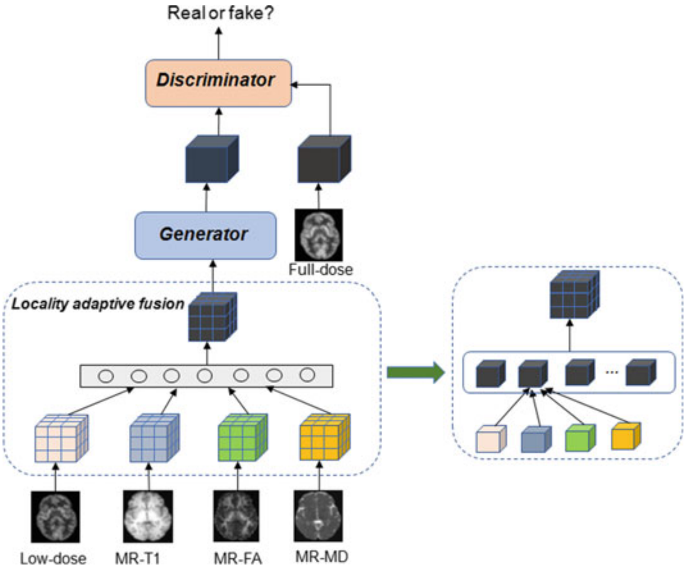


Fig. 8.10 Overview of locality adaptive multi-modality GANs

P_i^L,P_i^{T1},P_i^{FA},P_i^{MD} $(i = 1, 2, \ldots, N)$. Then, the patches at the same location from the four modalities, indicated by the same color, are convolved with different 1 × 1 × 1 kernels with parameters w_i^L, w_i^{T1}, w_i^{FA}, w_i^{MD}, respectively. Formally, the combined patch P_i^C can be obtained through the LAF module as

$$\begin{aligned} P_i^C = w_i^L P_i^L + w_i^{T1} P_i^{T1} + w_i^{FA} P_i^{FA} + w_i^{MD} P_i^{MD}, \\ s.t. w_i^L + w_i^{T1} + w_i^{FA} w_i^{MD} = 1, \\ w_i^L, w_i^{T1}, w_i^{FA}, w_i^{MD} > 0, i = 1, \ldots, N. \end{aligned} \tag{8.35}$$

In this way, N groups of different convolution kernels for $N \times 4$ patches at N local regions can be learned. This end-to-end architecture can reduce the increase of parameters under a limited number of training samples. The fused image will be fed into the generator via the locality-adaptive fusion module to synthesize a realistic F-PET image, which will be optimized by competition between the generator and the discriminator.

8.4.2.3 Experimental Results

To test the effectiveness of the proposed LA-GAN, a phantom dataset consisting of only L-PET, T1-MRI, and F-PET was utilized. Each image volume of size 128 × 128 × 128 was extracted to 125 3D image patches of size 64 × 64 × 64. To train the LAF network, each 3D image patch was further partitioned into non-overlapping 8 × 8 × 8 regions for fusion. All experiments were performed with a "Leave-One-Subject-Out" strategy and measured by PSNR and the structural similarity index measurement (SSIM). Figure 8.11 shows three examples of the synthesized F-PET by LA-GAN (the middle column), with the corresponding L-PET and F-PET images, left and right, respectively. The PSNR is given under each image. A significant improvement over L-PET was achieved in the synthetic results by LA-GAN. To verify the locality-varying contributions of different modalities, the color coded weights for different regions in different modalities are shown in Fig. 8.12. We also evaluated LA-GAN on a clinical dataset that had 8 normal control (NC) and 8 MCI subjects, each with an L-PET image, an F-PET image, a T1-MRI image, an FA-DTI image, and an MD-DTI image. The strategies used for the clinical dataset were consistent with those for the phantom data. To study the contribution of the newly proposed LAF module, the common multi-channel GAN was compared with LA-GAN. LA-GAN gave fewer artifacts and clearer details (red rectangles), as shown in Fig. 8.13. PSNR and SSIM for these methods are displayed in Table 8.9. LA-GAN had the highest PSNR and SSIM values, indicating its superiority over the common multi-channel processing method. Its improvement against the multi-channel method was statistically significant (p = 0.0482 for NC subjects and p = 0.0161 for MCI subjects, paired t-test, significant at $p < 0.05$). The qualitative and quantitative results demonstrated the effectiveness of the LAF network for the full-dose PET synthesis task. To study the contributions of the multimodal MRI images for F-PET synthesis against

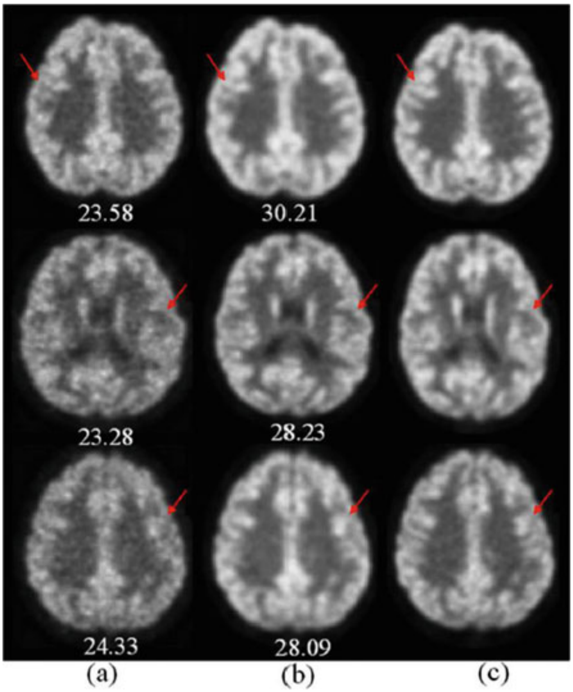


Fig. 8.11 Examples of the synthetic F-PET image by our auto-context LA-GANs method from three subjects: **a** L-PET **b** Synthesized F-PET **c** real F-PET. The values under the images denote the PSNR of the corresponding image. Image courtesy of [121]

single modality images, we used (1) MRI images (T1+DTI), (2) L-PET image, (3) L-PET+T1images, and (4) L-PET+T1+DTI images, for synthesis. The group just receiving L-PET for F-PET synthesis exactly followed the single modality GAN in [120]. Quantitative comparison results for PSNR and SSIM are given in Fig. 8.14. The lower measurement values for MRI are due to the difference in imaging mechanism between PET and MRI. Compared with the single modality of L-PET, the two modalities (L-PET and T1) achieved a better result, which was further improved when DTI images were incorporated. The proposed multi-modality LA-GAN statistically significantly outperformed (paired t-test, significant at $p < 0.05$) the single modality model in [120], with p-values of 0.0089 for NC subjects and 0.0035 for MCI subjects.

Four state-of-the-art approaches: (1) mapping-based sparse representation method (m-SR) [118], (2) tripled dictionary learning method (t-DL) [117], (3) multi-level CCA method (m-CCA) [1], and (4) auto-context CNN method (auto-CNN) [129] were compared with LA-GAN to validate its advances. The quantitative evaluation (Fig. 8.15) shows that LA-GAN had the highest values of PSNR and SSIM, the best performance among the methods.

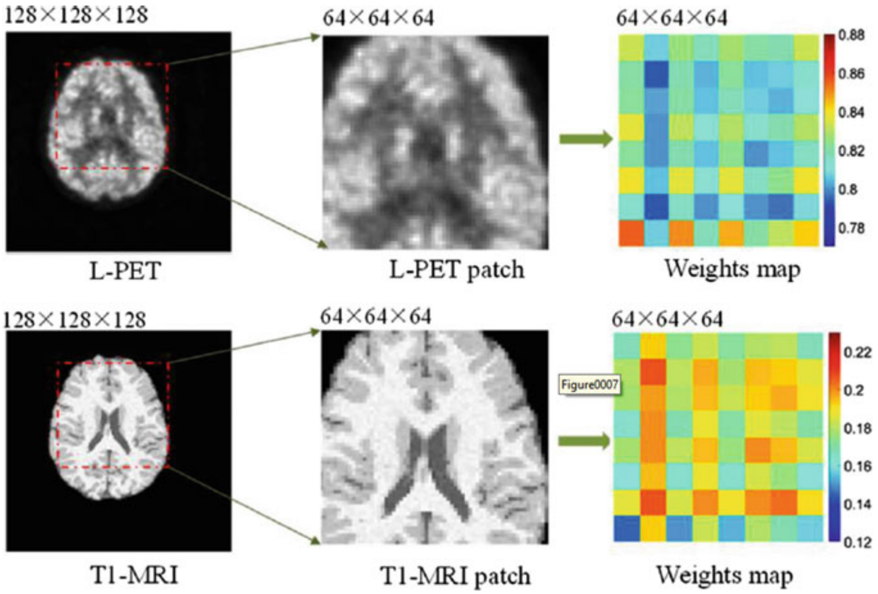


Fig. 8.12 Locality adaptive weights of different regions in different modalities. Image courtesy to [121]

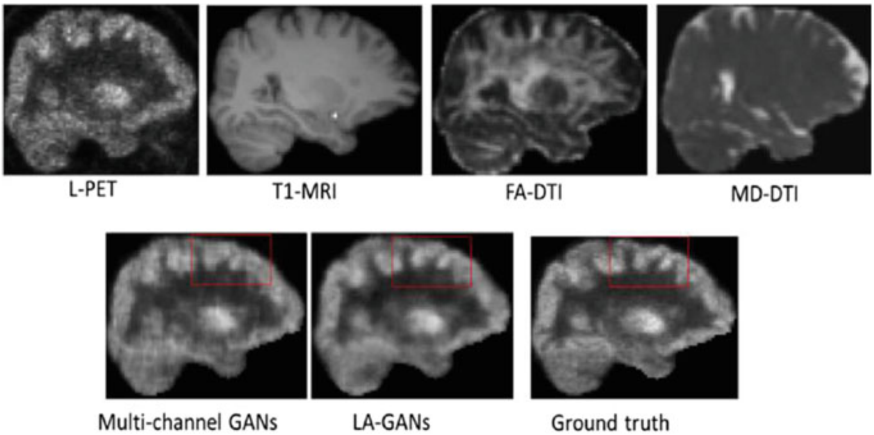


Fig. 8.13 Visual comparison with multi-channel GANs method. Image courtesy of [121]

Table 8.9 Quantitative comparison with the multi-channel GANs method on normal and MCI subjects. Mean (standard deviation), Median

	Methods	PSNR	SSIM
		Mean (std.) Med.	Mean (std.) Med.
Normal Subjects	L-PET	19.88 (2.34) 20.68	0.9790 (0.0074) 0.980
	Multi-channel	24.36 (1.93) 24.78	0.9810 (0.0065) 0.983
	LA-GAN	**24.61 (1.79) 25.32**	**0.9860 (0.0053) 0.987**
MCI Subjects	L-PET	21.33 (2.53) 21.62	0.9760 (0.0102) 0.979
	Multi-channel	24.99 (2.03) 25.36	0.9795 (0.0098) 0.982
	LA-GAN	**25.19 (1.98) 25.54**	**0.9843 (0.0097) 0.988**

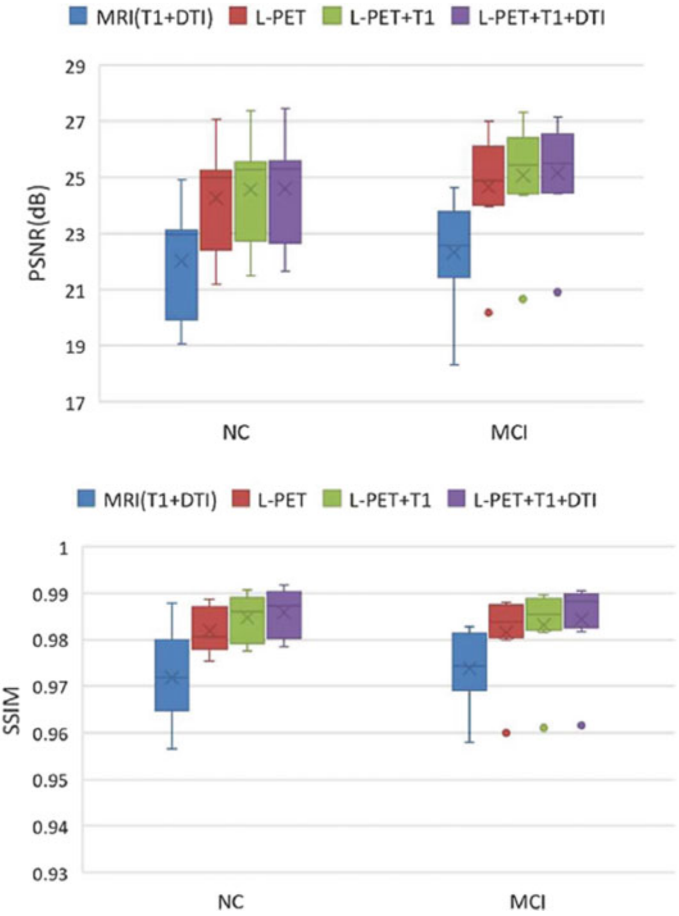


Fig. 8.14 Comparison results of our LA-GANs model using different modalities in terms of PSNR and SSIM. Image courtesy of [121]

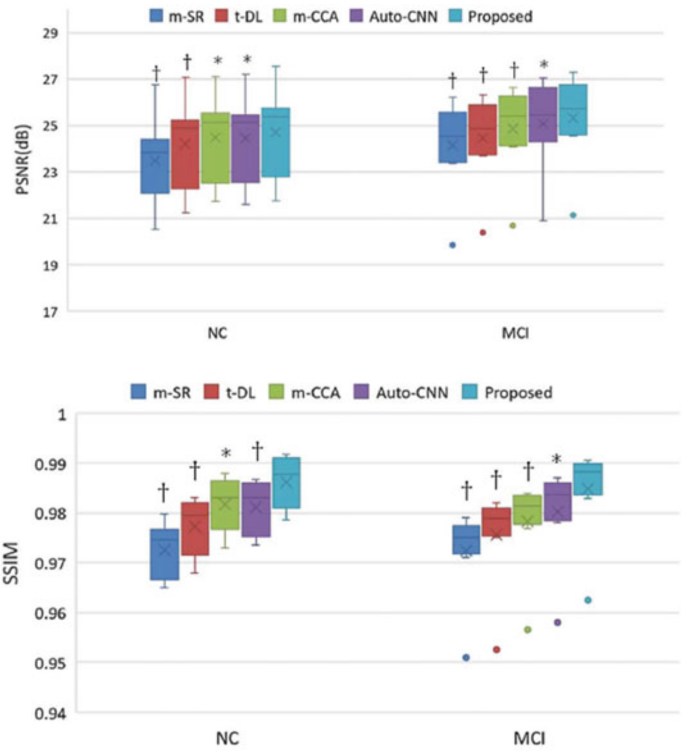


Fig. 8.15 Qualitative comparison with the state-of-the-art PET estimation methods in terms of PSNR and SSIM. † indicates $p < 0.01$ in the t-test while * means $p < 0.05$. Image courtesy of [121]

8.5 Conclusion

Machine learning methods have been widely used in computer-assisted medical image analysis. In this chapter, we briefly reviewed representative machine learning methods for classification, segmentation, synthesis, and multi-modality fusion, which are used in medical imaging applications. Due to the challenges faced by existing methods, we presented several advanced machine learning models from our work that improve performance. We demonstrated the development of our models and verified their effectiveness with experimental results. Our proposed methods are a general methodology that could be applied to medical image analysis beyond those presented in this chapter.

References

1. An L, Zhang P, Adeli-M E, Wang Y, Ma G, Shi F, Lalush D, Lin W, Shen D (2016) Multi-level canonical correlation analysis for standard-dose PET image estimation. In: IEEE Transactions on Image Processing pp 3303–3315
2. Armanious K, Jiang C, Fischer M, Küstner T, Nikolaou K, Gatidis S, Yang B (2018) MedGAN: Medical image translation using GANs. arXiv:180606397
3. Arsigny V, Fillard P, Pennec X, Ayache N (2006) Log-euclidean metrics for fast and simple calculus on diffusion tensors. Magn Reson Med 56(2):411–421. https://doi.org/10.1002/mrm.20965
4. Bakas S, Akbari H, Sotiras A, Bilello M, Rozycki M, Kirby JS, Freymann JB, Farahani K, Davatzikos C (2017) Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features. Sci. Data 4:170117
5. Ben-Cohen A, Klang E, Raskin SP, Amitai MM, Greenspan H (2017) Virtual PET images from CT data using deep convolutional networks: initial results. In: International workshop on simulation and synthesis in medical imaging. Springer, pp 49–57
6. Bi L, Kim J, Kumar A, Feng D, Fulham M (2017) Synthesis of positron emission tomography (PET) images via multi-channel generative adversarial networks (GANs). In: Molecular imaging, reconstruction and analysis of moving body organs, and stroke imaging and treatment. Springer, pp 43–51
7. Biao J, Daoqiang Z, Bo C, Dinggang S, null n (2015) Manifold regularized multitask feature learning for multimodality disease classification. In: Human brain mapping, pp 489–507
8. Bishop CM (2006) Machine learning and pattern recognition. Information science and statistics. Springer, Heidelberg
9. de Brebisson A, Montana G (2015) Deep neural networks for anatomical brain segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 20–28
10. Burgos N, Cardoso MJ, Thielemans K, Modat M, Pedemonte S, Dickson J, Barnes A, Ahmed R, Mahoney CJ, Schott JM et al (2014) Attenuation correction synthesis for hybrid PET-MR scanners: application to brain studies. IEEE Trans Med Imaging 33(12):2332–2341
11. Bustin A, Voilliot D, Menini A, Felblinger J, de Chillou C, Burschka D, Bonnemains L, Odille F (2018) Isotropic reconstruction of mr images using 3D patch-based self-similarity learning. In: IEEE transactions on medical imaging
12. Cao X, Gao Y, Yang J, Wu G, Shen D (2016) Learning-based multimodal image registration for prostate cancer radiation therapy. In: International conference on medical image computing and computer-assisted intervention
13. de Celis Alonso B, Tobón SH, Suarez PD, Flores JG, de Celis Carrillo B, Pérez EB (2014) A multi-methodological mr resting state network analysis to assess the changes in brain physiology of children with ADHD. PLoS ONE 6. https://doi.org/10.1371/journal.pone.0099119
14. Chandrakumar T, Kathirvel R (2016) Classifying diabetic retinopathy using deep learning architecture. In: International journal of engineering research & technology (IJERT)
15. Chao-Gan Y, Yu-Feng Z (2010) DPARSF: a matlab toolbox for “pipeline” data analysis of resting-state fMRI. Front Syst Neurosc 4
16. Chapelle O, Vapnik V, Bousquet O, Mukherjee S (2002) Choosing multiple parameters for support vector machines. Mach Learn 46(1–3):131–159
17. Chartsias A, Joyce T, Dharmakumar R, Tsaftaris SA (2017) Adversarial image synthesis for unpaired multi-modal cardiac data. In: International workshop on simulation and synthesis in medical imaging. Springer, pp 3–13
18. Chartsias A, Joyce T, Giuffrida MV, Tsaftaris SA (2018a) Multimodal MR synthesis via modality-invariant latent representation. IEEE Trans Med Imaging 37(3):803–814
19. Chartsias A, Joyce T, Giuffrida MV, Tsaftaris SA (2018b) Multimodal mr synthesis via modality-invariant latent representation. In: IEEE transactions on medical imaging, pp 1–1

20. Chaudhari AS, Fang Z, Kogan F, Wood J, Stevens KJ, Gibbons EK, Lee JH, Gold GE, Hargreaves BA (2018) Super-resolution musculoskeletal mri using deep learning. Magn Reson Med 80(5):2139–2154
21. Chen C, Liu X, Ding M, Zheng J, Li J (2019) 3D dilated multi-fiber network for real-time brain tumor segmentation in MRI. In: International conference on medical image computing and computer-assisted intervention. Springer, pp 184–192
22. Chen H, Zhang Y, Kalra MK, Lin F, Chen Y, Liao P, Zhou J, Wang G (2017a) Low-dose ct with a residual encoder-decoder convolutional neural network. IEEE Trans Med Imaging 36(12):2524–2535
23. Chen H, Zhang Y, Zhang W, Liao P, Li K, Zhou J, Wang G (2017b) Low-dose CT via convolutional neural network. Biomed Opt Express 8(2):679–694
24. Chen H, Dou Q, Yu L, Qin J, Heng PA (2018a) Voxresnet: deep voxelwise residual networks for brain segmentation from 3D MR images. NeuroImage 170:446–455
25. Chen M, Jog A, Carass A, Prince JL (2015) Using image synthesis for multi-channel registration of different image modalities. In: Medical Imaging 2015: Image Processing, International Society for Optics and Photonics, vol 9413, p 94131Q
26. Chen W, Liu B, Peng S, Sun J, Qiao X (2018b) S3D-UNet: separable 3D U-Net for brain tumor segmentation. In: International MICCAI brainlesion workshop. Springer, pp 358–368
27. Choi H, Lee DS (2018) Generation of structural mr images from amyloid PET: application to MR-less quantification. J Nucl Med 59(7):1111–1117
28. Chételat G, Desgranges B, Sayette VDL, Viader F, Baron JC (2003) Mild cognitive impairment: can FDG-PET predict who is to rapidly convert to alzheimer's disease? Neurology 60(8):1374–7
29. Costa P, Galdran A, Meyer M, Niemeijer M, Abramoff M, Mendonça A, Campilho A (2017) End-to-end adversarial retinal image synthesis. In: IEEE transactions on medical imaging
30. Cristianini N, Shawe-Taylor J, Elisseeff A, Kandola JS (2001) On kernel target alignment. In: Proceedings 14th neural information processing systems, British Columbia, Canada, pp 367–373
31. Dar SUH, Yurt M, Karacan L, Erdem A, Erdem E, Çukur T (2018) Image synthesis in multi-contrast MRI with conditional generative adversarial networks. arXiv:180201221
32. Deshpande G, Wang P, Rangaprakash D, Wilamowski B (2015) Fully connected cascade artificial neural network architecture for attention deficit hyperactivity disorder classification from functional magnetic resonance imaging data. IEEE Trans Cybern PP(99):1–1. https://doi.org/10.1109/TCYB.2014.2379621
33. Dolz J, Gopinath K, Yuan J, Lombaert H, Desrosiers C, Ayed IB (2018) Hyperdense-net: a hyper-densely connected CNN for multi-modal image segmentation. IEEE Trans Med Imaging 38(5):1116–1126
34. Dou Q, Chen H, Jin Y, Yu L, Qin J, Heng PA (2016) 3D deeply supervised network for automatic liver segmentation from CT volumes. In: International conference on medical image computing and computer-assisted intervention, Springer, pp 149–157
35. Dryden IL, Koloydenko A, Zhou D (2009) Non-euclidean statistics for covariance matrices, with applications to diffusion tensor imaging. In: The annals of applied statistics, pp 1102–1123
36. Elgendi M, Nasir MU, Tang Q, Fletcher RR, Howard N, Menon C, Ward R, Parker W, Nicolaou S (2020) The performance of deep neural networks in differentiating chest x-rays of covid-19 patients from other bacterial and viral pneumonias. In: Frontiers in medicine
37. Emami H, Dong M, Nejad-Davarani SP, Glide-Hurst CK (2018) Generating synthetic CTs from magnetic resonance images using generative adversarial networks. Med Phys 45(8):3627–3636
38. Farag A, Lu L, Roth HR, Liu J, Turkbey E, Summers RM (2016) A bottom-up approach for pancreas segmentation using cascaded superpixels and (deep) image patch labeling. IEEE Trans Image Process 26(1):386–399
39. Fefferman C, mitter ks, Narayanan H (2016) Testing the manifold hypothesis. In: Journal of the American mathematical society, pp 983–1049

40. Foster NL, Heidebrink JL, Clark CM, Jagust WJ, Arnold SE, Barbas NR, Decarli CS, Scott Turner R, Koeppe RA, Ra Higdon (2007) FDG-PET improves accuracy in distinguishing frontotemporal dementia and alzheimer's̈ disease. Brain 130(10):2616–2635
41. Fox CN, Schott MJ (2004) Imaging cerebral atrophy: normal ageing to alzheimer's disease. In: Lancet (London, England), pp 392–394
42. Gao Z, Wang L, Zhou L, Zhang J (2016) Hep-2 cell image classification with deep convolutional neural networks. IEEE J Biomed Health Inform 21(2):416–428
43. Gulshan V, Peng L, Coram M, Stumpe MC, Wu D, Narayanaswamy A, Venugopalan S, Widner K, Madams T, Cuadros J, Kim R, Raman R, Nelson PC, Mega JL, Webster DR (2016) Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. J Am Med Assoc (JAMA) 316(22):2402–2410
44. Han C, Hayashi H, Rundo L, Araki R, Shimoda W, Muramatsu S, Furukawa Y, Mauri G, Nakayama H (2018) Gan-based synthetic brain mr image generation. In: 2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)
45. Han X (2017) MR-based synthetic CT generation using a deep convolutional neural network method. Med Phys 44(4):1408–1419
46. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
47. Hiasa Y, Otake Y, Takao M, Matsuoka T, Takashima K, Carass A, Prince JL, Sugano N, Sato Y (2018) Cross-modality image synthesis from unpaired data using cyclegan. In: International workshop on simulation and synthesis in medical imaging, Springer, pp 31–41
48. Higdon R, Foster NL, Koeppe RA, Decarli CS, Minoshima S (2010) A comparison of classification methods for differentiating fronto-temporal dementia from alzheimer's disease using FDG-PET imaging. Stat Med 23(2):315–326
49. Hofmann M, Steinke F, Scheel V, Charpiat G, Farquhar J, Aschoff P, Brady M, Scholkopf B (1875) Pichler BJ (2008) MRI-based attenuation correction for PET/MRI: a novel approach combining pattern recognition and atlas registration. J Nucl Med 49(11)
50. Hofmann M, Bezrukov I, Mantlik F, Aschoff P, Steinke F, Beyer T, Pichler BJ, Schölkopf B (2011) MRI-based attenuation correction for whole-body PET/MRI: quantitative evaluation of segmentation-and atlas-based methods. J Nucl Med 52(9):1392–1399
51. Huang S, Li J, Sun L, Ye J, Fleisher A, Wu T, Chen K, Reiman E (2010) Learning brain connectivity of alzheimer's disease by sparse inverse covariance estimation. NeuroImage 50(3):935–949
52. Huang Y, Member S, IEEE, Shao L, Member S (2017) Cross-modality image synthesis via weakly-coupled and geometry co-regularized joint dictionary learning. IEEE Trans Med Imaging PP(99):1–1
53. Huynh T, Gao Y, Kang J, Wang L, Zhang P, Lian J, Shen D (2016) Estimating CT image from MRI data using structured random forest and auto-context model. IEEE Trans Med Imaging 35(1):174–183
54. Isensee F, Kickingereder P, Wick W, Bendszus M, Maier-Hein KH (2017) Brain tumor segmentation and radiomics survival prediction: contribution to the brats 2017 challenge. In: International MICCAI brainlesion workshop, Springer, pp 287–297
55. Isensee F, Kickingereder P, Wick W, Bendszus M, Maier-Hein KH (2018) No new-net. In: International MICCAI brainlesion workshop, Springer, pp 234–244
56. Isola P, Zhu JY, Zhou T, Efros AA (2016) Image-to-image translation with conditional adversarial networks. arXiv:161107004
57. Jaworek-Korjakowska J, Kleczek P, Gorgon M (2019) Melanoma thickness prediction based on convolutional neural network with vgg-19 model transfer learning. In: IEEE conference on computer vision and pattern recognition workshop (CVPRW)
58. Jayasumana S, Hartley R, Salzmann M, Li H, Harandi M (2013) Kernel methods on the riemannian manifold of symmetric positive definite matrices. In: IEEE computer society conference on computer vision and pattern recognition
59. Jia H, Xia Y, Song Y, Zhang D, Huang H, Zhang Y, Cai W (2019) 3D APA-Net: 3D adversarial pyramid anisotropic convolutional network for prostate segmentation in MR images. IN: IEEE transactions on medical imaging

60. Jog A, Carass A, Roy S, Pham DL, Prince JL (2017a) Random forest regression for magnetic resonance image synthesis. Med Image Anal 35:475–488
61. Jog A, Carass A, Roy S, Pham DL, Prince JL (2017b) Random forest regression for magnetic resonance image synthesis. Med Image Anal 35:475–488
62. Joyce T, Chartsias A, Tsaftaris SA (2017) Robust multi-modal MR image synthesis. In: International conference on medical image computing and computer-assisted intervention, Springer, pp 347–355
63. Kaiser M (2011) A tutorial in connectome analysis: topological and spatial features of brain networks. Neuroimage 57(3):892–907
64. Kamnitsas K, Bai W, Ferrante E, McDonagh S, Sinclair M, Pawlowski N, Rajchl M, Lee M, Kainz B, Rueckert D et al (2017a) Ensembles of multiple models and architectures for robust brain tumour segmentation. In: International MICCAI brainlesion workshop, Springer, pp 450–462
65. Kamnitsas K, Ledig C, Newcombe VF, Simpson JP, Kane AD, Menon DK, Rueckert D, Glocker B (2017b) Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. Med Image Anal 36:61–78
66. Kang E, Min J, Ye JC (2017) A deep convolutional neural network using directional wavelets for low-dose X-ray CT reconstruction. Med Phys 44(10):e360–e375
67. Keerthi SS (2002) Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms. IEEE Trans Neural Netw 13:1225–1229
68. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
69. Krizhevsky A, Sutskever I, Hinton GE (2017) Imagenet classification with deep convolutional neural networks. Commun ACM 316(22):84–90
70. Leynes AP, Yang J, Wiesinger F, Kaushik SS, Shanbhag DD, Seo Y, Hope TA, Larson PE (2017) Direct pseudo CT generation for pelvis PET/MRI attenuation correction using deep convolutional neural networks with multi-parametric mri: zero echo-time and dixon deep pseudoct (ZeDD-CT). In: Journal of nuclear medicine, pp jnumed–117
71. Li W et al (2015) Automatic segmentation of liver tumor in ct images with deep convolutional neural networks. J Comput Commun 3(11):146
72. Li X, Sroubek A, Kelly MS, Lesser I, Sussman E, He Y, Branch C, Foxe JJ (2012) Atypical pulvinar-cortical pathways during sustained attention performance in children with attention-deficit/hyperactivity disorder. J Am Acad Child Adolesc Psychiatry 51(11):1197–1207
73. Litjens G, Toth R, van de Ven W, Hoeks C, Kerkstra S, van Ginneken B, Vincent G, Guillard G, Birbeck N, Zhang J et al (2014) Evaluation of prostate segmentation algorithms for MRI: the PROMISE12 challenge. Med Image Anal 18(2):359–373
74. Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, Van Der Laak JA, Van Ginneken B, Sánchez CI (2017a) A survey on deep learning in medical image analysis. Med Image Anal 42:60–88
75. Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, van der Laak JA, van Ginneken B, Sánchez CI (2017b) A survey on deep learning in medical image analysis. Med Image Anal 42:60–88
76. Liu F, Wee CY, Chen H, Shen D (2014) Inter-modality relationship constrained multi-modality multi-task feature selection for alzheimer's disease and mild cognitive impairment identification. In: NeuroImage, pp 466–475
77. Liu F, Jang H, Kijowski R, Bradshaw T, McMillan AB (2017) Deep learning MR imaging-based attenuation correction for PET/MR imaging. Radiology 286(2):676–684
78. Liu J, Ji S, Ye J (2009) SLEP: sparse learning with efficient projections. Arizona State University. http://www.public.asu.edu/~jye02/Software/SLEP
79. Liu M, Zhang D, Adeli-Mosabbeb E, Shen D (2015) Inherent structure based multi-view learning with multi-template feature representation for alzheimer's disease diagnosis. In: IEEE transactions on biomedical engineering, pp 1–1
80. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3431–3440

81. Milletari F, Navab N, Ahmadi SA (2016) V-net: fully convolutional neural networks for volumetric medical image segmentation. In: 2016 fourth international conference on 3D vision (3DV), IEEE, pp 565–571
82. Misra C, Fan Y, Davatzikos C (2009) Baseline and longitudinal patterns of brain atrophy in mci patients, and their use in prediction of short-term conversion to ad: results from adni. NeuroImage, pp 1415–1422
83. Moeskops P, Viergever MA, Mendrik AM, De Vries LS, Benders MJ, Išgum I (2016a) Automatic segmentation of mr brain images with a convolutional neural network. IEEE Trans Med Imaging 35(5):1252–1261
84. Moeskops P, Wolterink JM, van der Velden BH, Gilhuijs KG, Leiner T, Viergever MA, Išgum I (2016b) Deep learning for multi-task medical image segmentation in multiple modalities. In: International conference on medical image computing and computer-assisted intervention, Springer, pp 478–486
85. Moeskops P, Veta M, Lafarge MW, Eppenhof KA, Pluim JP (2017) Adversarial training and dilated convolutions for brain MRI segmentation. In: Deep learning in medical image analysis and multimodal learning for clinical decision support, Springer, pp 56–64
86. Myronenko A (2018) 3D MRI brain tumor segmentation using autoencoder regularization. In: International MICCAI brainlesion workshop, Springer, pp 311–320
87. Nair T, Precup D, Arnold DL, Arbel T (2020) Exploring uncertainty measures in deep networks for multiple sclerosis lesion detection and segmentation. Med Image Anal 59:101557
88. Nie D, Cao X, Gao Y, Wang L, Shen D (2016) Estimating CT image from MRI data using 3D fully convolutional networks. In: Deep learning and data labeling for medical applications, Springer, pp 170–178
89. Nie D, Trullo R, Lian J, Wang L, Petitjean C, Ruan S, Wang Q, Shen D (2018) Medical image synthesis with deep convolutional adversarial networks. In: IEEE transactions on biomedical engineering
90. Olut S, Sahin YH, Demir U, Unal G (2018) Generative adversarial training for MRA image synthesis using multi-contrast MRI. In: International workshop on PRedictive intelligence in MEdicine, Springer, pp 147–154
91. Poudel RP, Lamata P, Montana G (2016) Recurrent fully convolutional neural networks for multi-slice MRI cardiac segmentation. In: Reconstruction, segmentation, and analysis of medical images, Springer, pp 83–94
92. Qayyum A, Anwar SM, Awais M, Majid M (2017) Medical image retrieval using deep convolutional neural network. Neurocomputing 266:8–20
93. Robinson MD, Chiu SJ, Toth CA, Izatt JA, Lo JY, Farsiu S (2010) New applications of super-resolution in medical imaging. In: Milanfar P (ed) Super-resolution imaging. CRC Press, pp 383–412
94. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention, Springer, pp 234–241
95. Roth HR, Lu L, Farag A, Shin HC, Liu J, Turkbey EB, Summers RM (2015) Deeporgan: multi-level deep convolutional networks for automated pancreas segmentation. In: International conference on medical image computing and computer-assisted intervention, Springer, pp 556–564
96. Roy S, Carass A, Prince JL (2013) Magnetic resonance image example-based contrast synthesis. IEEE Trans Med Imaging 32(12):2348–2363
97. Roy S, Butman JA, Reich DS, Calabresi PA, Pham DL (2018) Multiple sclerosis lesion segmentation from brain MRI via fully convolutional neural networks. arXiv:180309172
98. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-Fei L (2015) ImageNet large scale visual recognition challenge. Int J Comput Vis (IJCV) 115(3):211–252. https://doi.org/10.1007/s11263-015-0816-y
99. Dar SUh, Yurt M, Karacan L, Erdem A, Erdem E (2019) Image synthesis in multi-contrast mri with conditional generative adversarial networks. In: IEEE transactions on medical imaging

100. Schlemper J, Caballero J, Hajnal JV, Price AN, Rueckert D (2017) A deep cascade of convolutional neural networks for dynamic MR image reconstruction. IEEE Trans Med Imaging 37(2):491–503
101. Schölkopf B, Smola A, Müller KR (1998) Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput 10(5):1299–1319
102. Silverstein JW (1986) Eigenvalues and eigenvectors of large dimensional sample covariance matrices. Contemp Math 50:153–159
103. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:14091556
104. Singh SP, Wang L, Gupta S, Goli H, Padmanabhan P, Gulyás B (2020) 3D deep learning on medical images: a review. Sensors
105. Sra S (2012) A new metric on the manifold of kernel matrices with application to matrix geometric means. In: Pereira F, Burges C, Bottou L, Weinberger K (eds) Advances in neural information processing systems 25, Curran Associates, Inc., pp 144–152
106. Sugahara T, Korogi Y, Kochi M, Ikushima I, Shigematu Y, Hirai T, Okuda T, Liang L, Ge Y, Komohara Y et al (1999) Usefulness of diffusion-weighted MRI with echo-planar technique in the evaluation of cellularity in gliomas. J Magn Reson Imaging 9(1):53–60
107. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
108. Tajbakhsh N, Shin JY, Gurudu SR, Hurst RT, Kendall CB, Gotway MB, Liang J (2016) Convolutional neural networks for medical image analysis: full training or fine tuning? IEEE Trans Med Imaging 35(5):1299–1312
109. Thong W, Kadoury S, Piché N, Pal CJ (2018) Convolutional networks for kidney segmentation in contrast-enhanced ct scans. Comput Methods Biomech Biomed Eng Imaging Vis 6(3):277–282
110. Tran PV (2016) A fully convolutional neural network for cardiac segmentation in short-axis MRI. arXiv:160400494
111. Tzourio-Mazoyer N, Landeau B, Papathanassiou D, Crivello F, Etard O, Delcroix N, Mazoyer B, Joliot M (2002) Automated anatomical labeling of activations in spm using a macroscopic anatomical parcellation of the MNI MRI single-subject brain. Neuroimage 15(1):273–289
112. Van Nguyen H, Zhou K, Vemulapalli R (2015) Cross-domain synthesis of medical images using efficient location-sensitive deep network. In: International conference on medical image computing and computer-assisted intervention, Springer, pp 677–684
113. Vapnik V (2013) The nature of statistical learning theory. Springer science & business media
114. Varoquaux G, Baronnet F, Kleinschmidt A, Fillard P, Thirion B (2010) Detection of brain functional-connectivity difference in post-stroke patients using group-level covariance modeling. In: Medical image computing and computer-assisted intervention–MICCAI 2010, Springer, pp 200–208
115. Wang L (2008) Feature selection with kernel class separability. IEEE Trans Pattern Anal Mach Intell 30:1534–1546
116. Wang L, Xue P, Chan KL (2008) Two criteria for model selection in multiclass support vector machines. IEEE Trans Syst Man Cybern B, Cybern 38:1432–1448
117. Wang Y, Ma G, An L, Shi F, Zhang P, Wu X, Zhou J, Shen D (2016a) Semisupervised tripled dictionary learning for standard-dose PET image prediction using low-dose PET and multimodal MRI. In: IEEE transactions on biomedical engineering pp 1–1
118. Wang Y, Zhang P, An L, Ma G, Kang J, Shi F, Wu X, Zhou J, Lalush DS, Lin W et al (2016b) Predicting standard-dose PET image from low-dose PET and multimodal MR images using mapping-based sparse representation. Phys Med Biol 61(2):791
119. Wang Y, Ma G, An L, Shi F, Zhang P, Lalush DS, Wu X, Pu Y, Zhou J, Shen D (2017) Semisupervised tripled dictionary learning for standard-dose PET image prediction using low-dose PET and multimodal MRI. IEEE Trans Biomed Eng 64(3):569–579
120. Wang Y, Yu B, Wang L, Zu C, Lalush DS, Lin W, Wu X, Zhou J, Shen D, Zhou L (2018a) 3D conditional generative adversarial networks for high-quality PET image estimation at low dose. NeuroImage 174:550–562

121. Wang Y, Zhou L, Yu B, Wang L, Zu C, Lalush DS, Lin W, Wu X, Zhou J, Shen D (2018b) 3D auto-context-based locality adaptive multi-modality GANs for PET synthesis. In: IEEE transactions on medical imaging, pp 1328–1339
122. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612
123. Wee CY, Yap PT, Denny K, Browndyke JN, Potter GG, Welsh-Bohmer KA, Wang L, Shen D (2012a) Resting-state multi-spectrum functional connectivity networks for identification of MCI patients. PloS One 7(5):e37828
124. Wee CY, Yap PT, Zhang D, Wang L, Shen D (2012b) Constrained sparse functional connectivity networks for MCI classification. In: Medical image computing and computer-assisted intervention–MICCAI 2012, Springer, pp 212–219
125. Wei W, Poirion E, Bodini B, Durrleman S, Ayache N, Stankoff B, Colliot O (2018) Learning myelin content in multiple sclerosis from multimodal MRI through adversarial training. In: International conference on medical image computing and computer-assisted intervention, Springer, pp 514–522
126. Welander P, Karlsson S, Eklund A (2018) Generative adversarial networks for image-to-image translation on multi-contrast MR images-a comparison of cyclegan and unit. arXiv:180607777
127. Wolterink JM, Dinkla AM, Savenije MHF, Seevinck PR, Igum I (2017) Deep MR to CT synthesis using unpaired data. In: International workshop on simulation and synthesis in medical imaging
128. Wulczyn E, Steiner DF, Xu Z, Sadhwani A, Wang H, Flament-Auvigne I, Mermel CH, Chen PHC, Liu Y, Stumpe MC (2020) Deep learning-based survival prediction for multiple cancer types using histopathology images. PLoS One
129. Xiang L, Qiao Y, Nie D, An L, Lin W, Wang Q, Shen D (2017) Deep auto-context convolutional neural networks for standard-dose PET image estimation from low-dose PET/MRI. Neurocomputing 267:406–416
130. Xiong H, Swamy M, Ahmad MO (2005) Optimizing the kernel in the empirical feature space. IEEE Trans Neural Netw 16:460–474
131. Yang Q, Li N, Zhao Z, Fan X, Chang EI, Xu Y, et al. (2018) Mri cross-modality neuroimage-to-neuroimage translation. arXiv:180106940
132. Ye DH, Zikic D, Glocker B, Criminisi A, Konukoglu E (2013) Modality propagation: coherent synthesis of subject-specific scans with data-driven regularization. In: International conference on medical image computing and computer-assisted intervention, Springer, pp 606–613
133. Yi X, Babyn P (2018) Sharpness-aware low-dose ct denoising using conditional generative adversarial network. J Digit Imaging 31(5):655–669
134. Yu B, Zhou L, Wang L, Shi Y, Fripp J, Bourgeat P (2019) Ea-GANs: edge-aware generative adversarial networks for cross-modality MR image synthesis. In: IEEE transactions on medical imaging
135. Yu B, Zhou L, Wang L, Yang W, Yang M, Bourgeat P, Fripp J (2020) Learning sample-adaptive intensity lookup table for brain tumor segmentation. In: International conference on medical image computing and computer-assisted intervention, Springer, pp 216–226
136. Yu L, Yang X, Chen H, Qin J, Heng PA (2017) Volumetric convnets with mixed residual connections for automated prostate segmentation from 3D MR images. In: Thirty-first AAAI conference on artificial intelligence
137. Zeng K, Zheng H, Cai C, Yang Y, Zhang K, Chen Z (2018) Simultaneous single-and multi-contrast super-resolution for brain MRI images based on a convolutional neural network. Comput Biol Med 99:133–141
138. Zhang D, Wang Y, Zhou L, Yuan H, Shen D (2011) Multimodal classification of alzheimer's disease and mild cognitive impairment. NeuroImage, pp 856–867
139. Zhang J, Wang L, Zhou L, Li W (2015a) Learning discriminative stein kernel for SPD matrices and its applications. IEEE Trans Neural Netw Learn Syst 27(5):1020–1033
140. Zhang J, Zhou L, Wang L, Li W (2015b) Functional brain network classification with compact representation of SICE matrices. IEEE Trans Biomed Eng 62(6):1623–1634

141. Zhang J, Gao Y, Gao Y, Munsell BC, Shen D (2016) Detecting anatomical landmarks for fast alzheimer's disease diagnosis. IEEE Trans Med Imaging 35(12):2524–2533
142. Zhang Z, Yang L, Zheng Y (2018) Translating and segmenting multimodal medical volumes with cycle-and shape-consistency generative adversarial network. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 9242–9251
143. Zhou T, Fu H, Chen G, Shen J, Shao L (2020) Hi-net: hybrid-fusion network for multi-modal mr image synthesis. IEEE Trans Med Imaging PP(99):1–1
144. Zhou Y, Xie L, Shen W, Wang Y, Fishman EK, Yuille AL (2017) A fixed-point model for pancreas segmentation in abdominal CT scans. In: International conference on medical image computing and computer-assisted intervention, Springer, pp 693–701
145. Zhu Q, Du B, Turkbey B, Choyke PL, Yan P (2017) Deeply-supervised CNN for prostate segmentation. In: 2017 international joint conference on neural networks (IJCNN), IEEE, pp 178–184
146. Zreik M, Leiner T, De Vos BD, van Hamersvelt RW, Viergever MA, Išgum I (2016) Automatic segmentation of the left ventricle in cardiac CT angiography using convolutional neural networks. In: 2016 IEEE 13th international symposium on biomedical imaging (ISBI), IEEE, pp 40–43
147. Zu C, Wang Y, Zhou L, Wang L, Zhang D (2018) Multi-modality feature selection with adaptive similarity learning for classification of alzheimer's disease. In: IEEE international symposium on biomedical imaging (ISBI)

Chapter 9
EfficientNet-Based Convolutional Neural Networks for Tuberculosis Classification

Vinayakumar Ravi, Harini Narasimhan, and Tuan D. Pham

Abstract Tuberculosis (TB) is an infectious disease that remained as a major health threat in the world. The computer-aided diagnosis (CAD) system for TB is one of the automated methods in early diagnosis and treatment, particularly used in developing countries. Literature survey shows that many methods exist based on machine learning for TB classification using X-ray images. Recently, deep learning approaches have been used instead of machine learning in many applications. This is mainly due to the reason that deep learning can learn optimal features from the raw dataset implicitly and obtains better performances. Due to the lack of X-ray image TB datasets, there are a small number of works on deep learning addressing the image-based classification of TB. In addition, the existing works can only classify X-ray images of a patient as TB or Healthy. This work presents a detailed investigation and analysis of 26 pretrained convolutional neural network (CNN) models using a recently released and large public database of TB X-ray. The proposed models have the capability to classify X-ray of a patient as TB, Healthy, or Sick but non-TB. Various visualization methods are adopted to show the optimal features learnt by the pretrained CNN models. Most of the pretrained CNN models achieved above 99% accuracy and less than 0.005 loss with 15 epochs during the training. All 7 different types of EfficientNet (ENet)-based CNN models performed better in comparison to other models in terms of accuracy, average and macro precision, recall, F1 score. Moreover, the proposed ENet-based CNN models performed better than other existing methods such as VGG16 and ResNet-50 for TB classification tasks. These results demonstrate that ENet-based models can be effectively used as a useful tool for TB classification.

V. Ravi (✉) · T. D. Pham
Center for Artificial Intelligence, Prince Mohammad Bin Fahd University,
Khobar, Saudi Arabia
e-mail: vravi@pmu.edu.sa

T. D. Pham
e-mail: tpham@pmu.edu.sa

H. Narasimhan
Indian Institute of Technology, Kanpur, India
e-mail: harinin@iitk.ac.in

T. D. Pham et al. (eds.), *Advances in Artificial Intelligence, Computation, and Data Science*, Computational Biology 31,
https://doi.org/10.1007/978-3-030-69951-2_9

Keywords Tuberculosis · CAD · Deep learning · Pretrained CNNs · X-ray · Image classification

9.1 Introduction

Tuberculosis (TB) is a serious infectious disease caused by the mycobacterium tuberculosis bacteria consisting of both drug-resistant and drug-sensitive sets. The drug-resistant set being unreactive to the medicine is the reason for many deaths all over the world. The drug-sensitive set is curable when treated properly at the early stages and the rate of fatality is more than 50% when left untreated [1]. Though the bacteriological culture and molecular analysis based diagnosis methods are accurate in detecting TB, the time-consuming and cost-restrictive attributes make them less effective for developing countries. Delay in the TB diagnosis increases the risk of disease transmission to contacts causing high infection rate in the world [2].

A recent report from the World Health Organization (WHO) states that tuberculosis is one of the top 10 causes of deaths in the lower-middle-income countries [3]. Also, the potential impact of the COVID-19 pandemic on global TB deaths results in a minimum increase of 200,000 deaths bringing the total from 1.6 to 1.8 million in the year 2020 [4]. These aspects make the early and accurate diagnosis of TB as a necessity to treat the patients effectively. The most popular and cheapest diagnostic methods are sputum smear microscopy and Chest X-rays (CXRs). For an early screening, the CXR is the most promoted tool by WHO and it is the most sensitive tool unlike sputum smear microscopy [5]. The CXR also performs ideally on detecting fibrotic lesions which is at the risk of developing active TB disease in the future [6]. Radiologists prefer the CXR for not just diagnosing the TB but also to predict the risk of developing TB with the patients which in turn helps to deliver the right treatment. Analyzing each X-rays with a qualified radiologist is time consuming and difficult for widespread adoption. Recent advances in computer-aided tuberculosis diagnosis (CTD) tools help radiologists to analyze CXRs in a faster, cheaper, and more accurate way.

Utilizing the machine learning and deep learning methods on detecting TB has shown improved performance in the recent years. Machine learning algorithms aim at extracting valuable features from CXRs and then classify them to detect the presence of TB. Using CNN for TB classification has evident improvement in the results as reported in [7–9]. These methods incorporated binary classification of CXRs as TB or non-TB. The binary classification might lead the model to perform on detecting the sickness but not particularly TB. Other deep learning models as proposed in [10, 11] classified different types of TB but were restricted to limited samples of data. The promising performance of deep learning in the analysis of chest X-rays can be accomplished by training the deep networks with some large size of samples. One such dataset proposed by Liu et al. [12] consists of a large number of samples with three classes: TB, Sick but non-TB, and Healthy. Using this dataset to train a deep learning model makes it better to identify TB from other lung infections. In [12], the architecture used for TB classification backbone are Resnet-50 and VGG-16 with both pretraining and training from scratch. The use of transfer learning from a pretrained network on classifying the images produces better results as a pretrained network is well learned on the high-level features extracted from the images. Here, we study the use of 26 pretrained networks available in literature for TB classification. These networks are analyzed, and their performance metrics are compared in terms of accuracy, precision, recall, and F1 score. The contributions of this chapter are outlined as follows.

- We compare the performance on TB classification on using 26 pretrained deep networks.
- We compare the TB classification performance with the baseline networks Resnet-50 and VGG-16 reported in [12].
- We propose the use of the EfficientNet (ENet) as the state-of-the-art performance for TB classification.
- We propose the feature visualization techniques based on the t-distributed stochastic neighbor embedding (t-SNE) to analyze the performance of the pretrained networks.

The remaining sections of this chapter are organized as follows: Existing works for TB classification are discussed in Sect. 9.2. Section 9.3 includes the details for the proposed method for TB classification. Section 9.4 is the description of the dataset. Results and analysis for all the CNN-based pretrained models are included and discussed in Sect. 9.5. Conclusion and future works are discussed in Sect. 9.6.

9.2 Related Work

TB detection and classification using deep learning is commonly performed by means of CXRs or computed tomography (CT) scans. A well-known public challenge in TB classification from CT scans is the ImageCLEF tuberculosis task [13]. This task has three subtasks, including detecting, classifying, and severity scoring of TB infection, and a combination of CNN and recurrent neural network (RNN) models was proposed for classifying TB achieved the top 5 performances [13]. Various deep Learning approaches were also proposed on classifying and assessing the severity of TB infection from CT scans [14]. Among which the transfer-learning approach was proposed [15] using the ResNet-50, which achieved the best accuracy. Another study involving two subtasks as severity scoring (SVR) and CT report (CTR), where a deep learning approach coupled with the generation of semantic descriptors performed well with both subtasks [16]. Another transfer-learning-based CT report generation using the ShuffleNet-v2 was ranked third among several other methods [17].

Jaeger et al. [18] proposed an automated approach for the diagnosis of TB by extracting texture and shape features of segmented lung regions in CXRs. These extracted features were used to classify with the support vector machine (SVM). Deep learning methods can overcome the need of feature extraction as it is capable of learning high-level as well as low-level features of images. A CNN model proposed in [19] achieved more than 80% accuracy for the binary classification as TB and non-TB on CXRs. This method utilized 600 CXRs for training and then validated on 200 images. To address the limited availability of data samples, image augmentation techniques and the use of pretrained networks were proposed in [20]. These deep convolutional neural networks (DCNNs) ensembled Alexnet and GoogLeNet achieved the AUC of 0.99 on identifying TB X-rays, and these models had the limitation on flagging the non-TB prevalent locations as TB positive. TB visualization methods for CNNs such as saliency maps and grad CAMs are proposed in [21], which provided good visual explanations. The TB-Net [22] is a deep neural network with a data augmentation pipeline to meet the limitation of inadequate data samples.

Lopes et al. [23] demonstrated the potential use of pretrained CNNs as a feature extractor for medical images and then used the extracted features for classification with SVM classifiers. A similar approach [24] used features from the top stack layer of Inception-v3 and classified the features with the SVM and achieved an accuracy about 95%. In [25], a different method of transfer learning was used by freezing only a few sets of layers in the pretrained models, which helped the deep learning model to learn the low-level features of the CXRs and improved the classification results. In [26], the use of the ensembles of three pretrained CNN models achieved higher accuracy and surpassed the need of feature extraction and data preparation techniques. Munadi et al. [27] proposed three image enhancement methods to improve the performance of pretrained CNN models. Multiple public datasets were combined in [28] to create large data samples that were trained on nine different pretrained CNNs for the classification task. Another approach utilizes large sample data (about 5,000 images of normal CXRs and 5,000 images of TB-present CXRs) for training a CNN model [29]. All these methods showed the performance improvement with the use of transfer learning.

A multiclass classification of TB based on the severity of the infection was implemented by Chithra et al. [30], where image segmentation and feature extraction algorithms were used to identify the patient severity as normal, level 1, and level 2. A TB detection system that categorized TB lesions was introduced by Xie et al. [11], which achieved good performance on two public datasets. CNN models for identifying seven different TB manifestations were proposed in [31], which resulted in a good performance in detecting lung abnormalities. Though these methods carried out a multiclass classification, the data used did not include other sickness such as lung infections being similar to TB. In [12], a large dataset on TB detection and classification was introduced, where the classes in the dataset are TB, Healthy, and Sick but non-TB. Using this dataset for training the deep learning model could make it more robust to identify TB infections from CXRs.

9.3 Methodology

An overview of the proposed architecture based on the EfficientNet-B0 (ENetbo) [32] for TB classification is shown in Fig. 9.1. The architecture has two sections, and they are discussed below.

Data preprocessing: The original X-ray image data shape is 512×512. The X-ray image datasets are resized into 224×224 and normalization is employed to convert the values into 0-1 range.

Feature extraction and Classification: The transfer learning approach with the ENetbo [32] pretrained model is used in this work for feature extraction. This architecture has less number of parameters compared to the other existing models such as Xception, VGG-16, VGG-19, ResNet-50, ResNet-101, ResNet-152, ResNet-50-v2, ResNet-101-v2, ResNet-152-v2, Inception-v3, InceptionResNet-v2, DenseNet-121, DenseNet-169, DenseNet-201, and NASNetLarge. The architecture consists of a compound scaling which uniformly scales all dimensions of depth, width, and resolution. More importantly, the ENetbo model performs better than the existing aforementioned architectures for the TB classification. Parameters for the ENetbo architecture are shown in Table 9.1.

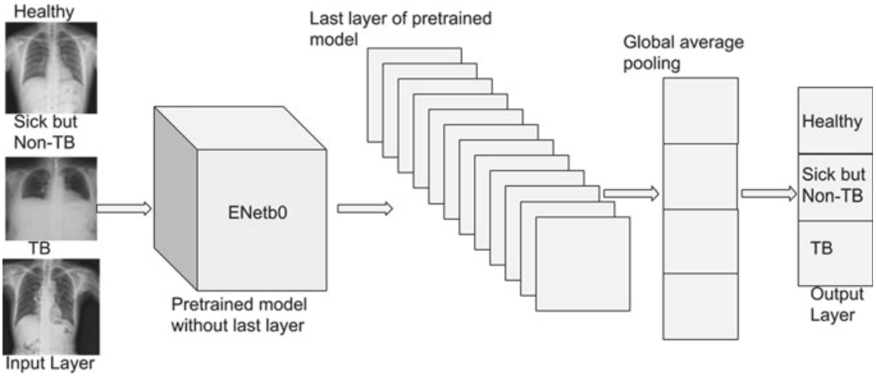


Fig. 9.1 Proposed architecture for TB classification using the ENetbo pretrained model

Table 9.1 Parameters for proposed model for TB classification based on ENetbo

Layer	Output shape	Parameters
EfficientNet-b0 (Functional)	(None, 8, 8, 1280)	4049564
global_average_pooling2d	(None, 1280)	0
Dense	(None, 3)	3843
Total params: 4,053,407		
Trainable parameters: 4,011,391		
Non-trainable parameters: 42,016		

In addition to ENetb0, this work has used other pretrained models such as Xception, VGG-16 [12], VGG-19, ResNet-50 [12], ResNet-101, ResNet-152, ResNe-50-v2, ResNet-101-v2, ResNet-152-v2, Inception-v3, InceptionResNet-v2, MobileNet, MobileNet-v2, DenseNet-121, DenseNet-169, DenseNet-201, NASNetMobile, NASNetLarge, EfficientNet-B1 (ENetb1), EfficientNet-B2 (ENetb2), EfficientNet-B3 (ENetb3), EfficientNet-B4 (ENetb4), EfficientNet-B5 (ENetb5), EfficientNet-B6 (ENetb6), and EfficientNet-B7 (ENetb7) for TB classification. The total parameters of Xception are VGG-16 [12], VGG-19, ResNet-50 [12], ResNet-101, ResNet-152, ResNet-50-v2, ResNet-101-v2, ResNet-152-v2, Inception-v3, InceptionResNet-v2, MobileNet, MobileNet-v2, DenseNet-121, DenseNet-169, DenseNet-201, NASNetMobile, NASNetLarge, ENetb1, ENetb2, ENetb3, ENetb4, ENetb5, ENetb6, and ENetb7 are 22,910,480, 138,357,544, 143,667,240, 25,636,712, 44,707,176, 60,419,944, 25,613,800, 44,675,560, 60,380,648, 23,851,784, 55,873,736, 4,253,864, 3,538,984, 8,062,504, 14,307,880, 20,242,984, 5,326,716, 88,949,818, 7,856,239, 9,177,569, 12,320,535, 19,466,823, 30,562,527, 43,265,143, and 66,658,687, respectively. All these pretrained models are CNN-based architectures. These pretrained models can reduce the training time, faster convergence rate, and achieve optimal performances for the TB classification. All these pretrained models were trained on ImageNet database which has more than one million images. This database has 1,000 classes and some of the classes of this ImageNet database are mouse, keyboard, etc. These models have learnt rich features which represent images from different classes. To achieve better performances on the ImageNet database, researchers used these different CNN architectures which have the same property with different scaling schemes. Scaling means arbitrarily increasing the CNN depth or width or using larger input image resolution for training and evaluation. In this work, these models are reused as a transfer-learning approach with the aim to transfer similar performances for the TB classification.

In all the models, the last layer is removed, and global average pooling and dense layer are added. The global average pooling operation estimates the average output of each feature map. As a result, the average pooling reduces the size of the feature maps. Finally, the dense layer has three neurons which are Healthy, Sick but non-TB and TB. The dense layer has a softmax activation function with categorical cross entropy as loss function. The softmax provides the probability values as a vector for all the three classes and the total sum is 1. The vector contains three values and choosing the maximum value results in a target class such as Healthy or Sick but non-TN or TB. Adam optimizer is used for all the models which helps to update the weights and learning rate with the aim to reduce the loss for the TB classification.

9.4 Description of Dataset

There are four different types of TB X-ray datasets available publicly for CTD research. However, all these datasets have very few numbers of X-ray data samples. The details of the existing datasets are shown in Table 9.2. Recently, deep

Table 9.2 Summary of TB datasets

Dataset	Number of X-ray images
Jaeger et al. [18]	138
Jaeger et al. [18]	662
Chauhan et al. [34]	156
Chauhan et al. [34]	150
Liu et al. [12]	**11,200**

Table 9.3 Summary of TB dataset [12]

Category	Healthy	Sick but non-TB	TB	Total
Train	2,495	2,589	544	5,628
Test	1,305	1,211	256	2,772
Total	3,800	3,800	800	8,400

learning models have achieved good performance in most of the image-based applications [33]. Training deep learning models requires large-scale dataset. Recently, with the aim to enhance the performance of CTD, a large-scale dataset is made publicly available for research purpose [12]. The detailed statistics of the datasets are shown in Table 9.3. To enhance the performance of the CTD, this dataset is used in the challenge.[1] In this work, the dataset was randomly divided into training (67%) and testing (33%) datasets. This large-scale TB dataset can help to enhance the performance of CTD and develop a real-time CTD system. The dataset has three classes: Healthy, Sick but non-TB, and TB. The class samples are shown in Fig. 9.1. All existing datasets have only two classes such as TB and Healthy. There are cases where the patient is Sick but non-TB. This can increase the false positive rate. To avoid this, another class such as Sick but non-TB was included. Around 5–10 expert radiologists were involved in the TB X-ray labeling process. The process included double-checking by radiologists to avoid errors in the labeling process.

9.5 Results and Discussions

In this work, TensorFlow[2] with Keras[3] python library was used to implement the models. The experiments of all the models were run inside Google Colab[4], which has Tesla K80 GPU of about 25GB RAM.

[1] https://competitions.codalab.org/competitions/25848.

[2] https://www.tensorflow.org/.

[3] https://keras.io/.

[4] https://colab.research.google.com.

The performances of all the models were estimated by the most commonly used metrics such as accuracy, precision, recall, and F1 score, which were computed from the confusion matrix. The confusion matrix is a representation of true positive (TP), true negative (TN), false negative (FN), and false positive (FP). Since the dataset is imbalanced, the metrics for both macro and weighted were used with accuracy, precision, recall, and F1 score. The literature survey shows that macro is the most commonly used metric with precision, recall and F1 score for an imbalanced dataset. Macro computes the precision, recall and F1 score for each class and returns the average without considering the proportion for each class in the TB dataset. The weighted computes the precision, recall, and F1 score for each class and returns the average by considering the proportion for each class in the TB dataset. The aforementioned metrics are defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{9.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{9.3}$$

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{9.4}$$

Here, TP and TN are the number of images correctly classified and misclassified to a given class by the pretrained model, respectively, FP is a measure that indicates that the presence of a TB when TB is not present, and FN is the opposite to FP that fails to indicate the presence of TB when TB is present.

All the pretrained models were trained using the training dataset for TB classification. During the training, the models were run for 15 epochs with the batch size of 64 and learning rate of 0.001. Most models achieved the training accuracy in the range 95–99.99% with the epochs in the range 10–15. We run experiments of ENetb0 for more than 15 epochs during the training and we observed successive reduction in train accuracy over epochs after 15. This may be due to overfitting. Thus, we decided to set 15 as the optimal value for epochs. The training accuracy for the best perform model such as ENetbo in terms of accuracy and loss is shown in Figs. 9.2 and 9.3, respectively. These figures show that the training model of ENetbo achieved a better performance in terms of accuracy and loss within 15 epochs. After 10 epochs most of the models obtained good performances in terms of accuracy and loss. For comparison purposes, the graphs for training architectures such as VGG-16 and ResNet-50 in terms of accuracy and loss are shown in Figs. 9.4 and 9.5, respectively. These graphs show that the ResNet-50 achieved an accuracy in the range 95–99.9% within 5. However, the VGG-16 showed successive performance enhancement in terms of accuracy and loss.

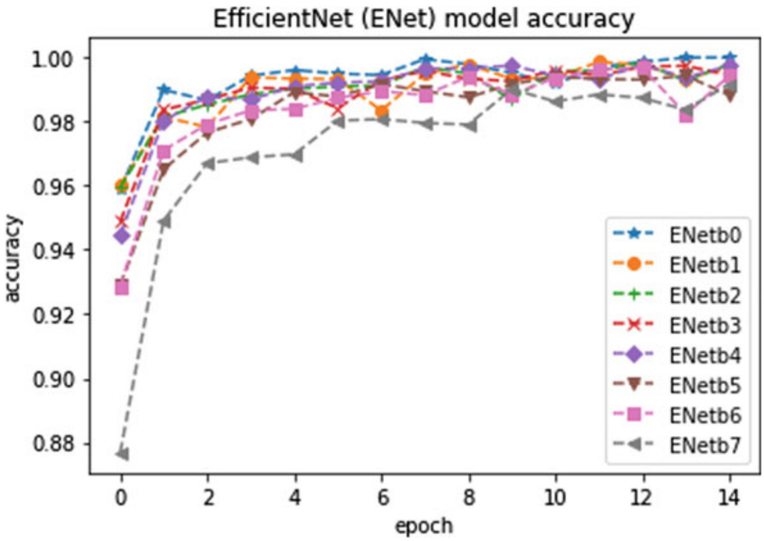


Fig. 9.2 Training accuracy of ENet models

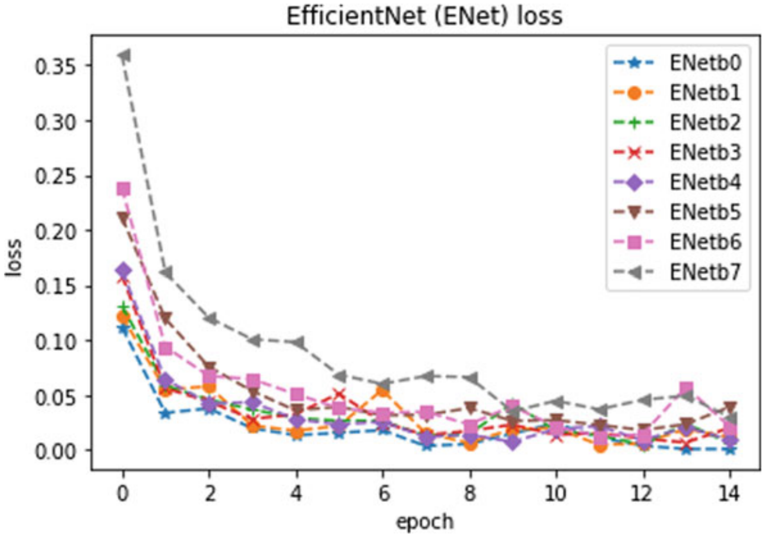


Fig. 9.3 Training loss of ENet models

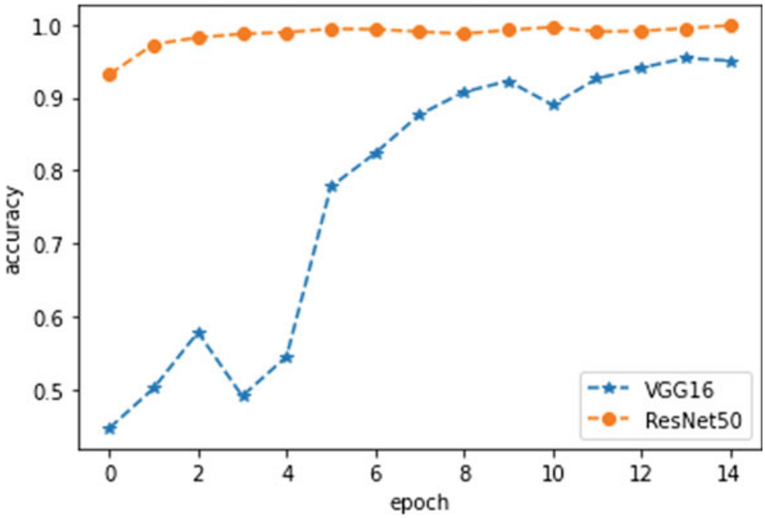


Fig. 9.4 Training accuracy of VGG-16 and ResNet-50 for TB classification

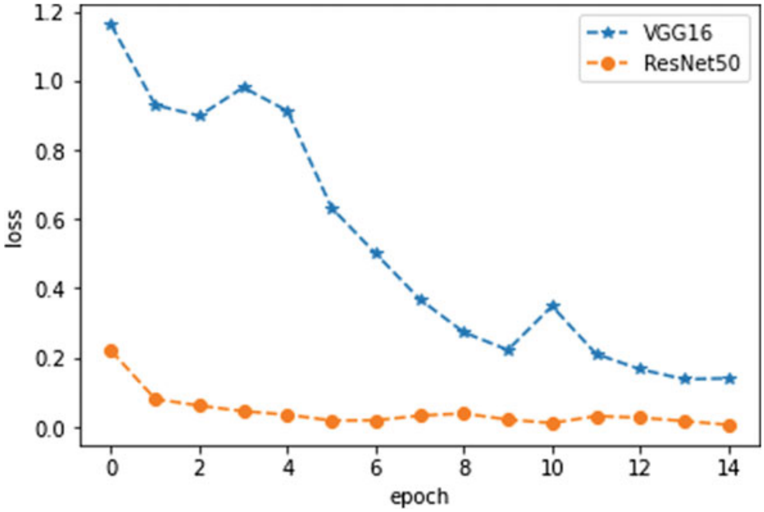


Fig. 9.5 Training loss of VGG-16 and ResNet-50 for TB classification

Next, the trained model performances were assessed using the test datasets for TB classification. The testing results are shown in Table 9.4, which shows that the performance of ENetbo is the best compared to other pretrained models and existing models such as VGG-16 and ResNet-50 in terms of precision, recall and F1 score. The model shows 99% for accuracy, precision, recall, and F1 score for both weighted and macro. Since the dataset is imbalanced, performances for both weighted and macro are reported for each metric such as precision, recall, and F1 score. Most importantly, the ENetbo has less number of parameters and as a result, the proposed model is computationally inexpensive compared to all other models. The performances of other pretrained models such as InceptionResNet-v2 and DenseNet-121 are closer to the ENetbo. There is only 1–2% difference between the precision, recall, and F1 score obtained from InceptionResNet-v2, DenseNet-121, and ENetbo. The performances of other models such as Xception, ResNet-101, Inception-v3, and MobileNet are similar but the performance of ENetbo is the best among these three models. All these three models achieved more than 90% for precision, recall, F1 score for both weighted and macro. MobileNet-v2, NASNetMobile, and NASNetLarge showed less than 50% for the recall and F1 score in both average and macro in comparison to other pretrained models. ENet models such as ENetb2 and ENetb4 showed similar performances as ENetb0. However, both ENetb2 and ENetb4 have more parameters than ENetbo. Hence, they are computationally expensive during the training and testing in comparison to ENetb0. More importantly, the performances of VGG-16 and Renet-50 are not desirable and have larger differences in terms of weighted and macro precision, recall, and F1 score. In summary, the ENetbo is more robust and has the capability to accurately learn and classify the variants of the patients' X-rays. In other words, the ENetbo model can classify the unseen patients' X-ray data more accurately than other models.

Results for the best performing model (ENetbo) is shown in Table 9.5, which shows that the ENetb0 model has attained 100% for both precision and recall. For the class Sick but non-TB, the ENetbo achieved 100% for recall and 99% for both precision and F1 score.

The confusion matrices for ENet models such as ENetbo, ENetb1, ENetb2, ENetb3, ENetb4, ENetb5, ENetb6, and ENetb7 are shown in Fig. 9.6. The ENetb0 models misclassified 8 samples of Healthy as Sick but non-TB, 2 samples of Sick but non-TB as TB and 6 samples of TB as Sick but non-TB. Overall, the model has learned to distinguish the patients of Healthy or TB more accurately. There are 8 patients which were misclassified as Sick but non-TB. Overall, the model achieved an accuracy of 99.42% with a misclassification rate of 0.0058. For the Healthy class, ENetb1, ENetb4, ENetb5, and ENetb7 achieved better classification rates compared to the ENetbo but resulted in poorer classification rates for the classes Sick but non-TB and TB. Models such as ENetb2 and ENetb6 achieved better classification rates for the Sick but non-TB compared to the ENetb0 but had poorer classification rates for other two classes such as Healthy and TB. The ENetb4 attained better classification rate for the TB compared to the ENetbo model. Overall, each model has its unique characteristics and combining all the models using a feature fusion approach can be expected to improve the classification. Confusion matrices for existing models

Table 9.4 TB classification results

Model	Accuracy	Type	Precision	Recall	F1 score
Xception	0.93	Macro	0.94	0.91	0.92
		Weighted	0.94	0.93	0.93
VGG-16 [12]	0.94	Macro	0.91	0.85	0.87
		Weighted	0.94	0.94	0.94
VGG-19	0.47	Macro	0.16	0.33	0.21
		Weighted	0.22	0.47	0.30
ResNet-50 [12]	0.10	Macro	0.63	0.34	0.07
		Weighted	0.82	0.10	0.03
ResNet-101	0.91	Macro	0.93	0.79	0.83
		Weighted	0.92	0.91	0.91
ResNet-152	0.48	Macro	0.79	0.37	0.28
		Weighted	0.74	0.48	0.32
ResNet-50-v2	0.85	Macro	0.90	0.81	0.84
		Weighted	0.88	0.85	0.85
ResNet-101-v2	0.85	Macro	0.90	0.81	0.84
		Weighted	0.88	0.85	0.85
ResNet-152-v2	0.77	Macro	0.85	0.68	0.71
		Weighted	0.84	0.77	0.76
Inception-v3	0.94	Macro	0.87	0.96	0.90
		Weighted	0.96	0.94	0.95
InceptionResNet-v2	0.99	Macro	0.99	0.97	0.98
		Weighted	0.99	0.99	0.99
MobileNet	0.98	Macro	0.98	0.95	0.96
		Weighted	0.98	0.98	0.98
MobileNet-v2	0.11	Macro	0.68	0.35	0.09
		Weighted	0.89	0.11	0.06
DenseNet-121	0.99	Macro	0.99	0.97	0.98
		Weighted	0.99	0.99	0.99
DenseNet-169	0.52	Macro	0.63	0.61	0.40
		Weighted	0.75	0.52	0.38
DenseNet-201	0.71	Macro	0.71	0.76	0.63
		Weighted	0.86	0.71	0.70
NASNetMobile	0.47	Macro	0.21	0.35	0.24
		Weighted	0.24	0.47	0.31
NASNetLarge	0.47	Macro	0.82	0.33	0.22
		Weighted	0.75	0.47	0.30
ENetb0	**0.99**	**Macro**	**0.99**	**0.99**	**0.99**
		Weighted	**0.99**	**0.99**	**0.99**

(continued)

Table 9.4 (continued)

Model	Accuracy	Type	Precision	Recall	F1 score
ENetb1	0.99	Macro	0.99	0.98	0.99
		Weighted	0.99	0.99	0.99
ENetb2	0.99	Macro	0.99	0.99	0.99
		Weighted	0.99	0.99	0.99
ENetb3	0.96	Macro	0.93	0.97	0.95
		Weighted	0.97	0.96	0.96
ENetb4	0.99	Macro	0.99	0.99	0.99
		Weighted	0.99	0.99	0.99
ENetb5	0.98	Macro	0.98	0.96	0.97
		Weighted	0.98	0.98	0.98
ENetb6	0.99	Macro	0.99	0.98	0.99
		Weighted	0.99	0.99	0.99
ENetb7	0.99	Macro	0.98	0.98	0.98
		Weighted	0.99	0.99	0.99

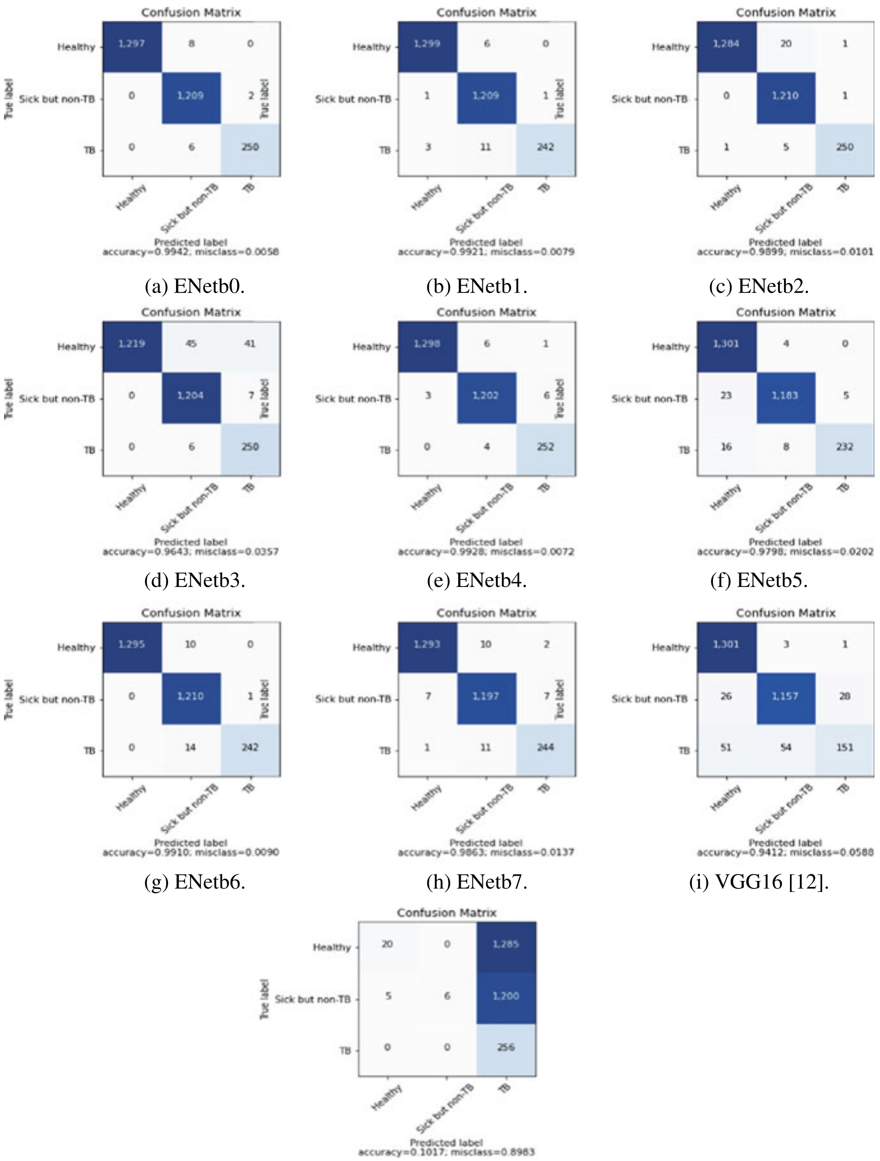
Table 9.5 Class-wise results of ENetb0 model for TB classification

Category	Precision	Recall	F1 score
Healthy	1.00	0.99	1.00
Sick but non-TB	0.99	1.00	0.99
TB	0.99	0.98	0.98

VGG-16 and ResNet-50 are shown in Fig. 9.6, whose performance are not better than the ENetb0; however, VGG-16 and Renset-50 achieved better classification rates for Healthy and TB. Thus, the optimal features of these two models can be used along with the ENetb0 to enhance the performance for TB classification. However, this requires a feature selection approach to extract important features from VGG-16 and Resnet-50 models.

9.5.1 Feature Visualization of pretrained models for TB classification

All the pretrained models have more than one hidden layers and patients' image data passed across these hidden layers to learn to differentiate between Healthy, Sick but non-TB, and TB. Generally, the size of hidden layers is large in nature. Since deep learning models are black-box [33], feature visualization is an important task in TB classification. Feature visualization helps to gain insight into the learned optimal features of a model to allow more detailed analysis of the results. Generally,



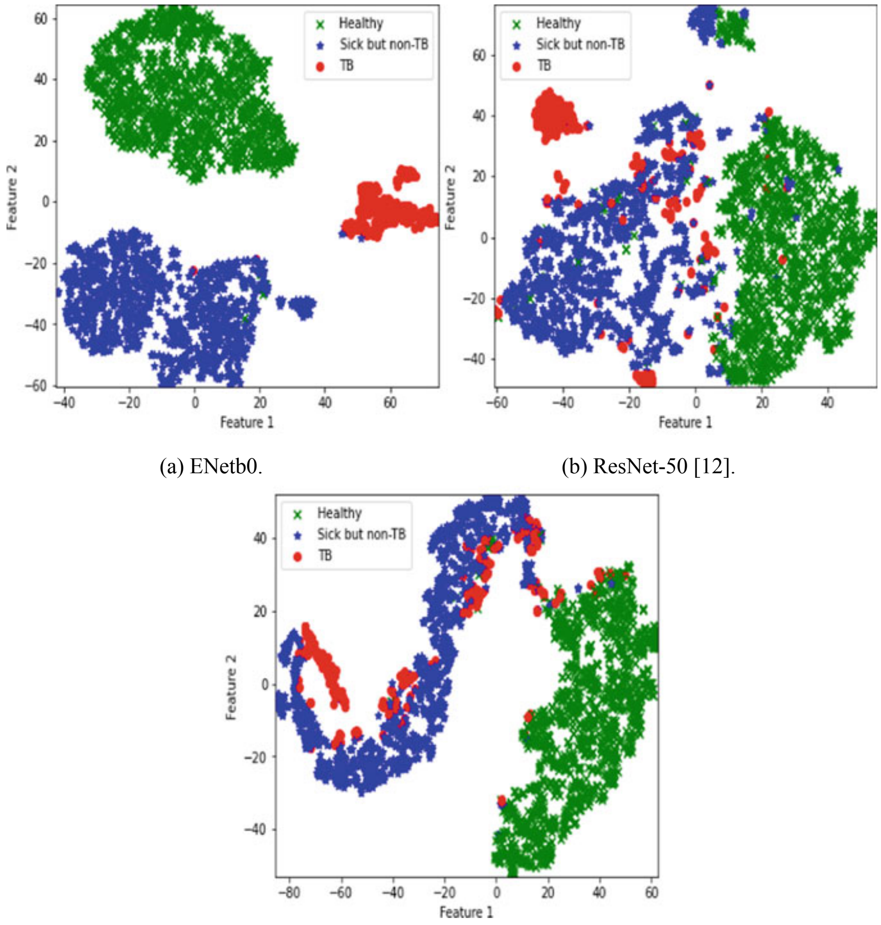
(a) ENetb0. (b) ENetb1. (c) ENetb2.

(d) ENetb3. (e) ENetb4. (f) ENetb5.

(g) ENetb6. (h) ENetb7. (i) VGG16 [12].

(j) ResNet50 [12].

Fig. 9.6 Confusion matrix



(a) ENetb0.

(b) ResNet-50 [12].

(c) VGG-16 [12].

Fig. 9.7 Penultimate layer feature representation using *t*-SNE

the information can help to find the reason behind the misclassifications. Following, in this work, the penultimate layer features of size 1,280 of EBetb0 and existing methods such as VGG-16 and ResNet-50 are passed into the *t*-SNE. This reduces the dimension of features of size 1,280 to two dimensions using the principal component analysis. The low-dimensional feature representation of ENetb0, VGG-16, and ResNet-50 is shown in Fig. 9.7. The *t*-SNE plot of Enetbo shows that the model has completely learned the optimal features to distinguish Healthy, Sick but non-TB and TB X-rays. We can observe from the figure that the X-rays of Healthy, Sick but non-TB, and TB appear to be clustered. Moreover, the model obtained only very few misclassifications between Sick but non-TB and TB. The *t*-SNE plot for VGG-16 shows that the model learnt optimal features to differentiate the patients of Healthy condition and TB but there were misclassifications for Healthy, Sick but non-TB, and TB. The ResNet-50 learned optimal features to recognize the Healthy patients but it completely failed to classify the Sick but non-TB and TB. The penultimate layer features of ENetbo, VGG-16, and ResNet-50 can be merged using a feature fusion approach. It is possible that this type of method can obtain a better performance and be considered as an issue for future work.

9.6 Conclusions

TB classification using deep learning is an important area of research that has been recognized to be clinically useful by radiologists. In this study, ENetbo-based model was proposed for TB classification using X-rays. A detailed analysis of the proposed model was carried out using very large datasets. The performance of the proposed model was compared with 23 pretrained CNN models and existing VGG-16 and Resnet-50. In all experiments, the proposed model performed better than the two existing models and other pretrained CNN models for TB classification.

All these models can also be used for other image modality such as CT scans for TB classification. The proposed model can be used as a diagnostic tool for an early diagnosis of TB. Moreover, the model has the capability to analyze very sensitive details of X-rays, and as a result it can provide more accurate results. The dataset used in this study has information of TB areas as bounding boxes with labels as Healthy, Sick but non-TB, and TB. The class of TB has stage information such as activate, latent, and uncertain together with the comprehensive clinical information such as age and gender for X-ray of patients. Thus, the model can be developed for detecting TB stages and the existing clinical information can be used to enhance TB diagnosis.

References

1. Fauci AS, Braunwald E, Kasper DL, Harrison TR (2009) Princípios de medicina interna, vol I. McGraw-Hill Interamericana, Madrid
2. Golub JE, Bur S, Cronin WA, Gange S, Baruch N, Comstock GW, Chaisson RE (2006) Delayed tuberculosis diagnosis and tuberculosis transmission. Int J Tuberc Lung Dis 10(1):24–30
3. World Health Organization (2019) Global tuberculosis report 2019. World Health Organization, Geneva
4. World Health Organization (2020) Global tuberculosis report 2020. World Health Organization, Geneva
5. Leung CC (2011) Reexamining the role of radiography in tuberculosis case finding. Int J Tuberc Lung Dis 15(10):1279

6. Melendez J, Sánchez CI, Philipsen RH, Maduskar P, Dawson R, Theron G, Van Ginneken B (2016) An automated tuberculosis screening strategy combining X-ray-based computer-aided detection and clinical information. Sci Rep 6:25265
7. Colombo Filho ME, Galliez RM, Bernardi FA, de Oliveira LL, Kritski A, Santos MK, Alves D (2020) Preliminary results on pulmonary tuberculosis detection in chest X-ray using convolutional neural networks. In: International Conference on Computational Science, pp 563–576. Springer, Cham
8. Oloko-Oba M, Viriri S (2020) Tuberculosis abnormality detection in chest X-rays: a deep learning approach. In: International Conference on Computer Vision and Graphics, pp 121–132. Springer, Cham
9. Abideen ZU, Ghafoor M, Munir K, Saqib M, Ullah A, Zia T, Zahra A (2020) Uncertainty assisted robust tuberculosis identification with Bayesian convolutional neural networks. IEEE Access 8:22812–22825
10. Cao Y, Liu C, Liu B, Brunette MJ, Zhang N, Sun T, Curioso WH (2016) Improving tuberculosis diagnostics using deep learning and mobile health technologies among resource-poor and marginalized communities. In: 2016 IEEE first international conference on connected health: applications, systems and engineering technologies (CHASE), pp 274–281. IEEE
11. Xie Y, Wu Z, Han X, Wang H, Wu Y, Cui L, Chen Z (2020) Computer-aided system for the detection of multicategory pulmonary tuberculosis in radiographs. J Healthcare Eng
12. Liu Y, Wu YH, Ban Y, Wang H, Cheng MM (2020) Rethinking computer-aided tuberculosis diagnosis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 2646–2655
13. Sun J, Chong P, Tan YXM, Binder A (2017) ImageCLEF 2017: ImageCLEF tuberculosis task-the SGEast submission. In: CLEF (working notes)
14. Cid YD, Liauchuk V, Kovalev V, Müller H (2018) Overview of ImageCLEFtuberculosis 2018-etecting multi-drug resistance, classifying tuberculosis types and assessing severity scores. In: CLEF (working notes)
15. Gentili A (2018) ImageCLEF2018: transfer learning for deep learning with CNN for tuberculosis classification. In: CLEF (working notes)
16. Hamadi A, Cheikh NB, Zouatine Y, Menad SMB, Djebbara MR (2019) ImageCLEF 2019: deep learning for tuberculosis CT image analysis. In: CLEF (working notes)
17. Che J, Ding H, Zhou X (2020) Chejiao at ImageCLEFmed tuberculosis 2020: CT report generation based on transfer learning. In: CLEF2020 working notes. CEUR workshop proceedings, Thessaloniki, Greece, CEUR-WS. http://ceurws.org (September 22–25 2020)
18. Jaeger S, Karargyris A, Candemir S, Folio L, Siegelman J, Callaghan F, Thoma G (2013) Automatic tuberculosis screening using chest radiographs. IEEE Trans Med Imaging 33(2):233–245
19. Hooda R, Sofat S, Kaur S, Mittal A, Meriaudeau F (2017) Deep-learning: a potential method for tuberculosis detection using chest radiography. In: 2017 IEEE international conference on signal and image processing applications (ICSIPA), pp 497–502. IEEE
20. Lakhani P, Sundaram B (2017) Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks. Radiology 284(2):574–582
21. Pasa F, Golkov V, Pfeiffer F, Cremers D, Pfeiffer D (2019) Efficient deep network architectures for fast chest X-ray tuberculosis screening and visualization. Sci Rep 9(1):1–9
22. Ghorakavi RS (2019) TBNet: pulmonary tuberculosis diagnosing system using deep neural networks. arXiv:1902.08897
23. Lopes UK, Valiati JF (2017) Pre-trained convolutional neural networks as feature extractors for tuberculosis detection. Comput Biol Med 89:135–143
24. Samuel RDJ, Kanna BR (2019) Tuberculosis (TB) detection system using deep neural networks. Neural Comput Appl 31(5):1533–1545
25. Nguyen QH, Nguyen BP, Dao SD, Unnikrishnan B, Dhingra R, Ravichandran SR, Chua MC (2019) Deep learning models for tuberculosis detection from chest X-ray images. In: 2019 26th international conference on telecommunications (ICT), pp 381–385. IEEE
26. Evangelista LGC, Guedes EB (2019) Ensembles of convolutional neural networks on computer-aided pulmonary tuberculosis detection. IEEE Lat Am Trans 17(12):1954–1963

27. Munadi K, Muchtar K, Maulina N, Pradhan B (2020) Image enhancement for tuberculosis detection using deep learning. IEEE Access
28. Rahman T, Khandakar A, Kadir MA, Islam KR, Islam KF, Mazhar R, Ayari MA (2020) Reliable tuberculosis detection using chest X-ray with deep learning, segmentation and visualization. IEEE Access 8:191586–191601
29. Sathitratanacheewin S, Pongpirul K (2018) Deep learning for automated classification of tuberculosis-related chest X-ray: dataset specificity limits diagnostic performance generalizability. arXiv:1811.07985
30. Chithra RS, Jagatheeswari P (2020) Severity detection and infection level identification of tuberculosis using deep learning. Int J Imaging Syst Technol 30(4):994–1011
31. Guo R, Passi K, Jain CK (2020) Tuberculosis diagnostics and localization in chest X-rays via deep learning models. Front Artif Intell 3:74
32. Tan M, Le QV (2019) Efficientnet: rethinking model scaling for convolutional neural networks. arXiv:1905.11946
33. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
34. Chauhan A, Chauhan D, Rout C (2014) Role of gist and PHOG features in computer-aided diagnosis of tuberculosis without segmentation. PLoS One 9(11):e112980

Chapter 10
AI in the Detection and Analysis of Colorectal Lesions Using Colonoscopy

Zhe Guo, Xin Zhu, Daiki Nemoto, and Kazunori Togashi

10.1 Introduction

Figure 10.1 illustrates the organization of this chapter. We begin with a brief review of colon anatomy and an overview of general information on colorectal cancers (CRCs). In Sect. 1.2, we introduce the details of colonoscopy, the most important tool for the screening, diagnosis, and therapy of CRCs. The colorectal cancer stages are introduced in Sect. 1.3. In Sect. 1.4, we give a brief introduction about the history of implementing artificial intelligence (AI) to assist endoscopists in the detection and analysis of colorectal polyps and cancer.

10.1.1 Colorectum and Colorectal Cancer

CRCs develop in colon or rectum, i.e., large intestine as shown in Fig. 10.2. The colon and rectum are parts of human digestive system, i.e., gastrointestinal system. Digestive systems process food for acquiring energy, water, and nutrition, and remove solid waste (fecal matter or stool). After food is chewed and swallowed in mouths, remaining travels from esophagus to stomach, small intestine, and then colon.

In Fig. 10.2, circumambient sections composed of tubes are colorectum. The first section of colon starting from cecum is called ascending colon, i.e., the beginning of

Z. Guo · X. Zhu (✉)
Biomedical Information Engineering Lab, The University of Aizu, Aizuwakamatsu, Fukushima 965-8580, Japan
e-mail: zhuxin@u-aizu.ac.jp

D. Nemoto · K. Togashi
Division of Proctology, Aizu Medical Center, Fukushima Medical University, Aizuwakamatsu, Fukushima 969-3492, Japan

T. D. Pham et al. (eds.), *Advances in Artificial Intelligence, Computation, and Data Science*, Computational Biology 31,
https://doi.org/10.1007/978-3-030-69951-2_10

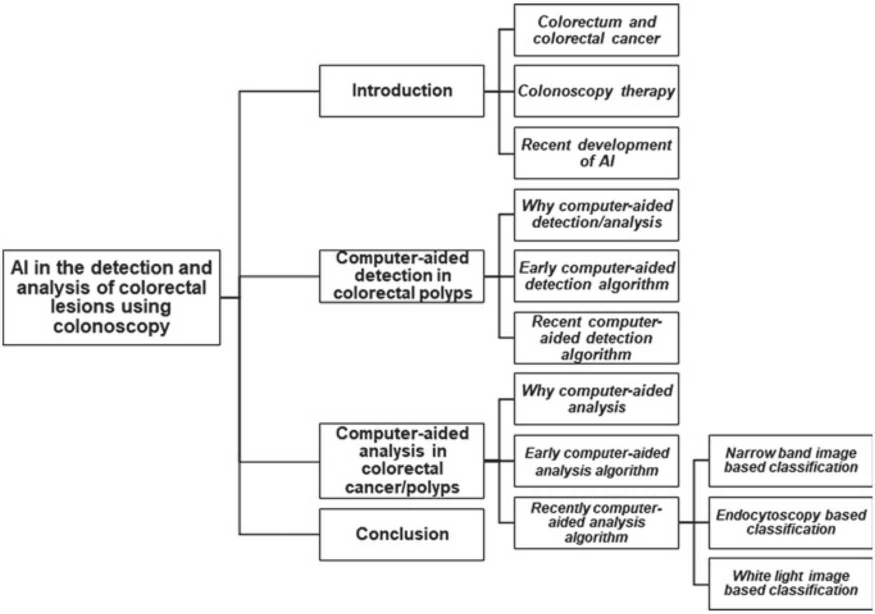

AI in the detection and analysis of colorectal lesions using colonoscopy
Introduction
Colorectum and colorectal cancer
Colonoscopy therapy
Recent development of AI
Computer-aided detection in colorectal polyps
Why computer-aided detection/analysis
Early computer-aided detection algorithm
Recent computer-aided detection algorithm
Computer-aided analysis in colorectal cancer/polyps
Why computer-aided analysis
Early computer-aided analysis algorithm
Recently computer-aided analysis algorithm
Narrow band image based classification
Endocytoscopy based classification
White light image based classification
Conclusion

Fig. 10.1 The organization of this chapter

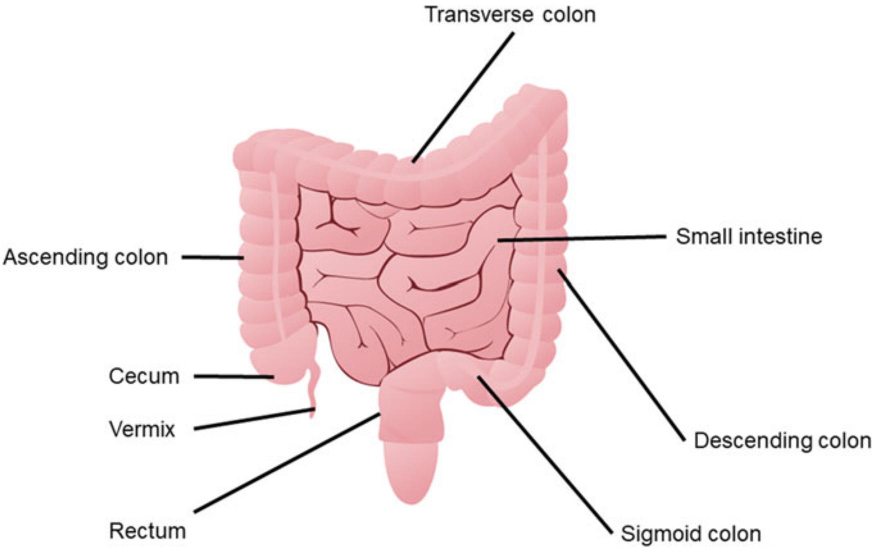


Fig. 10.2 The structure of colorectum

colorectum. Digested food from small intestine moves upward via ascending colon at the right side of the abdomen to transverse colon, the second section of colorectum. Transverse colon is named as it crosses the body from the right to the left side. Descending colon, the third section of colorectum, descends at the left side. The fourth section is called sigmoid colon because of its “S” shape. The sigmoid colon joins rectum, the final part of colorectum.

Cancer is a leading cause of death, killing millions of people annually in the world. CRC has the world’s third largest incidence and fourth largest mortality among all cancer [1]. The incidence of CRCs is increasing rapidly and has been more than that of lung cancer recently, especially in developed countries [2]. CRC is prevalent in both developed and developing countries. The number of subjects suffered from CRCs in developing countries was even larger than those of developed countries in the 1990s [3]. The discrepancy might attribute to the fact that CRCs are rarely screened by physicians using colonoscopy in developing countries because the stool-based testing of the traditional CRC screening method is less sensitive than colonoscopy examinations [4]. In Japan, CRCs were cancer with the largest incidence in 2017 and cancer with the second largest mortality in 2018 [5]. The incidence and mortality rates of CRCs are expected to steadily increase in the future because the coming of aging society [5]. The incidence of CRCs is also increasing in the young generations, although they account for a relatively lower proportion of overall incidences [6, 7].

Fortunately, CRCs are usually developed from precursors, such as adenomatous polyps with a rather slow progressive speed, and sessile serrated lesions. Therefore, CRCs are preventable by screening through timely colonoscopy examinations. The incidence and mortality of CRCs would be reduced by 76–90% and 53%, respectively, by colonoscopic polypectomy according to the 1993 and 2012 National Polyp Studies [8, 9].

Typically, the group of over 60 years old has the highest incidence of colorectal cancer; therefore, the screening of CRCs should be performed at the age of 40–50 years old. Subjects with a family history of colorectal polyps or cancer are also at a higher risk, so they are recommended to have a colonoscopy examination every 3–5 years. Patients with Crohn’s and ulcerative colitis also have higher risks of CRCs [10]. Subjects of African origin have a higher risk of CRCs and also have a higher risk of CRC-related death [11]. Some measures in daily life may reduce people’s risks of CRCs. Evidence suggests a healthy diet, rich in fruits and vegetables, low in fat, high in fiber, avoiding smoking, and maintaining a good weight [12, 13]. Furthermore, if we can perform early therapy for colorectal cancer, the survival rate can be significantly improved even although surgery operations are still the main therapeutic methods for colorectal cancer in case of deep invasiveness and metastasis.

A fecal occult blood test is also used for the screening of colorectal cancer because of convenience and low cost. However, CRCs are very complicated and versatile in symptoms as follows:

i. When CRCs is in an early stage with no clinical symptoms, stool would be completely normal. Therefore, subjects with normal stools may also have intestinal cancer. Colonoscopy or other examinations are needed to avoid false negative.

ii. If a tumor causes intestinal mucosal damage and bleeding, there will be symptoms of blood in stool, such as bloody stools or blood clots.

iii. If a tumor grows and causes intestinal stenosis, it will make the stool thin. If it also causes changes in intestinal functions, symptoms of constipation or diarrhea would appear.

As a result, the sensitivity of fecal occult blood test is significantly lower than that of colonoscopy examination.

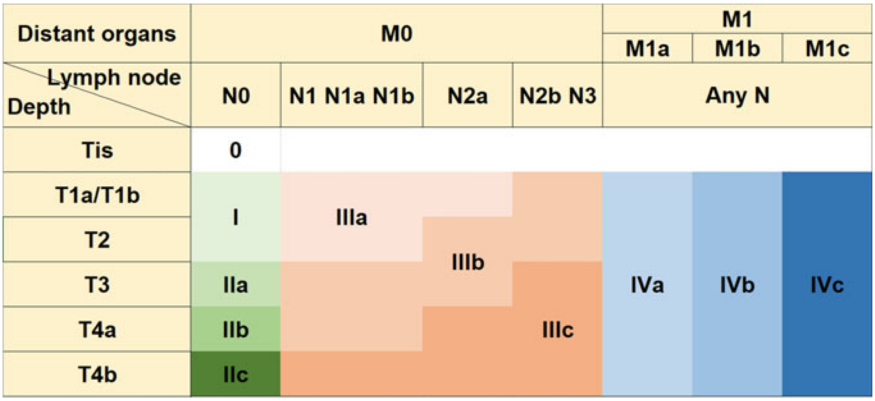
10.1.2 Colorectal Cancer Stages

The colorectal cancer stages as listed in Table 10.1 [14] are mainly determined by the extent to which CRC has spread through those layers of the colorectum wall and/or into lymph nodes or other organs. There are three parameters as follows to describe the CRCs:

T (tumor): This describes the size and invasive depth of an original tumor.
N (node): This indicates whether cancer is present in lymph nodes.
M (metastasis): This refers to whether cancer has spread to other organs or other parts of the body. The T, N, and M scores collaboratively determined the stages of CRCs.

A number (I-IV or 1-4) or a letter X is assigned to each parameter. Using this colorectal cancer staging system, a larger number indicates increasing severity. For instance, a T1 indicates a less invasive and smaller tumor than T2 or T3. X means the information could not be assessed.

Table 10.1 Colorectal cancer stages (modified according to the standard of America Joint Committee on Cancer [14])



Distant organs	M0				M1		
					M1a	M1b	M1c
Lymph node Depth	N0	N1 N1a N1b	N2a	N2b N3	Any N		
Tis	0						
T1a/T1b	I	IIIa	IIIa	IIIb	IVa	IVb	IVc
T2	I	IIIa	IIIb	IIIb	IVa	IVb	IVc
T3	IIa	IIIb	IIIb	IIIc	IVa	IVb	IVc
T4a	IIb	IIIb	IIIc	IIIc	IVa	IVb	IVc
T4b	IIc	IIIc	IIIc	IIIc	IVa	IVb	IVc

Stage 0: Abnormal cells or growths, such as hyperplastic polyps, are found on the mucosa (the inner lining of the colorectum). This is known as carcinoma in situ as a result of the cells are confined to their place of origin and there is no evidence they have spread to other layers of the colorectum or to lymph nodes or have metastasized to other organs or other parts of the body. Cells found in stage 0 CRCs may be cancerous or precancerous.

Stage I: The CRCs have grown into the intestinal wall, through the mucosa (inner lining), and into the submucosa and may have entered the muscle. There is no evidence the CRCs have spread to lymph nodes or distant organs or other parts of the body.

Stage II: Stage II CRCs are divided into three categories:

- Stage IIA: The CRCs have grown into the serosa (the outermost layer of the colorectum), but have not grown through it. It has not reached nearby organs or lymph nodes, and has not spread to distant organs or other parts of the body.
- Stage IIB: The CRCs have grown through all the layers of the colorectum, but have not spread to lymph nodes or distant organs or other parts of the body.
- Stage IIC: The CRCs have grown through all the layers of the colorectum and also have grown into nearby organs or tissues. The CRCs have not spread to the lymph nodes or distant organs or other parts of the body.

Stage III: Stage III CRCs are divided into three categories:

- Stage IIIA: The CRCs have grown into the intestine wall, through the mucosa (the inner lining), and into the submucosa and may have entered the muscle. Cancer has spread to up to three lymph nodes near the site of the primary tumor.
- Stage IIIB: The CRCs have grown into or through the outermost layer of the colorectum and may have spread into nearby organs or tissues. The CRCs have spread to up to three lymph nodes near the primary site but have not spread to distant organs or other parts of the body.
- Stage IIIC: The CRCs have grown into or through the outermost layer of the colorectum and may have spread to four or more lymph nodes near the primary site. The CRCs have also spread to nearby organs.

Stage IV: Stage IV is the most advanced stage of CRCs. If patients have been diagnosed with stage IV of CRCs, it means that the CRCs have metastasized to distant organs or other parts of the body, such as the liver or stomach. The CRCs may or may not have grown through the wall of the colorectum, and lymph nodes may or may not have been affected. Stage IV of CRCs is further divided into two categories, depending on whether or not the metastasis has affected more than one organ or other parts of the body. The original tumor can be of any size and lymph nodes may or may not be involved, but if the CRCs have spread to another organ it is considered stage IVA, while more than one organ would be defined as stage IVB.

Stage IV of CRCs can be defined by any T or N category, with the only difference stemming from whether the M1 or M2 assignment is more appropriate.

In both forms of stage IV CRCs, the tumor can be of any size (T), and lymph nodes may or may not be involved (N). M1a indicates that the CRCs have spread to just one organ, while M1b would mean that more than one organ has been affected.

10.1.3 Colonoscopy and Colorectal Polyps

Figure 10.3 illustrates a scene of colonoscopy. Colonoscopy is the gold standard for CRCs screening and the most common therapeutic tool for CRCs [15, 16]. Most colon polyps are harmless, but some may develop into cancer [17, 18]. Hyperplastic polyps are benign and usually do not have a prompt threat, but they are recommended to be removed because they would grow up and cause colon obstructions. However, adenomatous polyps and sessile serrated lesions may develop to CRCs, and therefore should be timely detected and removed. 80–90% of CRCs develop from polyps. The evolution process of normal intestinal mucosa → diminutive polyps ($\leq$ 5mm) → small polyps ($> 5 \sim 10$mm) → large polyps ($>$ 10mm) → dysplasia → carcinoma → metastasis.

Endoscopy can be used to perform minimally invasive surgical operations, with lower risks compared with traditional ones. Before endoscopy and diagnostic imaging were widely used, most internal diseases could only be treated through open surgery. For the therapy of colorectal polyps and cancer, polypectomy is a minimally invasive treatment technique that endoscopists use endoscopy tools to resect polyps under colonoscopy.

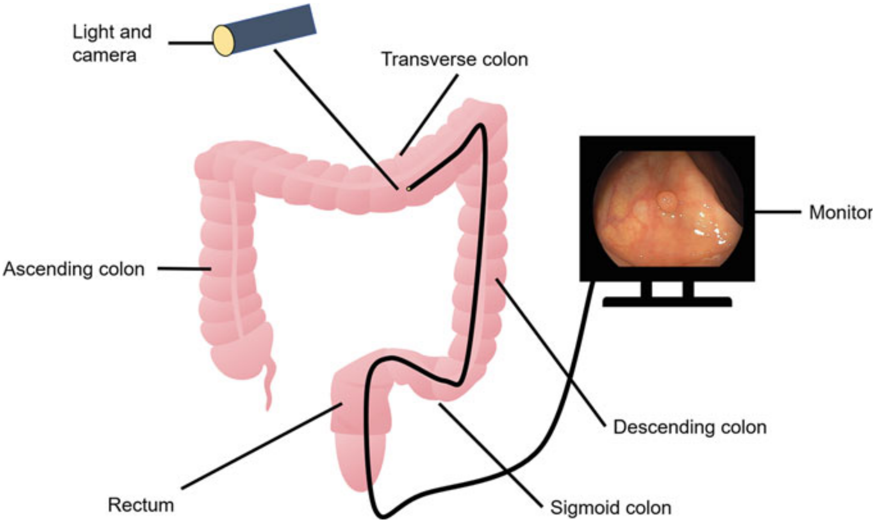


Fig. 10.3 An illustration of routine colonoscopy examination

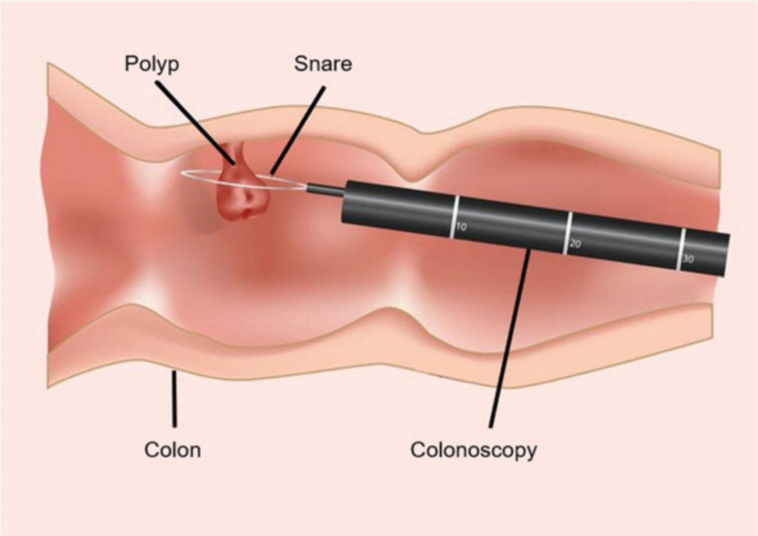


Fig. 10.4 The illustration of polypectomy

Polypectomy illustrated in Fig. 10.4 can be performed by excision using endoscopy tools. The procedure is relatively noninvasive and is usually carried out at the same time during a colonoscopy examination.

In addition, Endoscopic mucosal resection (EMR) and Endoscopic submucosal dissection (ESD) are also used for the resection of large colorectal polyps and cancer. Colon bleed is the main complication of polypectomy, EMR, and ESD. Thanks to the efforts of endoscopists and new-generation endoscopy devices, the rate of bleeding complications has been significantly reduced.

Olympus, Fujifilm, and PENTAX Medical are currently the three leading endoscope manufacturers in the world. Olympus is the inventor of endoscope and has maintained its leading position in the industry since the 1980s. Endoscopes integrate traditional optics, ergonomics, precision machinery, modern electronics, mathematics, software, and other multidisciplinary testing equipment. Classical endoscopes usually have two optical fibers. The "optical fiber" carries light into the body cavity, and the "imaging fiber" carries the image of the body cavity back to the endoscopists' sight glass. There is also a separate port that allows dosing, suction, and flushing. The port as shown in Fig. 10.5 can also be used to introduce small folding tools such as forceps, scissors, brushes, snares, and baskets for tissue resection (removal), sampling, or other diagnostic and therapeutic work. The endoscope can be used with a camera or video recorder. More advanced endoscopes are equipped with digital functions for recording, processing, and enhancing video images. The endoscopes usually have good flexibility and convenient operation performance. It can reach narrow or curved space in colons, and even small intestines. The image transmission sub-objective, the relay system and the eyepiece are used together to

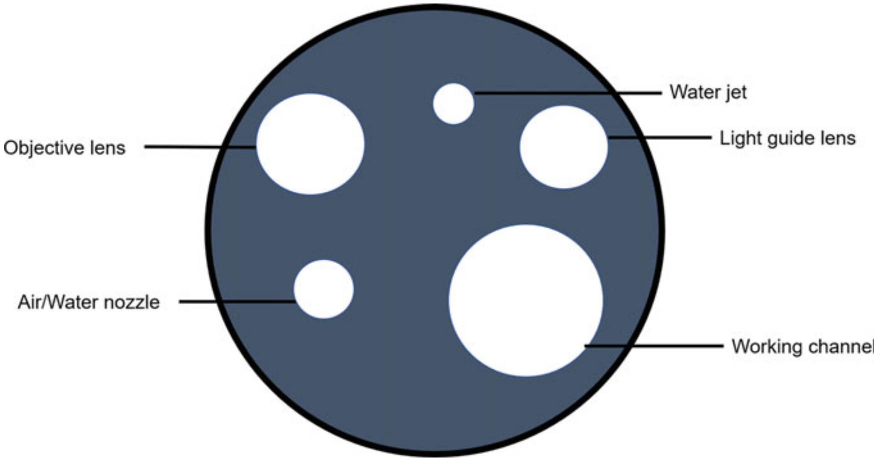


Fig. 10.5 Overview of the distal tip of endoscopy

transmit images. The illumination part transfers halogen, xenon, LED, or laser lights via optical fibers into colon. The main function of stomata is to supply air, water, and pass unyielding biopsy forceps. Currently, multifunctional high-definition electronic endoscopes have mainly being used in clinical medicine. They can not only obtain diagnostic information of tissues and organs but also measure various physiological functions of tissues and organs. Image processing technology or special light can also be used to obtain special module images of lesions, and image analysis technology can be used to achieve quantitative analysis and diagnosis of colon diseases. For example, narrow-band imaging (NBI) is proposed by Olympus Corp., and blue light imaging (or blue laser imaging) and linked color imaging (LCI) are employed by Fujifilm Corp. for the analysis of colorectal lesions.

10.1.4 Application of AI in Colonoscopy

With the progress of deep learning, artificial intelligence is currently referred to deep learning. However, AI actually has a broader meaning than deep learning. Figure 10.6 illustrates the relationship between AI, machine learning, and deep learning. AI was proposed by McCarthy et al. at the 1956 Dartmouth Conference. AI has shown great potentials in many aspects after decades of development. Humans are dreaming of realizing a super-complex computer, thinking like human beings, and having similarities even beyond human intelligence. Machine learning is a type of artificial intelligence that allows systems to automatically learn from data and improve performance without prediction rules being explicitly programmed. Deep neural network was inspired by the biological properties of the human brain, particularly neuronal connections. Deep learning is a type of machine learning to learn data using a hierarchical structure similar to brain neurons.

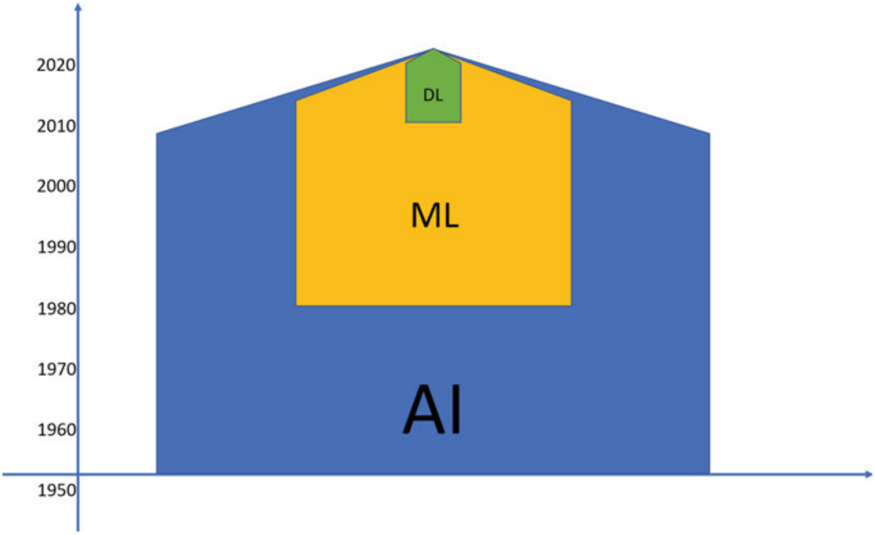


Fig. 10.6 Overview of the artificial intelligence

However, traditional AI was far away from being as intelligent as what humans expect, but more like a robot carrying certain tasks followed commands. Careful designed features and rules should be employed to get satisfying performance for detection and classification based on traditional AI. In contrast, deep learning is able to learn features automatically from training datasets instead of using handcrafted features. Deep learning, has also shown its potential in many aspects, including but not limited to image classification/recognition, natural language processing, machine translation, autonomous driving. Some unilateral capabilities such as image recognition, language analysis, and board games have reached a level surpassing humans. For example, in March 2016, AlphaGo (AI of board game Go) defeated a professional player (former World Go champion) in Korea. This incident has made people realize that AI is a reality that would develop into science fiction scenes seen in movies and novels. At the same time, the development of graphics processing unit provides a low-cost, fast, and powerful parallel processing environment for the study of AI.

With the implementation of deep learning in image procession, AI is also adopted in medical image-based diagnoses and has gained unprecedented interest and success, such as pathology, radiology, and endoscopy in the medical field. The availability of enhanced computing power in graphics processing units (GPU), and access to large datasets also accelerate the progress of this field. For example, a convolutional neural network (CNN) trained on 129,450 skin images is comparable to the performance of expert dermatologists in distinguishing benign and malignant lesions [19]. In another study, the accuracy of a deep neural network algorithm was similar to those of expert pathologists in evaluating the presence of breast cancer metastasis in sentinel axillary lymph node specimens in 129 pathological slice images [20]. When performing limited-time exercises that simulate real-life clinical practice, the best-performing algorithm can achieve better diagnostic performance than 11 pathologists.

Computer-aided diagnosis system is a kind of designed AI to assist clinicians to interpret medical images. AI is applied in gastrointestinal endoscopy in a good time, particularly on the diagnosis and detection of colorectal polyps in recent years. Many studies have shown the feasibility of using AI to classify CRCs, and the results are encouraging. Hence, AI can be considered as a virtual second observer to assist diagnose CRCs, and it can be of great help to endoscopists under CRCs, particularly for the less-experienced endoscopists on the detection and diagnosis of CRCs.

10.2 Computer-Aided Detection in Colorectal Polyps

In the previous section, we mainly summarize CRCs and related knowledge, and we are acquainted that the resection of polyps, the precursors of CRCs, can effectively prevent the progress of CRCs [9]. In this section, we address how to perform polyp detection by using computer-aided detection (CADe) system assistants endoscopists resection.

10.2.1 Why Computer-Aided Detection

As mentioned in the previous section, colonoscopy is the gold standard for the detection of polyp. However, the efficacy of colonoscopy depends on endoscopists' techniques and their expertise accumulated over their careers. Endoscopists usually have various proficient skills. As a result, the adenoma detection rate (ADR) showed a wide variation among them, consequently causing an adenoma miss rate of 6–27% found in clinical trials [21]. In the last two decades, techniques to increase ADR and decrease the adenoma miss rate have been studied and developed. Evidence has demonstrated the usefulness of image-enhanced endoscopy, optimal observation time and various distal attachments have provided colonoscopists with tools to increase the ADR [21, 22].

Corley et al. found each 1.0% increase of ADR would reduce 3.0% of the risk of cancer [24, 25]. However, in a systematic review study, the miss rate with colonoscopy was 22% for polyps, especially 26% for diminutive adenomas (1–5 mm) [26]. Substantial evidence suggests that colonoscopy screening is less effective for right-sided than left-sided CRCs. The main reasons include poor bowel preparation, the narrow inspection field of view, diminutive polyps, and less-experienced endoscopists. More recently, a research indicated an 8.6% post-colonoscopy CRCs rate within 3 years of colonoscopy [27]. Evidence suggests that these may be caused by missed cancer diagnosis or incomplete resection of the lesions [28].

Many devices and optical strategies have been proposed in colonoscopy screening to improve ADR. Various physical distal colonoscope attachments expand viewing areas to facilitate the detection; however, a meta-analysis indicates only small improvements in ADR. Advanced image-enhanced technologies such as NBI, BLI, and LCI increase ADR by enhancing the color of polyp, but studies have failed to demonstrate conclusive increments in ADR during colonoscopy using these new imaging modules.

Endoscopists might fail to detect lesions even when polyps are within the endoscopic field of view. That is why some studies have shown that nurses participate in colonoscopy as a second observer can improve the ADR [29]. However, this requires human resource consumption, and the detection rate of adenoma has not been significantly improved. Specific visual cues might alert endoscopists achieving high levels of detection to subtle lesions that could otherwise be overlooked

10.2.2 Early Computer-Aided Detection Algorithm

Generally, there are four modalities of endoscopic images. (1) Standard white light endoscope images are used in routine screening of CRCs. Researchers also focus on analyzing polyp appearance on images from standard endoscopes since our aim is to assist the endoscopist to find more polyps during routine screening; (2) Magnifying endoscope images are usually used by endoscopists to clinically view the characteristics of polyps, but are not commonly used for the research of polyp detection; (3) NBI is an enhanced imaging method, with different color ranges from those of standard endoscopes. The concave patterns of polyps can be clearly observed in the magnified endoscope. They are used in areas where the endoscopist suspects abnormalities, not typically seen using standard scopes; (4) Wireless capsule endoscope images are generally regarded as poor image quality and difficult to obtain. We excluded CADe algorithms with narrow-band imaging and chromoendoscopy because these methods were usually used after detecting the polyps.

In Table 10.2, we summarized the main features of early methods in polyp detection. Early computer-aided polyp detection focused on techniques guided by polyp characteristics, such as color, shape, or texture. However, since there is no open database available to test different methods, it is difficult to compare them. Therefore, we can only compare different methods based on certain metrics (such as precision, recall, and F1), always taking into account that these metrics are calculated based on private datasets.

An efficient CADe algorithm may be able to assist colonoscopists in finding suspicious lesions, thus increasing the ADR and decreasing the adenoma miss rate. In our previous study using still images [30], the sensitivity of CADe for neoplastic lesions was 99.0% (194/196) and the specificity was 98.2% (110/112), showing excellent results. However, these CADe algorithms have a significant drawback because they only work on still images, instead of videos needed by clinical practice. For the detection of colorectal polyps in videos, we require a real-time CADe system with high sensitivity and specificity, and a low false positive rate with nearly no time delay. The CADe system can be useful as a second observer in clinical practice if satisfies

Table 10.2 Early computer-aided detection based on machine learning

	Dataset	Techniques	Feature class
Krishnan et al.	2 normal and 4 abnormal images	Edge detection to extract contours, curvature analysis	Shape
Maroulis et al.		Wavelet transform-based statistical features	Texture
S. A. Karkanis et al.	66 videos	Color wavelet, covariance texture features	Texture, color
Tjoa et al.	12 normal and 54 abnormal images	Texture spectrum color histograms	Texture, color
Li et al.	12 normal and 46 abnormal images	Wavelet coefficients, lab histograms	Texture, color
Coimbra et al.	899 images	MPEG-7 descriptors; shape descriptor; edge histogram; scalable color	Shape, texture
B. V. Dhandra et al.	50 normal and 50 abnormal images	Combination of region color, texture, and edges	Shape, color
Hwang et al.	27 polyp shots	Curve direction, curvature, edge distance, intensity	Shape, color
D. C. Cheng et al.		Region texture	Texture
J. Bernal et al.	300 images from 15 colonoscopy videos	Shape-descriptors	Shape
S. Y. Park et al.	35 colonoscopy videos	Spatial and temporal features	Shape, color
Wang et al.	42 images	Shape-descriptors	Shape
Guo et al.	299 abnormal and 255 normal images	Texture spectrum, color histograms	Texture, color

the above requirements. Apparently, the early CADe algorithms do not satisfy the requirement for a CADe system.

10.2.3 Recent Computer-Aided Detection Algorithms

Recently, due to the development of algorithms, the availability of enhanced computing power in graphics processing units, and access to large datasets, i.e. big data, deep learning-based methods are now gaining unprecedented interest and success in medical image analysis. Particularly, deep learning-based algorithms using CNN (Table 2.2) showed powerful polyp detection capabilities and are expected to be used for CRCs screening in the near future. In Oct. 2020, Olympus starts selling ENDO-AID CADe, an AI-aided diagnosis system for colonoscopy(https://www.olympus.co.jp/news/2020/nr01856.html). Fujifilm would sell its CAD EYE^{TM}, a computer-aided function for the diagnosis using endoscopy on Nov. 30th, 2020(https://www.fujifilm.com/jp/ja/news/list/5581).

Misawa et al. [31] developed an algorithm based on a convolutional three-dimensional neural network to detect polyps from colonoscopic videos and achieved a per-frame sensitivity of 90.0%, per-frame specificity of 63.3% and the accuracy was 76.5% on 135 short videos. The area under the receiver operating characteristic curve (AUC) was 0.87. Convolutional three-dimensional neural networks, designed for processing spatiotemporal data especially suitable for video data, differ from previous CNN. This preliminary study used a dataset consisting of 73 colonoscopy video sequences with 155 polyps removed from the cecal intubation to the anus and 391 short videos without polyps. Flat lesions accounted for 64.5% of the dataset. Each frame containing polyps was retrospectively annotated by two endoscopy experts as a reference for the existence of polyps. The dataset is randomly separated to train and test CNN. According to the analysis of receiver operating characteristics (ROC), the detection probability of polyps is set to a cut-off value of 15%.

Urban et al. [32] developed a deep learning algorithm and tested it on 20 full colonoscopy videos with a false positive rate of 7%. The cross-validation accuracy of polyps detected by CNN is 96.4%, and the AUC is 0.991. Two colonoscopy video datasets were further analyzed, and three colonoscope experts (ADR $\geq$ 50%) identified frames containing polyps without the help of CNN. A senior expert (ADR $\geq$ 50% and >20,000 colonoscopy) also reviewed the video with CNN's detection results, i.e., green boxes superimpose on detected polyps with a confidence value greater than 95%. Three experts (without using CNN) reviewed intact videos and found a totally 36 polyps. With the aid of CNN, senior experts identified a total of 45 polyps. Among the other nine polyps found with the assistance of CNN, the confidence values determined by senior experts were lower for six and higher for three polyps. The second dataset included 11 videos with 73 polyps. These videos were recorded when the colonoscopy expert did not move the colonoscope closer to identify polyps during withdrawal to purposely simulate the procedures of missed polyps. CNN identified 67 polyps in 73 cases, with a false positive rate of 5% frame by frame. The results of this feasibility study support the concept that CNN's assistance can improve ADR by showing polyps that may be missed. Urban and his colleagues evaluated the proposed CNN using a dataset of 8641 hand-selected colonoscopy images, which are consisted of more than 40 images with polyps and 4,553 polyp-free images from more than 2,000 patients. A group of colonoscopists annotated images with polyps. Senior experts set a confidence level (high or low) for polyps as references. The first dataset consists of nine videos where colonoscopists removed 28 polyps.

Wang et al. [33] proposed a polyp detection algorithm based on SegNet using a deep learning algorithm [34] to real-time detect polyps in clinical colonoscopy with high sensitivity and specificity. They used data from 1,290 patients to train the model, and validate it using 27,113 colonoscopy images from 1,138 patients. They achieved a sensitivity of 94.38%, specificity of 95.92% and AUC is 0.984. Furthermore, on 138 colonoscopy videos with polyps, the algorithm realizes per-polyp sensitivity of 100%, and on 54 colonoscopy videos without polyps achieved per-frame of specificity of 95.4%. Besides, in a prospective randomized controlled clinical trial [35], the same group obtained a significant difference in ADR (CADe 29% vs. control 20%) on 1058 patients. 536 and 522 patients were randomized to standard colonoscopy and colonoscopy with CADe, respectively. The AI system significantly increased ADR (29.1%vs20.3%) and the mean number of adenomas per patient (0.53vs0.31, $p < 0.001$). This difference is mainly caused by the numbers of detected diminutive adenomas (185 vs 102), while there was no statistical difference in larger adenomas (77vs 58). In the following double-blind randomized study, the ADR was significantly greater in the CADe group than that in the control group.

Through the collaboration with Aizu Medical Center, Fukushima Medical University, we started to perform the development of a polyp detection algorithm based on deep learning to remind endoscopists of undiscovered lesions. We focused on the detection of diminutive polyps (no more than 5 mm) and small polyps (no more than 10 mm but larger than 5 mm). These two kinds of polyps are the mostly missed polyps in colonoscopy examinations but may have higher potential for malignancy, such as adenomous polyps [36, 18]. Through finetuning a CADe system based on YoloV3, we achieved a per-video sensitivity of 88% for polyp detection and a per-frame false positive rate of 2.8% using a confidence level of $\geq$30% [37]. Four endoscopists performed detection using the same validation dataset. The per-video sensitivity of two senior endoscopists was 88%, and the sensitivities of the two physicians in training were 84% and 76%, respectively. For endoscopists, the numbers of frames with missed polyps appearing on short videos were significantly less than those of the frames with detected polyps. No trends were observed regarding polyp sizes, morphology, or colors. Furthermore, for full video readings, the CADe algorithm achieved a per-polyp sensitivity of 100%, a per-frame false positive rate of 1.7%, and per-frame specificity of 98.3%.

In order to validate various polyp detection algorithms, it is necessary to build a balanced and open database with good image quality. Medical Imaging Computing and Computer-Assisted Intervention 2015 conference provided an open database for automated polyp detection and sought to define performance metrics on publicly annotated databases to enable comparisons of multiple methods. The quality of the database should be improved for comparing various methods. Despite this, the results from the challenge competition also showed that CNN-based methods were state of the art, and a combination of methods led to improved performance.

CADe based polyp detection systems, especially those using deep learning technology, have broad prospects in providing clinicians with real-time support and may reduce human behavior differences. However, most methods are validated on small private datasets composed of high-quality still images, polyp morphology lacks variability, and obtained from devices made by the same manufacturer. In addition, few studies have explored the potential impact of other quality variables (such as bowel preparation and withdrawal time) on performance. The detection performance of CADe polyps may vary depending on the characteristics of the operator (such as withdrawal speed and inspection quality). Furthermore, analysis and detection using AI may significantly increase examination durations because more polyps may be found, and physicians may have to confirm the results of AI detection if there are too many false negatives.

To overcome these challenges and develop large-scale annotated datasets with more easily available consistent performance evaluation metrics, the collaboration between clinicians and computer scientists is necessary (Table 10.3).

10.3 Computer-Aided Classification in Colorectal Polyps

With the continuous improvement in the quality of endoscopic imaging systems, optical diagnosis is increasingly used on histology prediction of colorectal polyps, such as the presence of neoplastic lesions or even submucosal invasion instead of using pathological images. This information could be a great benefit in determining endoscopic removable polyps or superficial cancer. In this section, we mainly introduce the computer-aided diagnosis for polyp classification in clinical optical diagnosis and related methods. The reason why we need computer-aided diagnosis will be introduced in 3.1. for the early and advanced methods of computer-aided classification are introduced at 3.2 and 3.3.

10.3.1 Why Computer-Aided Classification

CRCs limited to mucosa and with superficial submucosal invasion without unfavorable histology have lower risks of lymph node metastases and could be removed by endoscopic therapy [38–40]. In contrast, CRCs with deep ($\geq$1 mm) submucosal invasion have a higher risk of metastases. Recurrent lesions may develop after endoscopically removing deeply invasive CRCs, especially those with unfavorable histology [40]. Therefore, it is imperative to discriminate submucosal invasion stage CRCs from superficially invasive ones. However, endoscopic images of submucosal invasion stage CRCs resemble those of mucosa/superficial submucosal invasion stage CRCs [41, 42], and colonoscopists frequently have difficulty in visual classification.

Clinical optical diagnosis of polyp histology, especially for the diminutive polyps (<=5 mm), can potentially result in enormous cost savings in ways of the “resect and discard” strategy for diminutive adenoma polyps and the “do not resect” strategy for diminutive hyperplastic polyps in the distal colon. The formal strategy using virtual chromoendoscopy assistant endoscopists to confirm resect polyps [43]. The later strategy suggests polyps could be left if optically characterized as non-neoplastic. Studies have also shown that using the NBI diagnosis of diminutive polyp histology, experts can accurately define successive surveillance colonoscopy intervals. If the histology of diminutive polyps can be characterized by endoscopists using colonoscopy, cost burden may be alleviated. However, chromoendoscopy has not been adopted in routine clinical practice especially in the west as it is perceived to

Table 10.3 Recent computer-aided detection based on deep learning

	Training dataset	Test dataset	Methods	CADe's results	Endoscopists's results	Real-time
Misawa et al.	Total 73 colnonscopy withdrwal videos;		convolutional neural network	90.0% per-frame sensitivity; 63.3% per-frame specificity; 2.8% per-frame false-positive rate; 76.5 accuracy; ROC-AUC 0.87;	-------	-------
	105 positive colonoscopy short videos; 306 negative colonoscopy short videos	50 positive colonoscopy videos; 85 negative colonoscopy videos				
Urban et al.	8641 frames (4088 polyp and 4553 non-polyp); Video dataset 1: 9 videos (containing 28 polyps removed by original colonoscopists); video dataset 2: 11 videos simulating missed polyp scenarios (containing 73 polyps)		convolutional neural network	96.4% cross validation accuracy; ROC-AUC 0.991; Video dataset 1: CNN overlay led to identification of 45 polyps (additional 9 polyps over experts not using CNN assistance); video dataset 2: 67 of 73 polyps identified with frame-by-frame false positive rate of 5% (CNN trained on 8641 images and fine-tuned with video dataset 1)	-------	98 fps
Wang et al.	1,290 colonoscopy videos;	27,113 colonoscopy images from 1,138 patients; public database of 612 polyp-containing images; 138 positive colonoscopy videos; 54 negative colnoscopy videos;	convolutional neural network	94.38% per-image sensitivity, 95.92% per-image specificity, ROC-AUC 0.984 on images; 88.24% per-image sensitivity on public database; 91.64% per-image-sensitivity, 100% per-polyp sensitivity, 95.4% per-image specificity on videos	-------	>25 fps; 76.80 ± 5.60ms
		1,058 patients, 522 with CADe, 536 standard		29.1% have one or more polyps	20% have one or more polyps	
		962 patients, 484 with CADe, 478 with sham system		34% have one or more polyps	28% have one or more polyps	
Su et al.	3,350 colonoscopy images	751 colonoscopy images	convolutional neural network	Acc95.47%, sen96.04%, spe94.46% ROC-AUC 0.99		25 fps
		623 patients, 308 with CADe, 315 with standard colonoscopy		28.9% have one or more polyps	16.5% have one or more polyps	
Guo et al.	1991 colonoscopy images from 283 patients	100 colonoscopy video clips and 15 full videos	convolutional neural network	100% per-polyp sensitivity; 98.3 per-frame specificity; 1.7% per-frame false-positive;	per-video sensitivity of both experts was 88%, and the sensitivities of the two physicians in training were 84% and 76%	50.2 ± 6.9 fps

be cumbersome and impractical, requires extra time (for dye spraying), and increases the cost of the procedure (dye, spray catheter).

Polyp characterization during colonoscopy is highly limited in view of endoscopists' knowledge and experiences. More importantly, according to the reports by Preservation and Incorporation of Valuable Endoscopic Innovations of the American Society for Gastrointestinal Endoscopy, that recommends endoscopists require intensive training on image enhanced endoscopy that achieves a negative predictive value of >90% in predicting the absence of adenomatous histology. The proposed "resect and discard" approach for diminutive colorectal polyps relies heavily on the accuracy of prediction of colorectal polyp histology. The standard of "diagnose and leave" strategy can be implemented in academic settings, but not always in community practice, which limits the time it can be incorporated into routine care [44, 45]. The performance difference between experts and non-experts and other reasons make it extremely important for clinically accurate and stable uniform inspection standards.

The subtle lesions that may be overlooked are recognized by AI systems because of specific features, and visual clues may remind endoscopists to perform detailed analysis to identify cancer. And many studies have shown AI systems have the potential ability for histology recognition. It could improve non-expects' diagnostic accuracies.

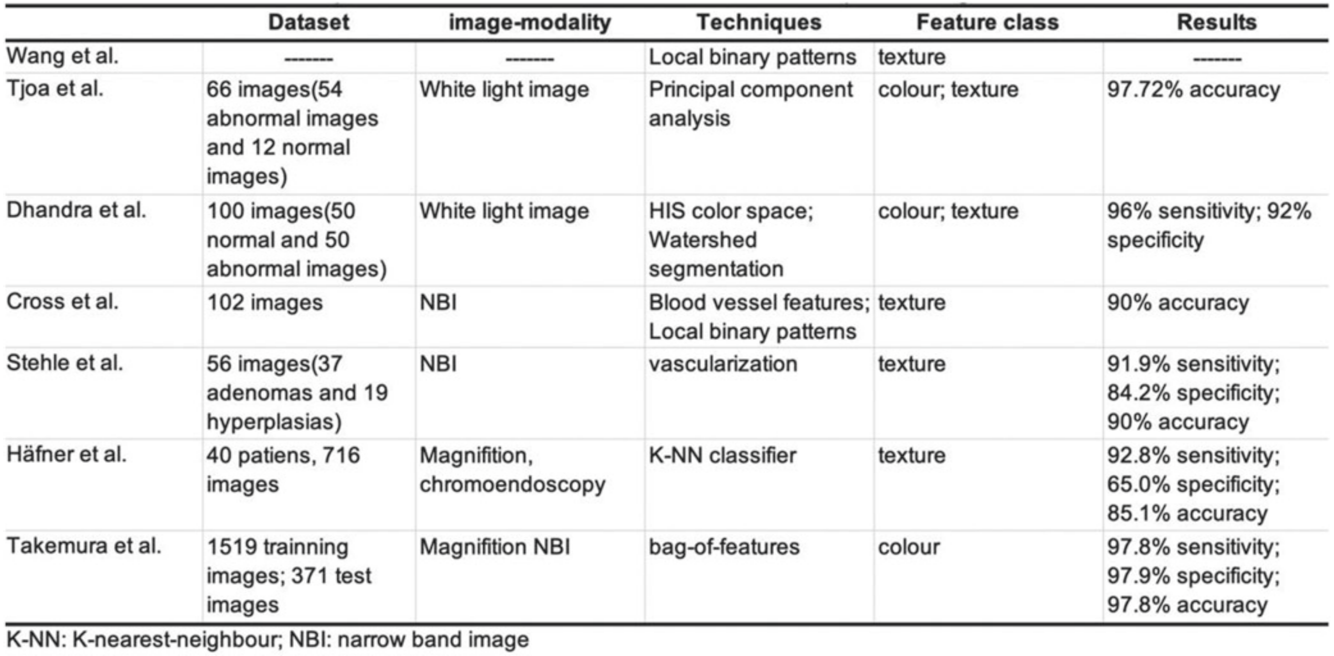
10.3.2 Early Computer-Aided Analysis (CADx)

Early image-based CADx systems are based on traditional computer vision and machine learning methods as described in Table 3.1. These methods require researchers to manually select meaningful image features, design algorithms to extract and pack features and construct a trainable predictive algorithm classifier. Since the early computer-aided classification system has inherent defects such as poor practicability, poor effect, inconvenient operation, and slow running speed, we don't provide further information here (Table 10.4).

10.3.3 Recent Progress of CADx

Deep learning developed recently overcomes the above issues by automatic feature extraction for better-representing data in a trainable manner. To date, the best results can be obtained using CNN. CNN is composed of multiple layers of simple computing nodes, but with complex connections that can mimic the behaviors of human visual cortexes. Therefore, CNN can continuously learn high-level features. CAD with the recent progress of image-enhanced endoscopy incorporating AI especially deep learning technologies (summarized in Table 3.2), offers a promising

Table 10.4 Early computer-aided analysis based on traditional machine learning



	Dataset	image-modality	Techniques	Feature class	Results
Wang et al.	-------	-------	Local binary patterns	texture	-------
Tjoa et al.	66 images(54 abnormal images and 12 normal images)	White light image	Principal component analysis	colour; texture	97.72% accuracy
Dhandra et al.	100 images(50 normal and 50 abnormal images)	White light image	HIS color space; Watershed segmentation	colour; texture	96% sensitivity; 92% specificity
Cross et al.	102 images	NBI	Blood vessel features; Local binary patterns	texture	90% accuracy

Stehle et al.	56 images(37 adenomas and 19 hyperplasias)	NBI	vascularization	texture	91.9% sensitivity;
					84.2% specificity;
					90% accuracy

Häfner et al.	40 patiens, 716 images	Magnifition, chromoendoscopy	K-NN classifier	texture	92.8% sensitivity;
					65.0% specificity;
					85.1% accuracy

Takemura et al.	1519 trainning images; 371 test images	Magnifition NBI	bag-of-features	colour	97.8% sensitivity;
					97.9% specificity;
					97.8% accuracy

K-NN: K-nearest-neighbour; NBI: narrow band image

Table 10.5 Recent computer-aided analysis based on deep learning

	Training dataset	Test dataset	image-modality	CAD results	Experts results
Gross et al.	-------	214 patients, 434 polyps	Magnifition NBI	95.0% sensitivity; 90.3% specificity; 93.1% accuracy	-------
Kominami et al.	-------	41 patients, 118 polyps	Magnifition NBI	95.9% sensitivity; 93.3% specificity; 94.9% accuracy	-------
Tamai et al.	-------	121 polyps	Magnifition NBI	83.9% sensitivity; 82.6% specificity; 82.8% accuracy	70.0% sensitivity; 96.1% specificity; 92.6% accuracy
Byrne et al.	-------	125 polyps	Magnifition NBI	98% sensitivity; 83% specificity; 94% accuracy	-------
Chen et al.	-------	193 patients, 284 polyps	Magnifition NBI	96.3% sensitivity; 78.1% specificity; 90.1% accuracy	97.3% sensitivity; 77.1% specificity; 90.5% accuracy
Mori et al.	-------	152 patients, 176 polyps	Endocytoscopy	92.0% sensitivity; 79.5% specificity; 89.2% accuracy	-------
Mori et al.	-------	123 patients, 205 polyps	Endocytoscopy	89% sensitivity; 88% specificity; 89% accuracy	91% sensitivity; 89% specificity; 90% accuracy
Misawa et al.	-------	100 polyps	Endocytoscopy	84.5% sensitivity; 97.6% specificity; 90.0% accuracy	-------
Takeda et al.	5543 images	200 images	Endocytoscopy	98.1% sensitivity; 100% specificity; 99.3% accuracy	-------
Mori et al.	-------	325 patients, 466 polyps	Endocytoscopy, NBI	93.3% sensitivity; 91.0% specificity; 92.7% accuracy	-------
Maeda et al.	12900 images	525 lesions	Endocytoscopy	74% sensitivity; 97% specificity; 91% accuracy	-------
Komeda et al.	-------	10 polyps	White light endoscopy, NBI, chromoendoscopy	70% accuracy	-------
Tokunaga et al.	2751 images	691 images	White light endoscopy	96.7% sensitivity; 75.0% specificity; 90.3% accuracy	96.5% sensitivity; 72.5% specificity; 89.4% accuracy
Nakajima et al.	313 lesions, 1839 images	44 lesions, 58 images	White light endoscopy	81% sensitivity; 87% specificity; 84% accuracy	86% sensitivity; 100% specificity; 93% accuracy

solution to help endoscopists by providing decision support during colonoscopy examinations (Table 10.5).

10.3.3.1 Narrow Band Image-Based Classification

NBI is a technology using rotating filters in front of light sources to filter the bandwidth of the projected light centered to 30 nm wide spectra of 415 nm (blue) and 549 nm (green) to create pseudo-colored images.

Tischendorf et al. first evaluated computer-aided classification of colorectal polyps using NBI magnification images to classify neoplastic and non-neoplastic [46]. Then, they improved the computer-aided system in another prospective study where they analyzed 434 polyps of no more than 10 mm from 214 patients [47]. They evaluated nine classification features and achieved a sensitivity, specificity, and accuracy of

95%, 90.3%, and 93.1%, respectively. These were similar to those of experts and superior to those of non-experts.

Tamai et al. developed a novel method based on AI systems to classify hyperplastic polyps, adenoma/adenocarcinoma lesions, and deep submucosal lesions using magnified NBI [48]. On a test using 121 reviewed lesions, their AI system demonstrates similar performance to that of experienced endoscopists. The sensitivity, specificity, positive and negative predictive values, and accuracy of the AI system for neoplastic lesions were 83.9%, 82.6%, 53.1%, 95.6%, and 82.8%, respectively. The values for deep submucosal lesions were 83.9%, 82.6%, 53.1%, 95.6%, and 82.8%, respectively.

Byrne et al. developed an AI model to differentiate hyperplastic polyps and adenomatous from diminutive adenomas for real-time assessment [49]. The polyps were detected in the normal mode and then viewed in the magnified mode before being resected. The AI model uses NBI frames with a mixture of normal focus and near focus for training and testing. The training data consisted of 223 polyp videos in NBI. Further validation was performed using 40 videos. The final test set consisted of 125 consecutively identified diminutive polyps (74 adenomas and 51 hyperplastic polyps). The AI model can process videos for 50 ms per frame. A probability score was calculated together with the classification according to NICE criteria. The model did not build enough confidence to predict the histology in 19 polyps, but for the remaining 106 polyp videos the model has an overall accuracy, sensitivity, specificity, PPV, and NPV of 94%, 98%, 83%, 90%, and 97%, respectively. They expect to conduct clinical trials to further evaluate the algorithm.

More recently, Chen et al. developed a deep neural network system to characterize diminutive polyps using magnification NBI images [50]. Images from 1476 neoplastic polyps and 681 hyperplastic polyps were collected for training their deep AI system. Regions of interest were manually selected from high-quality images by two endoscopists. The deep AI system was tested on 96 hyperplastic and 188 neoplastic diminutive polyp images collected in the same way as the training data. Histological results were used as the reference standard. The deep AI system differentiated neoplastic from hyperplastic polyp images with a sensitivity of 96.3%, a specificity of 78.1%, an accuracy of 90.1%. The performance of the deep AI system was superior to that of four novice endoscopists (<1 year of colonoscopy experience) and non-inferior when compared with two experts. The deep AI system provided a diagnosis in 0.45 s, which was shorter than the time taken by endoscopists (expert group 1.54 s and non-experts 1.77 s).

10.3.3.2 Endocytoscopy-Based Classification

Although the magnified NBI based AI system has provided a powerful tool for the analysis of diminutive polyps, the generalizability of this system is limited because magnification colonoscopy is not accessible at most endoscopy centers worldwide. A more powerful and advanced magnified endoscopy, endocytoscopy, with a hundreds-fold ultra-magnifying contact microscope for visualization of nuclei in vivo, has been developed for the prediction of histologic severity. Kudo and his team conducted a series of clinical research to verify the performance of endocytoscopy with an AI system [51–56].In a recent study on ulcerative colitis, 525 validation data of 525 independent segments were collected from 100 patients, and 12,900 endocytoscopy images from the 87 patients were used to construct their AI system. The AI system provided a diagnostic sensitivity, specificity, and accuracy of 74%, 97%, and 91%, respectively.

10.3.3.3 White Light Image-Based Classification

Generally, the visual recognition of deeply invasive colorectal cancer requires expert experience and is very difficult for most endoscopists. Tokunaga et al. extracted 3442 white light non-magnified images from 1035 consecutive colorectal lesions to develop and validate a computer-aided diagnosis system diagnosis for the prediction of invasion depth [57]. They evaluated the performance of the proposed system to distinguish shallowly and deeply invasive lesions by comparing with the results of two experts and two trainees. The artificial system distinguished deeply invasive lesions with a sensitivity, specificity, and accuracy of 96.7%, 75.0%, and 90.3%, respectively. The results were significantly higher than trainees (92.1%, 67.6%, and 84.9%, respectively) and were compared to experts (96.5%, 72.5%, and 89.4%, respectively). This study also suggests trainees assisted by AI system demonstrated a diagnostic capability comparable to those of experts. This may indicate a potential role of CADx to predict invasion depth of CRCs.

Recently, Nakajima et al. proposed a computer-aided analysis system (CADx) using only white light non-magnified plain endoscopic images to classify CRCs with deep submucosal invasion and mucosal/superficial submucosal invasion, and compared the performance of CADx with those of six endoscopists [58]. This system was trained by 1839 images from two medical institutions and tested by 44 CRCs of 78 images for the third one; it achieved a sensitivity, specificity, and accuracy of 81%, 87%, and 84%, respectively. The specificity (primary outcome) was 100% and 96% for two experts, 61% for two gastroenterology trainees, and 48% and 22% for two novices. The specificity of the proposed system was superior to novices and gastroenterology trainees but slightly inferior to experts.

10.3.4 Limitations of CADx

Unlike CADe systems, the performance of CADx is far from satisfying. The reasons may be as follows,

(1) Data imbalance because of few cases in some subtypes. For example, sessile serrated lesions (SSL) are also precursors as well as adenoma polyps; however, the incidence of SSL is much less than adenoma polyps. As a result, it is very hard

to collect enough cases for the classification of SSL from other polyps. Recently, oversampling and focal loss may help lessen the influence caused by data imbalance.

(2) High sensitivity and low specificity for malignant lesions. Physicians usually try to maintain lower sensitivity but higher specificity for malignant colon lesions because further surgical operations could be performed even if deeply invasive cancer is regarded as mild lesions and resected by endoscopy in advance. However, AI systems usually maintain a high sensitivity and low specificity to avoid missing malignant lesions. This issue may be resolved by finding a feasible threshold using ROC curves.

10.4 Conclusion

Colonoscopy is the gold standard for CRC screening. Polypectomy, EMR, and ESD have been the most common treatment therapeutic procedures to prevent the progression of polyps to CRCs. To date, it has not been fully verified whether AI can significantly improve medical performance, reduce medical costs, and increase the satisfaction of patients and medical staff who use AI in the medical field. In addition, even if the same AI is used, treatment results may vary depending on the indications and scope of application. Moreover, it is difficult to prove the efficacy of AI through clinical trials and establish guidelines for the application of AI. Therefore, it is necessary to prove the clinical efficacy of AI based on more prospective trials, develop an AI-friendly endoscopists interface, and train endoscopists to use AI. AI would help endoscopists but endoscopists are responsible for admitting the detection and classification results of AI. When physicians use AI to interpret endoscopic images, AI would help doctors reduce misses in the interpretation of endoscopic images and improve medical efficiency. Therefore, endoscopists should seek the best way to provide better care for their patients with the help of AI. However, it is necessary to reach a social consensus on forensic issues and educate future AI physicians, because AI may cause legal and ethical issues.

AI based diagnosis could make optical diagnosis widely applicable but is currently in its early developmental stage. The automatic detection of polyps based on deep learning or the analysis of endoscopy images of CRCs should be further validated through clinical trials. In the next few decades, CNN-based automatic endoscopy diagnosis is expected to become the mainstream diagnostic technology, and deep learning is expected to help endoscopists provide a more accurate diagnosis by automatically detecting and classifying endoscopic lesions. One of the most important factors in the development of endoscopy would be the availability of a large number of high-quality endoscopy images/videos, and endoscopists' enthusiasm, knowledge, and experiences AI technology. Many endoscopists are focusing on this new technology and trying to promote progress on CRCs diagnosis.

References

1. Torre LA, Bray F, Siegel RL, Ferlay J, Lortet-tieulent J, Jemal A (2015) Global cancer statistics, 2012. CA Cancer J Clin 65:87–108
2. Siegel R, DeSantis C, Jemal A (2014). Colorectal cancer statistics, 2014. CA: Cancer J Clinicians 64(2):104–117
3. Chen W, Zheng R, Baade PD, Zhang S, Zeng H, Bray F, He J (2016). Cancer statistics in China, 2015. CA: Cancer J clinicians 66(2):115–132
4. Pickhardt PJ (2016) Emerging stool-based and blood-based non-invasive DNA tests for colorectal cancer screening: the importance of cancer prevention in addition to cancer detection. Abdominal Radiol 41(8):1441–1444
5. Cancer statistic of Japan in 2020. [Online]. Available: http://ganjoho.jp/reg_stat/statistics/stat/summary.html, Oct. 14, 2020
6. Meyer JE, Narang T, Schnoll Sussman FH, Pochapin MB, Christos PJ, Sherr DL (2010) Increasing incidence of rectal cancer in patients aged younger than 40 years. Cancer 116(18):4354–4359
7. Gado A, Ebeid B, Abdelmohsen A, Axon A (2014) Colorectal cancer in Egypt is commoner in young people: Is this cause for alarm? Alexandria J Med 50(3):197–201
8. Winawer SJ, Zauber AG, Ho MN, et al (1993) Prevention of colorectal cancer by colonoscopic polypectomy. The National Polyp Study Workgroup. N Engl J Med 329:1977–81
9. Zauber AG, Winawer SJ, O'Brien MJ et al (2012) Colonoscopic polypectomy and long-term prevention of colorectal-cancer deaths. N Engl J Med 366:687–696
10. Jess T, Gamborg M, Matzen P et al (2005) Increased risk of intestinal cancer in Crohn's disease: a meta-analysis of population-based cohort studies[J]. Am J Gastroenterol 100(12):2724–2729
11. Agrawal S, Bhupinderjit A, Bhutani MS et al (2005) Colorectal cancer in african americans[J]. Am J Gastroenterol 100(3):515–523
12. Khan N, Afaq F, Mukhtar H (2010) Lifestyle as risk factor for cancer: Evidence from human studies[J]. Cancer Lett 293(2):133–143
13. Baena R, Salinas P (2015) Diet and colorectal cancer[J]. Maturitas 80(3):258–264
14. Amin M, Edge SB, Greene FL et al (2017) AJCC Cancer Staging Manual, 8th ed. Switzerland, Springer
15. Sonnenberg A, Amorosi SL, Lacey MJ et al (2008) Patterns of endoscopy in the united states: analysis of data from the centers for medicare and medicaid services and the national endoscopic database. Gastrointest Endosc 67:489–496
16. Siegel R, Desantis C, Jemal A (2014) Colorectal Cancer Statistics, 2014. CA Cancer J Clin 64:104–117
17. Gupta N, Bansal A, Rao D, Early DS, Jonnalagadda S, Wani SB, et al (2012) Prevalence of advanced histological features in diminutive and small colon polyps. Gastrointest Endosc 75(5):1022–30
18. Lieberman D, Moravec M, Holub J, Michaels L, Eisen G (2008) Polyp size and advanced histology in patients un- dergoing colonoscopy screening: implications for CT colonography. Gastroenterology 135(4):1100–1105
19. Esteva A, Kuprel B, Novoa RA et al (2017) Dermatologist-level classification of skin cancer with deep neural networks. Nature 542:115–118
20. Ehteshami Bejnordi B, Veta M, Johannes van Diest P, et al (2017) Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. JAMA 318: 2199–210
21. Ahn SB, Han DS, Bae JH et al (2012) The miss rate for colorectal adenoma determined by quality-adjusted, back-to-back colono- scopies. Gut Liv 6:64–70
22. Stoffel EM, Turgeon DK, Stockwell DH et al (2008) Chromoen- doscopy detects more adenomas than colonoscopy using intensive inspection without dye spraying. Cancer Prev Res (Phila) 1:507–13

23. Castaneda D, Popov VB, Verheyen E et al (2018) New technologies improve adenoma detection rate, adenoma miss rate, and polyp detection rate: a systematic review and meta-analysis. Gastrointest Endosc 88:209–222
24. Corley DA, Jensen CD, Marks AR et al (2014) Adenoma detection rate and risk of colorectal cancer and death. N Engl J Med 370:1298–1306
25. Kaminski MF, Regula J, Kraszewska E et al (2010) Quality indicators for colonoscopy and the risk of interval cancer. N Engl J Med 362:1795–1803
26. van Rijn JC, Reitsma JB, Stoker J et al (2006) Polyp miss rate determined by tandem colonoscopy: a systematic review. Am J Gastroenterol 101(2):343–350
27. Morris EJA, Rutter MD, Finan PJ, Thomas JD, Valori R (2015) Post-colonoscopy colorectal cancer (PCCRC) rates vary considerably depending on the method used to calculate them: a retrospective observational population-based study of PCCRC in the English National Health Service. Gut 64:1248–1256
28. Le Clercq CMC, Bouwens MWE, Rondagh EJA et al (2014) Postcolonoscopy colorectal cancers are preventable: a population-based study. Gut 63:957–963
29. Lee CK, Park DI, Lee SH et al (2011) Participation by experienced endoscopy nurses increases the detection rate of colon polyps during a screening colonoscopy: a multicenter, prospective, randomized study. Gastrointest Endosc 74:1094–1102
30. Zhu X, Wang Y, Nemoto D et al (2018) Identification of sessile serrated adenoma/polyp using convolutional neural network (Artificial Intelligence). Gastrointest Endosc 87:AB251
31. Misawa M, Kudo S, Mori Y et al (2018) Artificial intelligence- assisted polyp detection for colonoscopy: initial experience. Gastroenterology 154:2027–2029
32. Urban G, Tripathi P, Alkayali T et al (2018) Deep learning localizes and identifies polyps in real time with 96% accuracy in screening colonoscopy. Gastroenterology 155:1069–1078
33. Wang P, Xiao X, Glissen Brown JR et al (2018) Development and validation of a deep-learning algorithm for the detection of polyps during colonoscopy. Nat Biomed Eng 2:741–748
34. Badrinarayanan V, Kendall A, Cipolla R (2017) SegNet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans Pattern Anal Mach Intell 39:2481–2495
35. Wang P, Berzin TM, Glissen Brown JR et al (2019) Real-time automatic detection system increases colonoscopic polyp and adenoma detection rates: a prospective randomised controlled study. Gut 68:1813–1819
36. Gupta N, Bansal A, Rao D, Early DS, Jonnalagadda S, Wani SB et al (2012) Prevalence of advanced histological features in diminutive and small colon polyps. Gastrointest Endosc 75(5):1022–1030
37. Guo Z, Nemoto D, Zhu X, et al. A polyp detection algorithm can detect small polyps: An ex vivo reading test compared with endoscopists[J]. Digestive Endoscopy
38. Japanese Society for Cancer of the Colon and Rectum (2019) Japanese Classification of Colorectal, Appendiceal, and Anal Carcinoma: the 3d English Edition [Secondary Publication]. J Anus Rectum Colon 3:175–195
39. Pimentel-Nunes P, Dinis-Ribeiro M, Ponchon T et al (2015) Endoscopic submucosal dissection: European Society of Gastrointestinal Endoscopy (ESGE) guideline. Endoscopy 47:829–854
40. Draganov P, Wang A, Othman M et al (2019) AGA institute clinical practice update: endoscopic submucosal dissection in the united states. Clin Gastroenterol Hepatol 17:16–25
41. Saitoh Y, Obara T, Watari J et al (1998) Invasion depth diagnosis of depres- sed type early colorectal cancers by combined use of videoendoscopy and chromoendoscopy. Gastrointest Endosc 48:362–370
42. Horie H, Togashi K, Kawamura YJ et al. (2008) Colonoscopic stigmata of 1 mm or deeper submucosal invasion in colorectal cancer. Dis Colon Rectum, 1529–1534
43. Ignjatovic A, East JE, Suzuki N, Vance M, Guenther T, Saunders BP (2009) Optical diagnosis of small colorectal polyps at routine colonoscopy (Detect InSpect ChAracterise Resect and Discard; DISCARD trial): a prospective cohort study. Lancet Oncol 10:1171–1178
44. Ladabaum U, Fioritto A, Mitani A et al (2013) Real-time optical biopsy of colon polyps with narrow band imaging in community practice does not yet meet key thresholds for clinical decisions. Gastroenterology 144:81–91

45. Rees CJ, Rajasekhar PT, Wilson A et al (2017) Narrow band imaging optical diagnosis of small colorectal polyps in routine clinical practice: the Detect Inspect Characterise Resect and Discard 2 (DISCARD 2) study. Gut 66:887–895
46. Tischendorf JJW, Gross S, Winograd R et al (2010) Computer-aided classification of colorectal polyps based on vascular patterns: a pilot study. Endoscopy 42:203–207
47. Gross S, Trautwein C, Behrens A et al (2011) Computer-based classification of small colorectal polyps by using narrow-band imaging with optical magnification. Gastrointest Endosc 74:1354–1359
48. Tamai N, Saito Y, Sakamoto T et al (2017) Effectiveness of computer-aided diagnosis of colorectal lesions using novel software for magnifying narrow-band imaging: a pilot study[J]. Endoscopy Int Open 5(8):E690
49. Byrne MF, Chapados N, Soudan F et al (2019) Real-time differentiation of adenomatous and hyperplastic diminutive colorectal polyps during analysis of unaltered videos of standard colonoscopy using a deep learning model[J]. Gut 68(1):94–100
50. Chen PJ, Lin MC, Lai MJ et al (2018) Accurate classification of diminutive colorectal polyps using computer-aided analysis[J]. Gastroenterology 154(3):568–575
51. Mori Y, Kudo S, Wakamura K et al (2015) Novel computer-aided diagnostic system for colorectal lesions by using endocytoscopy (with videos)[J]. Gastrointest Endosc 81(3):621–629
52. Misawa M, Kudo S, Mori Y, et al (2016) Characterization of colorectal lesions using a computer-aided diagnostic system for narrow-band imaging endocytoscopy[J]. Gastroenterology 150(7):1531–1532. e3
53. Mori Y, Kudo S, Chiu PWY et al (2016) Impact of an automated system for endocytoscopic diagnosis of small colorectal lesions: an international web-based study[J]. Endoscopy 48(12):1110–1118
54. Takeda K, Kudo S, Mori Y et al (2017) Accuracy of diagnosing invasive colorectal cancer using computer-aided endocytoscopy[J]. Endoscopy 49(08):798–802
55. Mori Y, Kudo S, Misawa M et al (2018) Real-time use of artificial intelligence in identification of diminutive polyps during colonoscopy: a prospective study[J]. Ann Intern Med 169(6):357–366
56. Maeda Y, Kudo S, Mori Y et al (2019) Fully automated diagnostic system with artificial intelligence using endocytoscopy to identify the presence of histologic inflammation associated with ulcerative colitis (with video)[J]. Gastrointest Endosc 89(2):408–415
57. Tokunaga M, Matsumura T, Nankinzan R, et al (2020) A computer-aided diagnosis system using only white-light endoscopy for the prediction of invasion depth in colorectal cancer[J]. Gastrointest Endosc 93(3):647–653
58. Nakajima Y, Zhu X, Nemoto D et al (2020) Diagnostic performance of artificial intelligence to identify deeply invasive colorectal cancer on non-magnified plain endoscopic images[J]. Endosc Int Open 8(10):E1341

Chapter 11
Deep Learning-Driven Models for Endoscopic Image Analysis

Xiao Jia, Xiaohan Xing, Yixuan Yuan, and Max Q.-H Meng

Abstract The advent of video endoscopy has led to an increased interest in the development of computer-aided diagnosis (CAD) approaches. Many of these focus on the use of deep learning methods as a means of automatically identifying abnormalities during endoscopy to lessen the workload on doctors. In this chapter, we take two tasks in endoscopic image analysis as examples, to survey the state of the art, recent advances, and future directions of CAD applications, especially with regard to deep learning models. We introduce the fundamentals of deep learning-driven methods and elaborate on their success in areas such as endoscopic image classification, detection of abnormal regions, and lesion boundary segmentation.

11.1 Introduction

Over the past few decades, there has been a great revolution in endoscopic imaging modalities, such as wireless capsule endoscopy (WCE) for gastrointestinal tract examination and video colonoscopy for large intestinal examination. Unlike many other medical imaging techniques, endoscopes used during examination procedures are inserted into the organ and allow the doctor to directly view the interior of the

X. Jia (✉) · X. Xing
Department of Electronic Engineering, The Chinese University of Hong Kong,
Hong Kong, China
e-mail: xjia@link.cuhk.edu.hk

X. Xing
e-mail: xhxing@ee.cuhk.edu.hk

Y. Yuan
Department of Electrical Engineering, City University of Hong Kong, Hong Kong, China
e-mail: yxyuan.ee@cityu.edu.hk

M. Q.-H. Meng
Department of Electronic and Electrical Engineering, Southern University of Science and
Technology, Shenzhen, China
e-mail: max.meng@ieee.org

T. D. Pham et al. (eds.), *Advances in Artificial Intelligence, Computation, and Data Science*, Computational Biology 31,
https://doi.org/10.1007/978-3-030-69951-2_11

organ. Computer-aided diagnosis (CAD) approaches, as a means of automatically identifying the abnormalities during endoscopy, will enhance diagnostic accuracy and efficiency.

Computational methods in endoscopy can be traced back to the early 2000s for tumor recognition in endoscopic images [1, 2]. Since then, several other methods, primarily based on computer vision techniques, have been proposed by computer engineers and scientists to support medical decision-making in the form of automated recognition of lesions [3].

The majority of early studies were based on hand-engineered methods, which typically started with a step of manually designed (also referred to as handcrafted) feature extraction, followed by a separate process to train a classifier.

With the progress of computational technologies, deep learning is making major advances in the computer vision field due to its unique capability to learn features. Deep learning has turned out to be highly capable of discovering representations via large amounts of available data and computation. The widespread use of deep learning for image analysis started in 2012, when Krizhevsky et al. [4] proposed their convolutional neural network (CNN) model, AlexNet, and won the ImageNet challenge by a large margin. To date, deep learning has become the dominant technology for almost all recognition tasks and has led to remarkable advances not only in natural image analysis [4, 5], but also in biomedical applications, such as cell classification [6], robotic tool detection [7], and gland segmentation [8].

Challenges in Endoscopic Image Analysis Despite the promising results obtained using computational methods, there remain several unsolved challenges facing the medical application of deep learning-driven strategies to automated lesion recognition with endoscopy. It is often difficult for deep learning models to extract features that discriminate between true abnormalities and artifacts due to the complex mucosa of the hollow organ. Further efforts and innovative approaches with enhanced learning capabilities are needed to improve on the present level of recognition accuracy. Annotated datasets in endoscopic imaging are small compared to natural image datasets. The generality of deep learning-driven approaches needs to be improved to address the challenges brought by relatively few training data being available for feature extraction and learning. The need for a cost-effective recognition system is increasingly demanded by clinical workflows. Trade-offs between recognition accuracy and computational efficiency should be carefully studied to achieve better performance in a clinical setting.

The remainder of this chapter is organized as follows. Section 11.2 describes various deep learning architectures applied in endoscopic image analysis, with an emphasis on the typical techniques encountered in the current literature for endoscopic image classification, detection of abnormal regions, and lesion boundary segmentation. Sections 11.3 and 11.4 examine application areas for gastrointestinal (GI) lesion screening. We take two tasks in endoscopic image analysis as examples, gastrointestinal hemorrhage recognition in wireless capsule endoscopy images (Sect. 11.3) and colorectal polyp recognition in colonoscopy images (Sect. 11.4), to elaborate on methods achieving state-of-the-art performance. Concluding remarks and some possible future directions are given in Sect. 11.5. Our earlier works related to this chapter were published in Refs. [9–14].

11.2 Deep Learning Architectures

11.2.1 Convolutional Neural Networks for Image Classification

Classification of images plays an important role in the clinical practice of screening lesions. Accurate classification can facilitate diagnosis, while missing an abnormal frame could lead to adverse or even life-threatening consequences for the patient. The classification result can provide information for subsequent lesion detection and segmentation tasks.

Convolutional neural networks (CNNs) are a type of deep neural network and form the basis of most deep learning methods used in computer vision. The basic building blocks to construct a CNN model are the convolutional and pooling layers. At each convolutional layer l, the input image is convolved with a set of k kernels $\mathcal{W}^l = \mathcal{W}_1^l, \mathcal{W}_2^l, ..., \mathcal{W}_k^l$ and added biases $b^l = b_1^l, b_2^l, ..., b_k^l$, each generating a new feature map $\mathbf{x}_k^l$. These features are subjected to an element-wise non-linear transform $\sigma(.)$ and the same process is repeated for every convolutional layer l ("$*$" denotes the convolution operation):

$$\mathbf{x}_k^{l+1} = \sigma(\mathcal{W}_k^l * \mathbf{x}_k^l + b_k^l). \tag{11.1}$$

A pooling layer (average or max-pooling) follows a convolutional layer to downsample the feature maps. With the final fully connected layers, an end-to-end CNN model can be constructed for image classification tasks. Training of CNN-based models is performed by backpropagation and stochastic gradient descent. When new data is available, a well-trained model is able to predict and generate the corresponding class labels.

Figure 11.1 shows an eight-layer convolutional neural network for GI hemorrhage recognition that is composed of three convolutional layers (conv1–conv3), three pooling layers (pool1–pool3), and two fully connected layers (fc1, fc2) [9]. The first convolutional layer (conv1) filters the 240 × 240 × 3 input image with 32 kernels of size 5 × 5 × 3 with a stride of 1 pixel (this is the distance between the receptive field centers of neighboring neurons in a kernel map) and a pad of 2 pixels (this specifies the number of pixels to implicitly add to each side of the input). The second convolutional layer (conv2) has 32 kernels of size 5 × 5 × 32 connected to the pooled outputs of the first convolutional layer (conv1). The third convolutional layer (conv3) has 64 kernels of size 5 × 5 × 32, and the first fully connected layer (fc1) has 50 neurons. Rectified linear units (ReLUs) are used as the activation function in convolutional layers (conv1–conv3) and the first fully connected layer (fc1). Max-pooling is applied

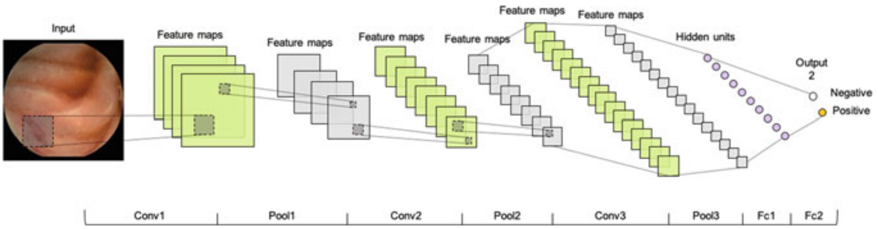


Fig. 11.1 Typical CNN framework for endoscopic image classification. The proposed deep model generates class labels for each input frame

in the pooling layers (pool1–pool3) to select the maximal activations over input patches. The output of the second fully connected layer (fc2) consists of 2 neurons (hemorrhage and normal) and can be activated by a softmax regression function, which is defined as

$$f_{\boldsymbol{\theta}}(\mathbf{x}^{(l)}) = \begin{bmatrix} P(y=1|\mathbf{x}^{(l)};\boldsymbol{\theta}) \\ P(y=2|\mathbf{x}^{(l)};\boldsymbol{\theta}) \\ \vdots \\ P(y=K|\mathbf{x}^{(l)};\boldsymbol{\theta}) \end{bmatrix} = \frac{1}{\sum_{j=1}^{K} \exp((\boldsymbol{\theta}^{(j)})^\top \mathbf{x}^{(l)})} \begin{bmatrix} \exp((\boldsymbol{\theta}^{(1)})^\top \mathbf{x}^{(l)}) \\ \exp((\boldsymbol{\theta}^{(2)})^\top \mathbf{x}^{(l)}) \\ \vdots \\ \exp((\boldsymbol{\theta}^{(K)})^\top \mathbf{x}^{(l)}) \end{bmatrix}, \tag{11.2}$$

where $\mathbf{x}^{(l)} \in \mathbb{R}^n$ are the input attributes with the corresponding labels $y^{(l)}$. K is the number of classes. The model parameters $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(K)} \in \mathbb{R}^n$ are trained to minimize the loss function:

$$L(\boldsymbol{\theta}) = -\left[\sum_{l=1}^{m}\sum_{k=1}^{K} 1\left\{y^{(l)} = k\right\} \log \frac{\exp((\boldsymbol{\theta}^{(k)})^\top \mathbf{x}^{(l)})}{\sum_{j=1}^{K} \exp((\boldsymbol{\theta}^{(j)})^\top \mathbf{x}^{(l)})}\right], \tag{11.3}$$

where m denotes the size of the training set. In the binary classification setting, we have $y^{(l)} \in \{0, 1\}$ and $K = 2$.

11.2.2 Region-Level CNNs for Lesion Detection

Apart from classification tasks that identify whether a frame contains a lesion, the output of region-level detection algorithms can also indicate the location of the lesion in the selected frames. The importance of region-level detection algorithms can be explained from two perspectives. One, the location boxes of GI lesions can provide auxiliary information for diagnosis and instruct doctors to conduct surgery and two, an indication of the location of the lesion enables explainable and reliable diagnosis.

Faster R-CNN [15] is an advanced CNN framework for object detection that can take arbitrarily sized images as input and output a set of class-specified rectangular

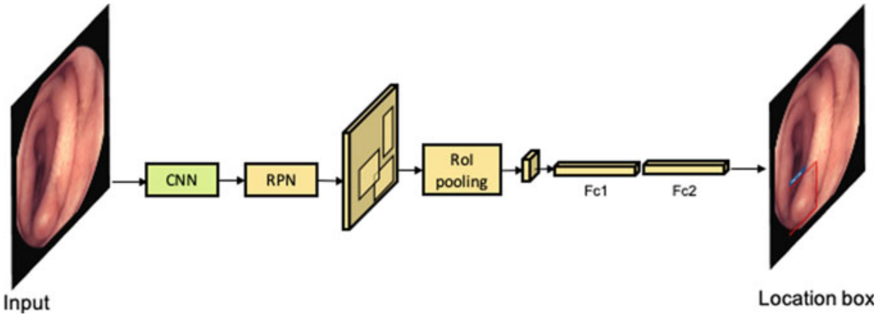


Fig. 11.2 Faster R-CNN framework for region-level lesion detection. The output highlights the location box areas with the presence of GI lesions

regions (also referred to as location or bounding boxes). Figure 11.2 illustrates a polyp detector we constructed in [13], which can be seen as a binary-class implementation of Faster R-CNN, where both Region Proposal Network (RPN) and Region of Interest (RoI) layers are adopted to perform classification and bounding box regression in parallel. The RPN is the core operation of Faster R-CNN and replaces the handcrafted RoI selection by embedding it into the network that can be trained end-to-end. The RoI pooling layer [16] is a max-pooling layer that allows downsampling of arbitrarily sized features inside the proposal regions into feature maps with a small, fixed spatial size (we use 7 × 7 by default). We use two 1,024-d fc layers (each followed by a ReLU) before the final class-specific classifier and bounding box regressor. The detection output highlights the bounding box areas with the presence of GI lesions.

In the training phase, the loss function can be minimized by adapting the multi-task loss defined in [16]:

$$L(p, c^*, b, r^*) = L_{cls}(p, c^*) + \lambda c^* L_{reg}(b, r^*). \tag{11.4}$$

Here p is the predicted probability distribution over the two categories (polyp and background). b is a vector showing the four coordinates of the predicted bounding box. c^* and r^* represent the ground truth class and bounding box regression target, respectively. c^* is 1 if the target is an area presenting a polyp, and 0 otherwise. L_{cls} is a log loss for binary classification, i.e., $L_{cls}(p, c^*) = -\log(p_{c^*})$ for true class c^*. For bounding box regression, we use the smooth L_1 loss [16] to eliminate sensitivity to outliers:

$$smooth_{L_1}(y) = \begin{cases} 0.5y^2 & \text{if } |y| < 1 \\ |y| - 0.5 & \text{otherwise,} \end{cases} \tag{11.5}$$

and $L_{reg}(b, r^*) = smooth_{L_1}(b - r^*)$ is activated simply for polyp targets. The hyper-parameter λ is applied for weight-balancing between L_{cls} and L_{reg}.

11.2.3 Fully Convolutional Neural Networks for Segmentation

To achieve finer recognition, the next step is to make a pixel-accurate inference which produces an object mask for each frame. Compared to image-level classification and region-level detection, lesion segmentation is considered as a higher-level recognition task as it aims to classify every pixel in an input image and predict the exact outlines of the lesion region. Pixel-by-pixel details are learned and provided for lesion recognition, as opposed to classification models, which identify whether an image contains a lesion, and detection models, which place a bounding box around the lesion. Segmentation is related to both classification and region detection, because these techniques facilitate learning in a segmentation model—after an abnormal frame is selected and the region of the lesion is isolated with a bounding box, further learning within a segmentation model can achieve pixel-accurate prediction.

Deep models for segmentation tasks are expected to be capable of encoding a lesion's spatial layout in the input frames. Thus, unlike the image-level recognition or region-level localization that inevitably lack spatial dimensions—feature maps in those models are collapsed into vector representations by the fully connected layers—the segmentation implementation is expected to maintain the explicit spatial structure throughout the network and has more powerful discrimination capability. To this end, we resort to the fully convolutional network (FCN) [5] which enables in-network upsampling and pixel-wise inference using a fully convolutional approach. As shown in Fig. 11.3, FCN preserves the pixel-to-pixel spatial correspondence by converting all fully connected layers to convolutions and performs per-pixel categorization for segmentation tasks. Upsampling (sometimes referred to as *deconvolution*) is the key element of FCN to increase the spatial dimension and thus bridge the resolution gap between coarse feature maps and dense output predictions. The upsampling operation is performed by bilinear interpolation [17] that computes the value of each output point from the four nearby input points on the feature map. 1×1 convolution is applied as a special case of the convolutional layer where a convolution kernel of size 1×1 is used for dimensionality reduction:

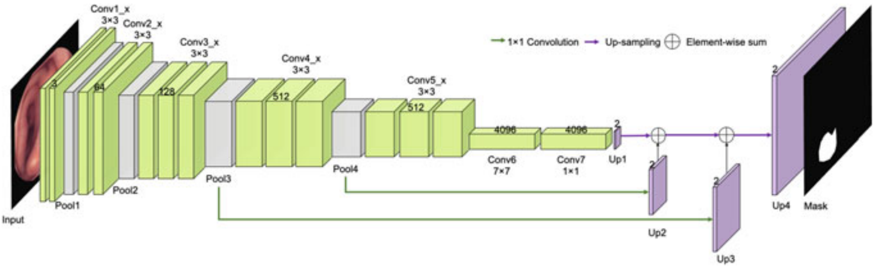


Fig. 11.3 FCN framework for pixel-wise segmentation. The proposed method produces a pixel-accurate lesion mask for each input frame

$$(\mathbf{x}_{l+1})_{1,1} = (\mathcal{W}_l * \mathbf{x}_l)_{1,1} + b, \tag{11.6}$$

where "$*$" denotes the convolution operation, $\mathbf{x}_l$ is the input neuron, $\mathbf{x}_{l+1}$ is the output neuron, and $\mathcal{W}_l$ denotes the weight parameter connecting $\mathbf{x}_l$ with $\mathbf{x}_{l+1}$. b is the bias unit.

We use a per-pixel binary loss to train the FCN for segmentation:

$$L_{seg}(z_i, t_i) = -\Big[t_i \log f(z_i) + (1 - t_i)\log(1 - f(z_i))\Big], \tag{11.7}$$

where z_i is the pixel-wise predicted score with the corresponding class label t_i. t_i is 1 if the pixel belongs to the lesion area, and is 0 for the background. $f(z_i)$ is a sigmoid function (logistic function) that squashes z_i in the range [0, 1]:

$$f(z_i) = \frac{1}{1 + e^{z_i}}. \tag{11.8}$$

11.3 Case Study I: Gastrointestinal Hemorrhage Recognition in WCE Images

11.3.1 Background of the Application

The Wireless Capsule Endoscope (WCE) [18] is a revolutionary device to provide direct, painless, and non-invasive inspection of the gastrointestinal (GI) tract. It takes two or more color pictures of the GI tract every second for around 8 h until the batteries are exhausted. These images are compressed and transmitted wirelessly to a data-recording device attached to the waist of the patient. Approximately 50,000 images are generated and downloaded to a computer. WCE sequences are reviewed by clinicians to analyze potential diseases and their sources in the GI tract.

Although the WCE has shown significant advantages over traditional endoscopes when inspecting the GI tract, there is still room for improvement. One problem is that WCE sequences are manually reviewed, which is laborious and time-consuming [19]. The mean reading time is 45 min (ranging from 30 to 75 min), while captured abnormal images usually occupy less than 5% of the collected images [20]. WCE frames show different spatial characteristics (shape, texture, and size), which can make it difficult for clinicians to reliably detect them in all circumstances [21]. In this regard, the development of computational support systems for automated analysis of WCE sequences is highly desirable.

Gastrointestinal (GI) hemorrhage, also known as GI bleeding, is one of the most common abnormalities in the GI tract [22]. GI bleeding can be categorized as active bleeding (i.e., ongoing bleeding) (Fig. 11.4b) versus inactive bleeding (Fig. 11.4c), such as erythema and arteriovenous malformation, and may occur anywhere through-

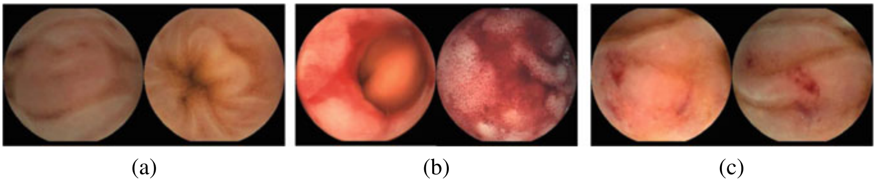


Fig. 11.4 Examples of WCE images in our dataset. **a** Normal WCE images. **b** Active bleeding WCE images. **c** Inactive bleeding WCE images

out the GI tract. Hemorrhagic abnormalities show great intra-class variation in terms of morphology, size, and location (Fig. 11.4), making hemorrhage recognition in WCE sequences vitally important, but challenging.

With the aim of relieving the workload of physicians, computerized approaches to gastrointestinal (GI) hemorrhage recognition in wireless capsule endoscope (WCE) images have become an active research area. Existing methods typically start with a manually designed feature (handcrafted features) extraction step, followed by a separate process training the classifier. In the first stage, features such as color [23], texture [24], and statistical information [25] are manually extracted from original WCE images. These feature vectors are used to learn a binary or discrete classifier, where both support vector machine (SVM) [23, 26] and k-nearest neighbor (KNN) [24, 25] techniques are commonly used.

11.3.2 Improved Learning Strategy

11.3.2.1 Image-Level Hemorrhage Recognition

Handcrafted features are useful in that they attempt to model the kinds of features that doctors look for when identifying a GI hemorrhage. For example, as hemorrhage has a distinct red hue, computer engineers could empirically define a certain range of chromaticity in which features of GI hemorrhage appear to be concentrated. However, their design circle relies heavily on domain knowledge. Another drawback of hand-engineered approaches is that they may lose useful information and not be optimized, because they are not part of an end-to-end learning system (i.e., feature design is separate from training the classifier). These issues motivated us to improve recognition performance by developing deep learning-driven methods to find higher-level and more discriminatory features. Deep learning methods are usually more complex computationally and require a large number of annotated images for model training. As we have obtained handcrafted features in previous studies, there is an appeal in combining the merits of domain knowledge and hierarchical feature learning, so that these two could complement each other, and potentially outperform either individual strategy for hemorrhage recognition.

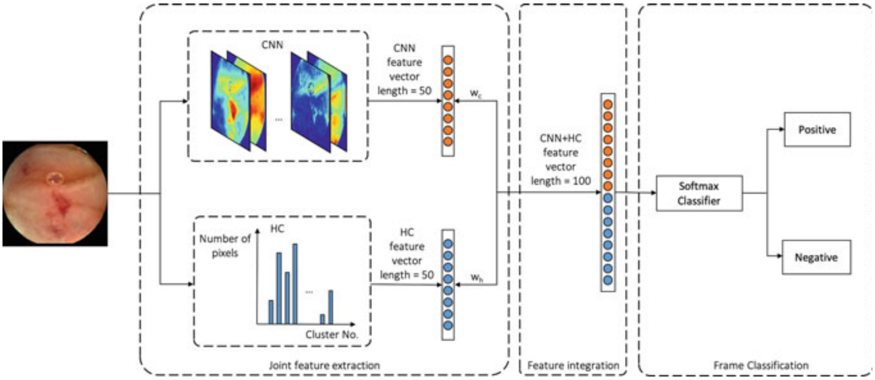


Fig. 11.5 An overview of the proposed framework for image-level hemorrhage recognition

Figure 11.5 shows an overview of the proposed framework for image-level hemorrhage recognition. Its three stages are joint feature extraction, feature integration, and classification. In joint feature extraction, both CNN-derived and handcrafted (HC) features are extracted from the input frame. We use k-means clustering to extract for handcrafted features and for feature learning via CNN, the output of the first fully connected layer in Fig. 11.1 constitutes the CNN feature vector of each input frame. Features of the two categories are integrated based on a defined strategy so that the combined feature vector serves as the input for classification. The final decision is obtained via a Softmax classifier.

Before feature integration, normalization is performed on the HC and CNN features. Then for each input WCE image, its HC and CNN features are jointly modeled through an integration process, which takes the form:

$$[CNN, HC]_i = [w_c \times CNN_i, w_h \times HC_i],\ w_c + w_h = 1, \tag{11.9}$$

where i denotes the index of the input sample. w_c, w_h are weighting factors determined empirically through the experiment. The combined feature vector with a total length of $50 + 50 = 100$ is used to learn a binary classifier (Fig. 11.5).

11.3.2.2 Pixel-Level Hemorrhage Recognition

Technically, the above work provides a solution for whole-image classification. The next step in the progression of hemorrhage recognition is to make a pixel-wise prediction, i.e., to segment the blood region within the screened frames. To this end, we considered casting the pre-trained networks into a fully convolutional form, by using fully convolutional networks (FCNs).

Before training the FCN models, one concern is that regions with active or inactive bleeding abnormalities may exhibit large differences in their color and texture

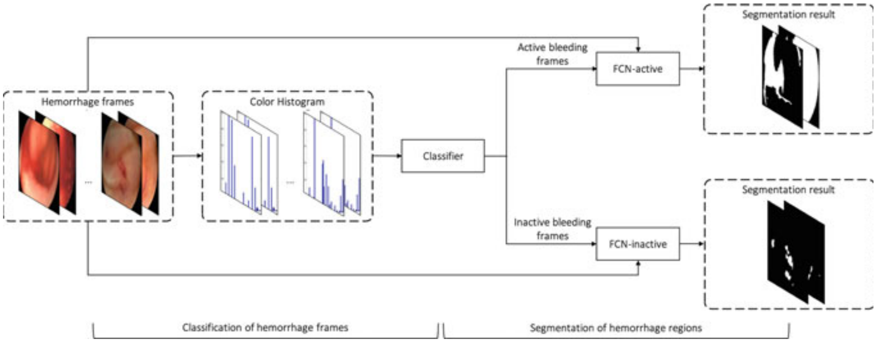


Fig. 11.6 An overview of the proposed framework for hemorrhage segmentation [11]. The proposed method first classifies the hemorrhage frames into active and inactive subgroups based on the statistical features derived from the histogram of the probability of the color space. Then for each subgroup, we highlight the blood regions via fully convolutional networks (FCNs)

components. To resolve this problem, we first classify the images with the presence of a hemorrhage into two distinct subgroups based on their statistical features. The purpose of this step is to enhance the learning process in the segmentation model.

Figure 11.6 depicts an overview of the proposed method, which comprises two stages: classification of hemorrhage frames and segmentation of hemorrhage regions. Specifically, the first stage aims to roughly classify the hemorrhage samples into two subgroups based on their statistical features, i.e., frames with apparent active bleeding regions and frames with small regions of inactive bleeding. The statistical features are described by a normalized color histogram, where a k-means algorithm and a softmax classifier are used to make decisions to classify the hemorrhage frame. In the segmentation stage, two fully convolutional network models, *FCN-active* and *FCN-inactive*, are trained separately for each subgroup. Subsequently, hemorrhage regions within the input frame can be identified by forward-passing the established network.

11.3.3 Dataset

To evaluate the performance of image-level recognition, we constructed a large WCE dataset for bleeding detection that contains 10,000 annotated images, including 2,850 GI bleeding frames and 7,150 normal frames (see Fig. 11.4). These images are extracted from 80 different patients' videos and manually annotated by expert physicians. Apart from the conventional bleeding samples with apparent active bleeding regions (Fig. 11.4b), this dataset also collects images that contain small regions of inactive bleeding (Fig. 11.4c), making the detection task more challenging. Images containing a bleeding region are labeled as positive (Fig. 11.4b, c), otherwise, they are labeled as negative (Fig. 11.4a). The size of each WCE image is 240 × 240 × 3.

To assess the performance of pixel-level recognition, we further construct a pixel-wise labeled WCE dataset. The dataset consists of 300 WCE images, all with blood-based abnormalities, including 148 active bleeding frames and 152 inactive ones. These images were extracted from 60 WCE video snippets of 12 different patients and manually annotated by expert physicians. Pixels within a bleeding region are labeled as positive, otherwise, they are labeled as negative.

11.3.4 Evaluation Metrics

We measure the performance of image-level hemorrhage recognition quantitatively by employing three commonly used metrics:

$$\begin{aligned} Recall &= \frac{TP}{TP+FN}, \\ Precision &= \frac{TP}{TP+FP}, \\ F_1\ score &= \frac{2 \times Precision \times Recall}{Precision + Recall}, \end{aligned} \tag{11.10}$$

where true positive (TP) is the number of positive samples correctly classified; false negative (FN) is the number of positive samples incorrectly classified as negative; true negative (TN) is the number of negative samples correctly classified, and false positive (FP) is the number of negative samples incorrectly classified as positive. In particular, the F_1 score is a measure of a test's accuracy in the statistical analysis of binary classification.

The performance of pixel-level recognition is measured quantitatively by reporting four metrics from common semantic segmentation tasks:

$$\begin{aligned} Mean\ IU &= \frac{1}{n_{cl}} \frac{\sum_i n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}}, \\ Frequency\ weighted\ IU &= \frac{1}{\sum_k t_k} \frac{\sum_i t_i n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}}, \\ Mean\ accuracy &= \frac{1}{n_{cl}} \frac{\sum_i n_{ii}}{t_i}, \\ Pixel\ accuracy &= \frac{\sum_i n_{ii}}{\sum_i t_i}, \end{aligned} \tag{11.11}$$

where n_{ij} is the number of pixels of class i predicted to belong to class j, $t_i = \sum_j n_{ij}$ is the total number of pixels of class i, and there are n_{cl} different classes. IU stands for the region intersection over union.

11.3.5 *Experimental Results*

11.3.5.1 Performance of Image-Level Hemorrhage Recognition

We show in Fig. 11.7a–c the constructed feature histograms of a normal WCE frame, an active bleeding frame, and an inactive bleeding frame, respectively. The histogram components vary greatly among the three categories of WCE samples, revealing that the proposed hand-engineered method is effective in feature extraction, and thus provides a reliable basis for the following recognition tasks. Since we apply k-means clustering to summarize the color information, the dimensionality of the WCE images could be considerably reduced leading to high-speed processing of the classification.

After the CNN model optimization is completed, the discriminatory regions of active bleeding and inactive bleeding are highlighted (Fig. 11.8d), suggesting that the feature maps generated by the CNN are able to characterize hemorrhage from the complex gastrointestinal mucosa. The recognition accuracy of the proposed CNN model reached 0.9813 recall, 0.9899 precision, and 0.9856 F_1 score, thereby validating that deep learning is highly capable of hemorrhage recognition.

At this point, we expect feature generation based on domain knowledge and CNN learning will be complementary, and joint modeling of handcrafted features and the CNN features could further improve the recognition performance.

We compared the image-level recognition performance of our proposed method with the latest approaches by Fu et al. [23] and Yuan et al. [27], where both utilized handcrafted features. We implemented their methods on our dataset to perform a direct comparison and report the results in Table 11.1. The proposed hand-engineered method (referred to as *Ours (HC only)*) and CNN-based method (referred to as *Ours (CNN only)*) are also involved to validate the key ideas we present. *Ours (HC only)* characterizes the input images using the proposed handcrafted manner only, whereas in *Ours (CNN only)* each WCE image is summarized via purely CNN-derived features. Apart from the image-level metrics defined in Eq. (11.11), TP, TN, FP, and FN are listed to provide a comparative reference. The two subsets of our method (i.e., *Ours (HC only)* and *Ours (CNN only)*) perform comparably or better than the state-of-the-art methods, demonstrating the efficacy of our ideas. More importantly, the proposed strategy with joint feature extraction (i.e., *Ours (CNN+HC)*) outperforms the other four comparisons by a considerable margin for

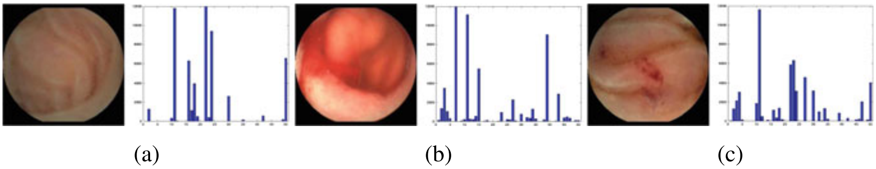

(a) (b) (c)

Fig. 11.7 Some examples of the constructed feature histogram [12]. **a** Example of a normal sample and its feature histogram. **b** Example of an active bleeding sample and its feature histogram. **c** Example of an inactive bleeding sample and its feature histogram

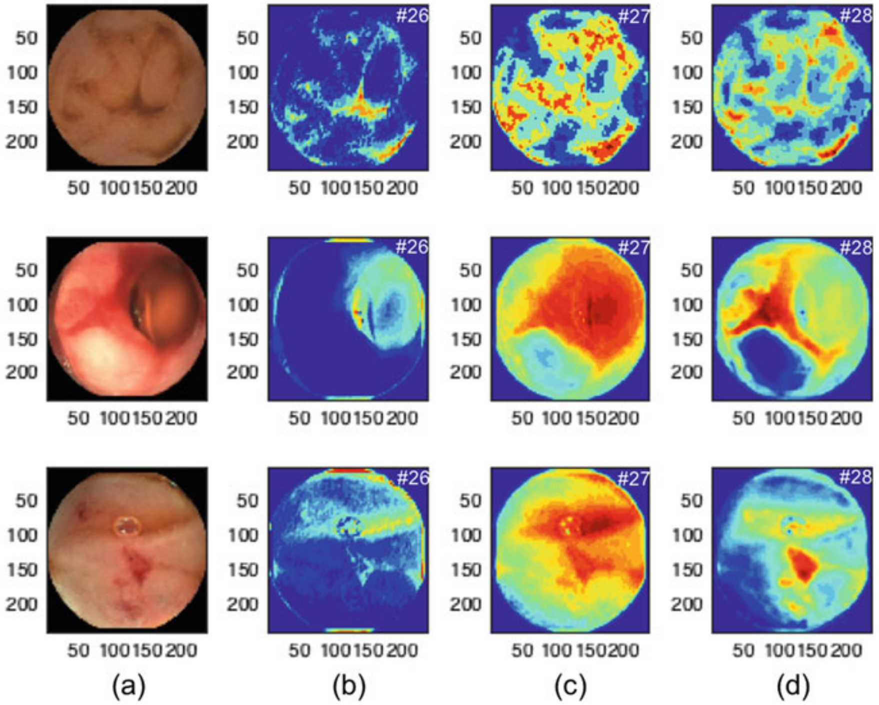


Fig. 11.8 Some training samples and their #26, #27, #28 feature maps after forward-passing the first convolutional layer (conv1) [9]. **a** Raw data. **b** The #26 feature maps. **c** The #27 feature maps. **d** The #28 feature maps

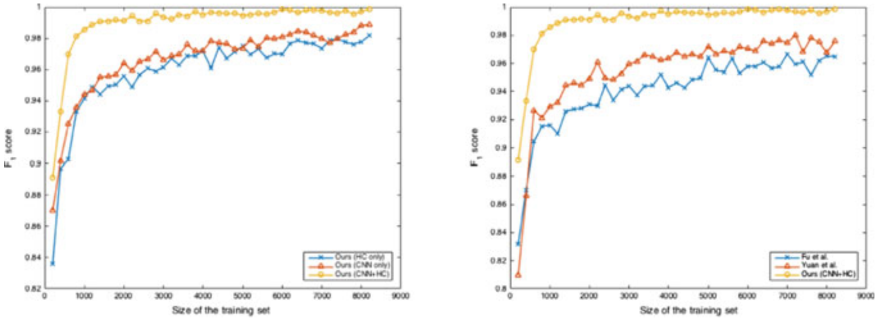
all seven measures, especially in terms of recall (from 0.9538 to 0.9975) and F_1 score (from 0.9744 to 0.9968), suggesting that the CNN and handcrafted features are effective complements to each other.

We further investigated the impact of the size of the training set on the performance of hemorrhage recognition. Experiments were performed by varying the size of the training set from 200 to 8200 (fixed ratio sampling, $positive : negative = 1 : 3$) with a step size of 200 images.

Figure 11.9 illustrates the comparative results in terms of F_1 score, where Fig. 11.9a gives the comparison among *Ours (CNN+HC)* and the two subset versions, Fig. 11.9b plots the performance curve for both *Ours (CNN+HC)* and the two leading approaches by [23, 27]. From Fig. 11.9a, the proposed method combining both CNN and HC features outperforms either class of feature extraction strategies individually, showing the improvement obtained by integrating HC and CNN features. Figure 11.9b shows the recognition performance (in terms of F_1 score) among *Ours (CNN+HC)* and the two comparison methods. Our method outperforms the state-of-the-art approaches in two points: (1) When only a relatively small-sized training set (<1000) is available, the proposed method leads to stronger results compared to

Table 11.1 Quantitative comparison of our methods with state-of-the-art hemorrhage recognition approaches. The same training and test sets are used in all five methods. Boldface indicates the best performance

Method	Type	TP	TN	FP	FN	Recall	Precision	F_1 score
Fu et al. [23]	HC	758	987	13	42	0.9475	0.9831	0.9650
Yuan et al. [27]	HC	763	997	3	37	0.9538	0.9961	0.9744
Ours (HC only)	HC	775	994	6	25	0.9688	0.9923	0.9804
Ours (CNN only)	CNN	785	992	8	15	0.9813	0.9899	0.9856
Ours (CNN+HC)	**CNN+HC**	**798**	**997**	**3**	**2**	**0.9975**	**0.9962**	**0.9968**



(a) (b)

Fig. 11.9 Comparison of hemorrhage recognition performance (in terms of F_1 score) with variable sizes of the training set. **a** Comparison of F_1 score among *Ours (HC only)*, *Ours (CNN only)*, and *Ours (CNN+HC)*. **b** Comparison of F_1 score among [23, 27] and *Ours (CNN+HC)*

[23, 27], indicating that the combined strategy benefits from the CNN architecture extracting richer information via high-level representations even with limited data; (2) As the number of training samples increases, the performance of both ours and the two other methods saturates quickly, thus, a clear improvement of the proposed method can be observed. Our method further advances the state of the art in WCE hemorrhage recognition.

Running Time Computational costs were recorded in terms of time efficiency: testing a 240 × 240 RGB image takes around ~100 ms using the joint modeling method on a desktop with an *Intel Xeon Processor E5-2630V3* and a single GPU of *NVIDIA GeForce GTX TITAN X*. In the training stage, the proposed CNN model converges in about 50 min. Including the time needed to extract handcrafted features (20 min in a pure MATLAB implementation), all training for CNN+HC can be completed within 2 h, which is significantly faster than directly training a very deep CNN model.

The above results validate that the proposed CNN and handcrafted features are efficacious complements to each other, with an achieved F_1 score of up to 0.9968. By performing high-level recognition while minimizing the computing resources required, our method combines the merits of domain knowledge and hierarchical feature learning, and thus further improves the classification accuracy of GI hemorrhage recognition in WCE images.

11.3.5.2 Performance of Pixel-Level Hemorrhage Recognition

Figure 11.10 shows the segmentation results on the test sets after 10, 000, 50, 000, and 100, 000 training iterations. Performance rises quickly for both subgroups. The segmentation results yielded after 50, 000 training iterations for *FCN-active* and 100, 000 training iterations for *FCN-inactive* are quite satisfactory when compared with the corresponding ground truth, validating the learning capability of our FCN-based models.

The performance of our proposed method is quantitatively compared with two state-of-the-art approaches by Fu et al. [23] and Yuan et al. [27]. Their methods were based on hand-engineered techniques and are capable of segmentation of hemorrhage regions. We implemented their methods on the same training and test sets to perform a direct evaluation. Experimental results on the test sets with active bleeding and inactive bleeding are reported in Table 11.2. We also evaluated the performance of a unified FCN model (called *FCN-unified*), which is trained without the classification step, therefore on an unseparated dataset containing both active and inactive bleeding samples. This strategy performs comparably to the latest methods for active bleeding detection, while bringing a minor improvement of mean IU to 0.7441. However, the *FCN-unified* provides little benefit on inactive bleeding detection, perhaps due to the large differences in color, morphology, and/or size between the two categories of bleeding. Therefore, the classification step is necessary, and separately training two FCN models shows the most accurate result in bleeding segmentation tasks.

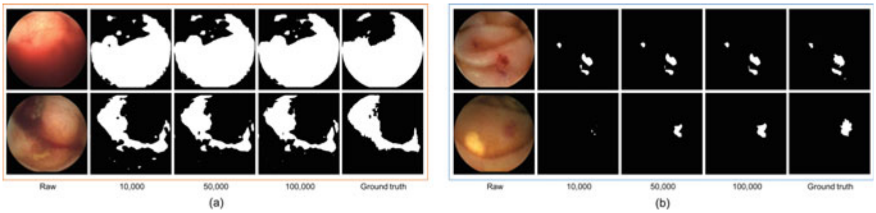


Fig. 11.10 Segmentation result after 10, 000, 50, 000, and 100, 000 training iterations [11]. **a** Segmentation of on-going bleeding regions via *FCN-active*. **b** Segmentation of small blood-based abnormalities via *FCN-inactive*

Table 11.2 Comparison of hemorrhage segmentation approaches. Boldface indicates the best performance [11]

Method	Active bleeding				Inactive bleeding			
	Mean IU	F.w. IU	Mean acc.	Pixel acc.	Mean IU	F.w. IU	Mean acc.	Pixel acc.
Fu et al. [23]	0.6973	0.7048	0.8348	0.8244	0.6770	0.9805	0.7079	0.9895
Yuan et al. [27]	0.7211	0.7327	0.8365	0.8448	0.6403	0.9706	0.7853	0.9805
FCN-unified	0.7441	0.7532	0.8580	0.8577	0.6620	0.9751	0.7736	0.9845
FCN-active	**0.7750**	**0.7854**	**0.8691**	**0.8796**	–	–	–	–
FCN-inactive	–	–	–	–	**0.7524**	**0.9848**	**0.8030**	**0.9917**

Table 11.2 illustrates that our method achieves an increase in the four metrics, especially in terms of mean IU. The *FCN-active* improves the segmentation score of active bleeding from 0.7211 mean IU to 0.7750 mean IU. For the segmentation task of inactive bleeding, the *FCN-inactive* scores 0.7524 mean IU for an improvement of more than 7% points. This result demonstrates the advantages of an end-to-end, deep-learning-based strategy in WCE diagnostic applications.

11.4 Case Study II: Colorectal Polyp Recognition in Colonoscopy Images

11.4.1 Background of the Application

Colorectal cancer (CRC) is the third most common cancer and the third-leading cause of cancer deaths among both men and women in the United States [28]. Fortunately, screening can prevent CRC through the detection and removal of precancerous polyps, as well as diagnose cancer at an early stage, when treatment is usually less extensive and more successful.

Colonoscopy is the recommended and gold standard screening method for CRC, allowing direct visual examination of the entire colon and rectum. CRC usually begins as an adenomatous polyp, therefore, accurate recognition of polyps in colonoscopy images is crucial for early CRC diagnosis and treatment [29]. Despite technical advances, current manual reviews of colonoscopies are laborious and time-consuming, requiring the undivided concentration of the endoscopist. In this regard, the development of computational support systems for automatic detection is highly desirable.

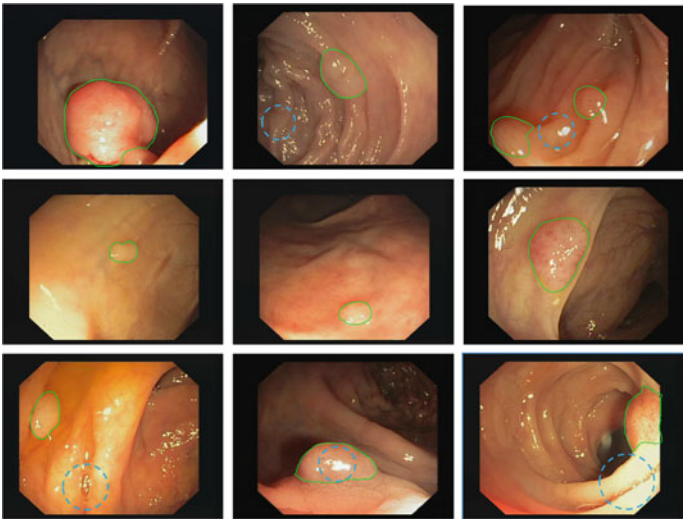


Fig. 11.11 Example of the technical challenges for automatic polyp recognition. From top to bottom: large intra-class variation of polyps and low inter-class variation between polyps and folds; obscure polyp boundaries with low contrast to the background; the presence of artifacts. In each image, the green line indicates the polyp contour. A dashed blue line highlights the presence of intestinal folds in the first row, while in the third row, it circles the intestinal content, specular highlights, and the overexposed region, respectively

Automatic recognition of polyps in colonoscopy images is, however, a very challenging task. The intra-class variation of polyps is large in terms of morphology, size, and location, while there is limited inter-class variation between polyps and other elements of the scene (folds, vessels), making it difficult to discriminate polyps from colonic mucosa. Polyps with a flat morphology or at their early stages show obscure boundaries and low contrast to the background, thus increasing the challenge. Artifacts generated due to the occurrence of intestinal content (bubbles, fecal particles), as well as the presence of specular highlights and overexposed regions, may further aggravate the situation. Figure 11.11 shows examples of these challenges.

To tackle these challenges, several approaches have been proposed to model discriminatory features and improve the accuracy of recognition. The majority of early studies were based on hand-engineered methods, which exploited the low-level features to obtain candidate polyps. For instance, Silva et al. [30] integrated texture and shape features, and Bernal et al. [31] proposed a method based on intensity valleys to boost detection accuracy. Despite sustained improvements, these hand-engineered methods are far from sufficient to represent the complex appearance of polyps.

Some of the most recent methods apply convolutional neural networks (CNNs) to polyp detection, taking advantage of their hierarchical feature learning and discriminatory capabilities. A pioneering study in this field constructed a 7-layer CNN model (with three convolutional layers) to extract features of polyps [32]. Faster R-

CNN [15], a Region-based CNN detector, has been applied to automatically detect polyps [33]. A 3-dimensional CNN framework has been used to localize polyps in colonoscopy videos [34]. Fully convolutional network (FCN) architectures were introduced [35, 36] for pixel-accurate detection of polyps and segmentation. These methods were extended by restructuring FCN with different convolutional backbone architectures (e.g., AlexNet, VGG-16, and ResNet), and their performance was assesed on publicly available datasets [37]. CNN architectures, DeepLab [38] and SegNet [39], were used to achieve pixel-wise segmentation of polyps [40, 41].

However, these developments are still incapable of routine diagnosis and require further improvement. Existing methods with shallow CNN architectures are limited in their ability to precisely characterize polyp lesions from the complex colonic mucosal surface and have been constrained by the relatively limited training data available for medical applications, while the efficiency of advanced deep models needs to be improved to have lower computational costs.

11.4.2 Improved Learning Strategy

The depth of feature representation is of crucial importance [42], and many effective recognition methods have benefited from very deep networks. The development of a very deep network architecture may obtain higher-level representations leading to an accurate recognition of polyps. Another consideration is the scale of representation as polyps show great variation in their morphology and size, which is a fundamental challenge. Multi-scale feature extraction that has richer semantics at all levels could give excellent gains in accuracy [43], which is the basis of our work.

We present a new approach using a very deep CNN architecture for polyp recognition, which gains accuracy from deeper and richer representations. The method, called PLPNet ("PLP" stands for "polyp"), can effectively detect polyps in colonoscopy images and generate high-quality segmentation masks in a pixel-to-pixel manner. An overview of the PLPNet model shows its two stages: polyp proposal and polyp segmentation (Fig. 11.12). The learning process would be hindered by the complex colonic wall if pixel-wise training was performed directly on the CNN model. Therefore, the two-stage framework uses the polyp proposal stage as a region-level polyp detector that can guide pixel-level learning in the polyp segmentation stage to accurately segment the area the polyp occupies in the image. The very deep ResNet-50 and pyramid components were applied to seek deeper and richer semantics from each frame. Feature sharing and skip schemes were developed to perform multi-scale transfer learning between stages I and II.

11.4.2.1 Multi-Level Feature Learning via Very Deep ResNet

To mask each polyp accurately, multi-level and discriminative features must be obtained from raw data. Residual Networks (ResNets) [42] are deep convolutional neural networks that use a series of residual blocks Res i_(·) and shortcut connections bypassing Res i_(·), where each residual block comprises several stacked convolutional layers. This residual representation addresses the degradation problem caused by a deeper network architecture and allows gains in accuracy from the considerably increased depth.

With $\mathbf{x}_l$ as the input, the residual block can be formulated in the general form:

$$\mathbf{x}_{l+1} = \mathbf{x}_l + \mathcal{F}(\mathbf{x}_l, \{\mathcal{W}_l\}), \tag{11.12}$$

where $\mathcal{F}(\mathbf{x}_l, \{\mathcal{W}_l\})$ represents the residual mapping function to be learned in the residual block. When $\mathbf{x}_l$ and $\mathcal{F}$ have the same dimension, an identity shortcut connection and element-wise addition are used to perform the "+" operation in Eq. (11.12). If this is not the case (e.g., dimension changes after downsampling), a linear projection $\mathcal{W}_s$ (performed by a 1 × 1 convolutional layer) is employed to match the dimensions between the input and output:

$$\mathbf{x}_{l+1} = \mathcal{W}_s\mathbf{x}_l + \mathcal{F}(\mathbf{x}_l, \{\mathcal{W}_l\}). \tag{11.13}$$

We can then construct a very deep architecture by simply stacking the Res i_(·) without degradation in training. Figure 11.12a illustrates the architecture of ResNet-50, which serves as the backbone of the PLPNet model. The feature activation output by each residual stage's last block, i.e., {Res 2_(c), Res 3_(d), Res 4_(f), Res 5_(c)}, is connected to the pyramid components to perform multi-scale feature learning.

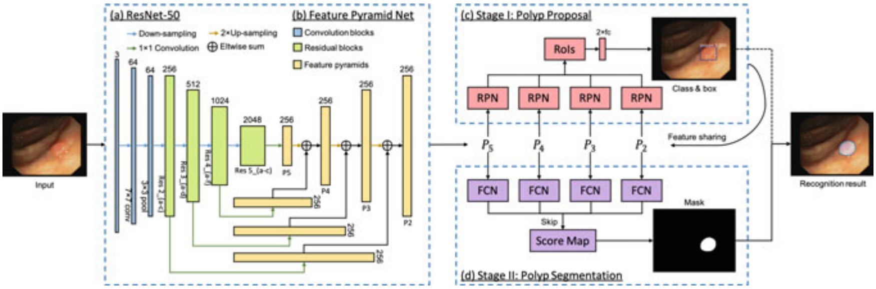


Fig. 11.12 An overview of the proposed PLPNet for accurate polyp recognition [13]. **a** Illustration of ResNet-50. The network consists of a 7 × 7 convolutional layer, a 3 × 3 max-pooling layer, and four residual stages {Res 2_(a-c), Res 3_(a-d), Res 4_(a-f), Res 5_(a-c)}. **b** Architecture of the Feature Pyramid Net. The set of feature maps in the pyramid layers is called {P2, P3, P4, P5}, corresponding to the output of the last residual blocks in each stage, i.e., {Res 2_(c), Res 3_(d), Res 4_(f), Res 5_(c)}, that are of the same spatial sizes, respectively. **c** Illustration of stage I for polyp proposal. RPN components and RoI layers are adopted to perform classification and bounding box regression. **d** Stage II is built as a fully convolutional network (FCN) for pixel-wise polyp segmentation. Semantic information in stage II is transferred from stage I via feature sharing and skip schemes

11.4.2.2 Multi-scale Feature Representation via FPNet

To recognize polyps of various sizes and morphologies, detecting objects at vastly different scales is beneficial. Here an effective, pyramidal architecture is utilized to extract strong semantics at all feature scales. Feature Pyramid Network (FPNet) [43] can construct feature pyramids from a single-scale input image using a top-down pathway and lateral connections. We build four pyramid levels referring to the number of residual stages (see Fig. 11.12b). The output set of feature maps in the pyramid layers is called {P2, P3, P4, P5} and forms the basic pyramid components in the PLPNet model to recognize polyps at different scales. The output widths and heights of {P2, P3, P4, P5} are {1/4, 1/8, 1/16, 1/32} of the input image size, corresponding to the scales of {Res 2, Res 3, Res 4, Res 5}.

We constructed the polyp proposal stage as an extension of Faster R-CNN to serve as a polyp detector that identifies the area of each lesion as a whole region-level prediction (Fig. 11.12c). We adapted the network by attaching the RPN block to each level of the pyramid components {P2, P3, P4, P5} and obtained multi-scale proposal regions. On the basis of the pyramid levels, RoI layers of different spatial scales are assigned to the corresponding RPN output. The feature maps (RoIs) resulting from pyramid components P2 to P5 can be directly concatenated and fed into a sequence of fully connected (fc) layers to extract feature vectors. The detection output of stage I is referred to as polyp proposals that highlight the bounding box of areas with the proposed presence of polyps.

11.4.2.3 Stage II: Pixel-Wise Polyp Segmentation

The segmentation task encodes the object's spatial layout in the input image. Thus, we adapt FCNs to our two-stage framework by designing skip schemes for each pyramid level and initializing the network by feature sharing from polyp proposals. Stage II, with a fully convolutional architecture, can seek richer and deeper semantics while requiring a lower training cost, leading to better accuracy and faster speed. The segmentation stage is built using the same backbone architecture as stage I, which makes feature sharing between the two tasks possible (Fig. 11.12d).

Figure 11.13 shows the detailed architecture inside stage II, which can be seen as an inverse top-down pathway of the backbone network. We built four FCN pathways referring to the pyramid components {P2, P3, P4, P5}, where the deepest/coarsest level (P5) passes through four upsampling steps to reach the final score map.

11.4.3 Dataset

Two publicly available datasets were used in the context of the validation study of polyp recognition, **CVC-ColonDB** and **CVC-ClinicDB**. CVC-ColonDB [31], also known as CVC-300, contains 300 Standard Definition (SD) frames with a resolution

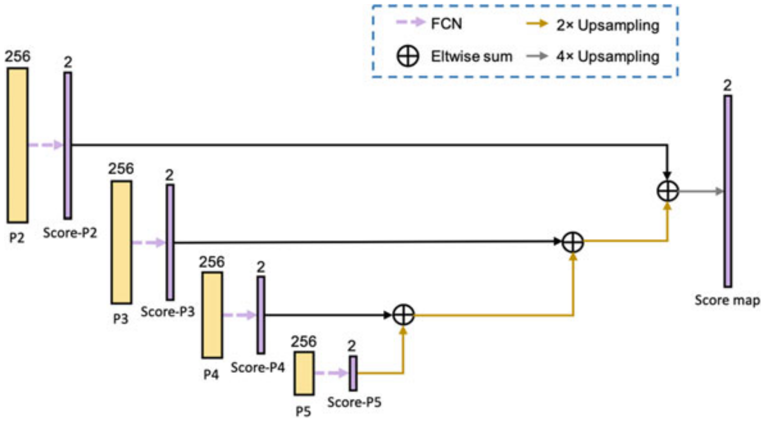


Fig. 11.13 Illustration of the fully convolutional pathway architecture in stage II [13]. The score map with 2 channel dimensions is generated for each pyramid component. The output score map is the same size as the input image

of 500 × 574 pixels that were obtained from 13 polyp video sequences acquired from 13 patients. CVC-ClinicDB [35], also known as CVC-612, contains 612 SD frames at 384 × 288 pixel resolution obtained from 31 different sequences of 23 patients. Both datasets were built to cover as many varieties of polyps as possible, such as having differences in morphology, size, location, and illumination. While each frame contains at least one polyp, some may have more than one: 30 frames in CVC-612 have two or three polyps and thus increase the complexity of detection. Images were annotated by expert video-endoscopists, and a binary mask was generated to represent the actual polyp area within each image.

To implement the polyp proposal stage, a ground truth setting, consisting of a polyp bounding box, was generated based on the image binary mask to fit the contour of the polyp in a sufficiently tight way.

Our model was trained using the CVC-300 database, following the dataset settings of the *GIANA 2017 Challenge on Automatic Polyp Segmentation*,[1] and CVC-612 was used during the testing stage. This imposed the constraint that one patient's records cannot be in different phases, and allowed validation of the system's effectiveness and generality.

11.4.4 Evaluation Metrics

The performance of polyp recognition was measured quantitatively by reporting four pixel-based metrics that are commonly used in medical image segmentation tasks [35] (Table 11.3, left panel). A pixel within a polyp is considered a **True Positive (TP)** if

[1] https://endovissub2017-giana.grand-challenge.org/polypsegmentation/.

Table 11.3 Definition of performance measures

Metrics for polyp segmentation	Metrics for polyp localization
Prec = $\frac{TP}{TP+FP}$	Prec = $\frac{TP}{TP+FP}$
Rec = $\frac{TP}{TP+FN}$	Rec = $\frac{TP}{TP+FN}$
J = $\frac{\lvert PR_{cl} \cap GT_{cl} \rvert}{\lvert PR_{cl} \cup GT_{cl} \rvert}$	F1 = $\frac{2 \times Prec \times Rec}{Prec+Rec}$
Dice = $\frac{2\lvert PR_{cl} \cap GT_{cl} \rvert}{\lvert PR_{cl} \rvert + \lvert GT_{cl} \rvert}$	F2 = $\frac{5 \times Prec \times Rec}{4 \times Prec+Rec}$

its prediction is polyp; otherwise, it is regarded as a **False Negative (FN)**. A pixel that falls outside the polyp is said to be a **False Positive (FP)** if its prediction is polyp. High precision (Prec) indicates a method can return substantially more relevant instances than irrelevant ones, while high recall (Rec) means a method can output most of the relevant instances. The Jaccard index (J, also known as the Intersection over Union) and Sørensen-Dice coefficient (Dice) are statistics for comparing the similarity of the sample sets, where PR_{cl} and GT_{cl}, respectively, denote the prediction set and the ground truth of one class. $\cap$ represents the set intersection and $\cup$ represents set union. The Mean Jaccard index and Dice similarity score along with their standard deviation were used as the performance metrics in the *GIANA 2017 Challenge on Automatic Polyp Segmentation*, and the same metrics were used to evaluate the recognition performance of our method.

To assess the performance of polyp proposal in stage I, the frame-based metrics for polyp localization of the *GIANA Challenge* [44] (Table 11.3, right panel), where precision and recall are defined as above, were used. Metrics for polyp localization were measured at the frame level: if the detection output is within the polyp, the method is said to be providing a TP, and only one TP will be considered per polyp. Any detection that falls outside the polyp is considered an FP. The absence of detection in a frame is considered an FN, counting one per polyp not recognized in the frame. F1 and F2 are the F1-score and F2-score that measure a test's accuracy. Higher F1 and F2 indicate a method has a better balance between the true and false predictions. In our implementation, the geometric center of each detection box was regarded as the prediction output of a polyp's location.

11.4.5 Experimental Results

11.4.5.1 Performance of Polyp Region Proposal

The effects of network depth and feature scales on polyp recognition on the CVC-612 test set with various backbones, where boldface indicates the best performance in each of the metrics, are shown in Table 11.4. The proposal stage output was evaluated by the performance of polyp localization using the frame-based metrics defined in Table 11.3.

Table 11.4 Comparison of polyp proposal performance on CVC-612 with different backbone architectures [13]

Backbone	TP	FP	FN	Prec (%)	Rec (%)	F1 (%)	F2 (%)
ZF	524	116	122	81.9	81.1	81.5	81.3
VGG-16*	535	**96**	111	**84.8**	82.8	83.8	83.2
ResNet-50	549	239	97	69.7	85.0	76.6	81.4
ResNet-101	557	3608	89	13.4	86.2	23.2	41.3
ResNet-50+FPNet	**595**	107	**51**	**84.8**	**92.1**	**88.3**	**90.5**

*Original form of Faster R-CNN

The effect of network depth can be observed by the gradually increasing recall, with over 5 points growth from ZF (AlexNet), VGG-16, ResNet-50, and ResNet-101, demonstrating that enriching feature depth can effectively improve the performance of detecting true polyps. However, the number of FP proposals tends to grow as the network depth increases—while ResNet-101 performs best in TP and recall, its large number of FPs worsens its performance on other metrics. One possible reason is difficulties with optimization caused by the deeper architecture, as ResNet-101 has about twice the parameters as ResNet-50, and so requires even greater computing resources and larger training sets. To achieve accurate recognition, it was desired to generate as many proposals as true polyp regions while keeping a relatively small FP count, which will benefit the learning process of stage II. We therefore instantiated PLPNet with ResNet-50 and explored FPNet as the backbone architecture. The performance of ResNet-50+FPNet exceeded other schemes with gains in most metrics—only FP was slightly larger than that of VGG-16 (107 versus 96). ResNet-50+FPNet improved recall by about 7 points over ResNet-50, and similar improvements in precision, F1, and F2 were also observed. As this difference is due solely to the benefits of the pyramid components, this highlighted that multi-scale feature representation is one key to enhance the discrimination capability of our model. Hence, we applied PLPNet with a ResNet-50+FPNet backbone and used the same backbone network for both stages to make feature sharing possible.

As a further interpretation, we applied class activation maps (CAMs) [45] to provide insights into the network's properties and visualize the internal representations learned by PLPNet. Given that CAMs can highlight the most discriminatory regions used by networks to identify objects of a specific class, we generated polyp-specific CAMs for the pyramid components by computing the weighted sum of their feature maps. Example CAM outputs of P2 and P5 on the CVC-612 test set and the corresponding polyp proposal results are shown in Fig. 11.14. PLPNet accurately located the polyp regions with high prediction confidence, and the detection box was very close to the ground truth (Fig. 11.14d). This validated the performance and discriminatory ability of the polyp proposal scheme. Semantic information extracted from P2 and P5 appears differently for the density and range of the highlighted areas, but they all correctly indicate RoIs with a true presence of a polyp. Features learned

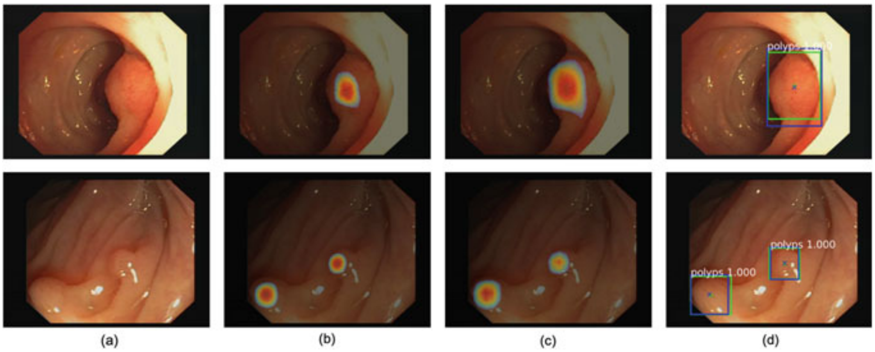


Fig. 11.14 Examples of polyp proposal results on the CVC-612 test set [13]. From left to right: **a** original images, **b** class activation maps (CAMs) generated from P2 and **c** P5 layers of PLPNet model, **d** polyp proposal output using PLPNet: stage I. The ground truth is in green and the detection output is in blue. Each output box is shown with a polyp class label and an associated prediction score in [0, 1]. A confidence threshold of 0.990 is used to display the results

from different pyramid levels should be complementary and capable of guiding the learning process in the segmentation stage.

11.4.5.2 Performance of Pixel-Wise Polyp Segmentation

Next, we investigated PLPNet for polyp segmentation and evaluated the performance by pixel-based metrics and compared it to a variant of PLPNet without the sharing of features from the polyp proposal stage. The segmentation performance on the CVC-612 test set is reported in Table 11.5. Removing the polyp proposal stage significantly degrades the accuracy of the segmentation by 6.7 points by J and 5.8 points by Dice, suggesting that a per-pixel learning scheme without feature sharing will be challenged by the fundamental difficulty of polyp recognition. This highlights the role of polyp proposal, where features transferred from stage I can lead to enhanced results in polyp segmentation.

We then compared PLPNet to state-of-the-art methods in polyp segmentation on the CVC-612 test set and reported the results in Table 11.6. PLPNet shows a

Table 11.5 Comparison of feature learning schemes [13]

Learning scheme	J (%)		Dice (%)	
	Mean	Std	Mean	Std
W/o polyp proposal stage	68.0	25.5	78.1	18.5
With polyp proposal stage	**74.7**	**20.5**	**83.9**	**13.6**

Table 11.6 Comparison of polyp segmentation performance on CVC-612 with different methods [13]

Method	Rec (%)	Prec (%)	J (%)		Dice (%)	
			Mean	Std	Mean	Std
VGG-16-FCN [35]	25.0	67.6	57.1	34.7	66.1	29.6
VGG-16-FCN+ [36]	36.1	64.0	61.0	31.0	71.0	24.8
ResNet-50-FCN [37]	43.4	79.3	66.2	27.2	76.4	20.2
PSPNet [46]*	45.1	81.6	68.0	26.8	78.0	20.1
DeepLab v3+ [47]*	48.7	85.8	70.4	24.3	80.2	18.2
PLPNet	**59.4**	85.9	**74.7**	**20.5**	**83.9**	**13.6**

*We simply revised PSPNet and DeepLab v3+ using the ResNet-50 backbone and fitted the polyp segmentation task

large improvement over the state-of-the-art models. ResNet-50-FCN [37] gives better results (mean J = 66.2) than the challenge's benchmarks VGG-16-FCN [35] (mean J = 57.1) and VGG-16-FCN+ [36] (mean J = 61.0), showing the gains in accuracy from increased network depth. PSPNet [46] and DeepLab v3+ [47] use the ResNet-50 backbone and improve the results, with mean J values of 68.0 and 70.4, respectively, suggesting gains in accuracy from exploiting multi-scale information. PLPNet showed significant increases in accuracy for all metrics, achieving a mean J of 74.7 on the test, which is a ∼6%, ∼10%, and ∼13% relative improvement over DeepLab v3+, PSPNet, and ResNet-50-FCN, respectively. The gains of PLPNet over [37, 46, 47] come from adding the proposal stage, using two-stage pyramid feature sharing, and the skip schemes. Compared to the challenge's benchmark [35], PLPNet improved mean J by 17.6 points (over 30% relative improvement) and mean Dice by 17.8 points, precision by 18.3 points, and doubled the accuracy on recall. This improvement can be attributed to the learning strengths of a much deeper and richer representational framework.

The output of PLPNet, ResNet-50-FCN [37], and DeepLab v3+ [47] on some challenging cases from CVC-612 are shown in Fig. 11.15. This comparison covers cases with multi-scale polyp lesion areas (Fig. 11.15a–c), obscure or low contrast polyp boundaries (Fig. 11.15a–c), overexposed regions (Fig. 11.15d), specular highlights (Fig. 11.15e), presence of intestinal folds (Fig. 11.15d–f), and a case with more than one lesion (Fig. 11.15f).

Both ResNet-50-FCN and DeepLab v3+ produced no activation in Fig. 11.15a, indicating a difficulty in recognizing small, obscure polyps. ResNet-50-FCN, the baseline method, may miss abnormalities and create spurious edges in cases with obscure lesion boundaries (Fig. 11.15a–c) and tended to provide a higher number of FPs (Fig. 11.15d–f). A fully convolutional architecture with an increased network depth may not be sufficient for polyp segmentation, and it might be influenced by the presence of overexposed regions, specular highlights, and intestinal folds. DeepLab v3+ improved the performance on segmentation (Fig. 11.15b, c) by applying an atrous

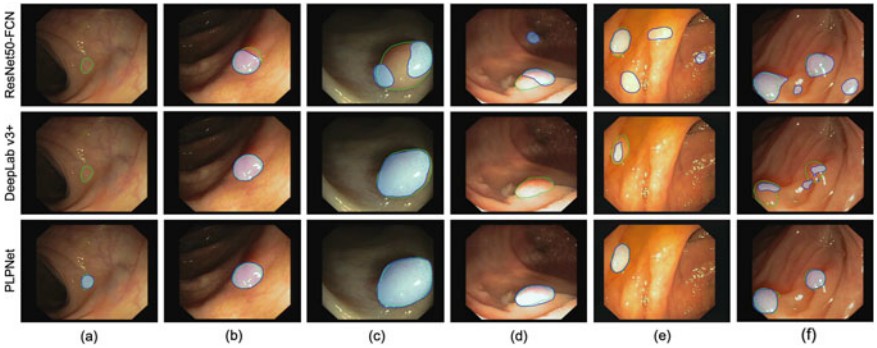


Fig. 11.15 Examples of polyp segmentation results on CVC-612 test images [13]. ResNet-50-FCN [37] (top), DeepLab v3+ [47] (middle), and PLPNet (bottom). ResNet-50-FCN exhibits errors in the inference of (**a**) small lesion objects, (**a**–**c**) obscure polyp boundaries, (**d**) overexposed regions, (**e**) specular highlights, and (**e**, **f**) intestinal folds. DeepLab v3+ shows errors in the inference of (**a**) small lesion objects, (**d**) overexposed regions, and (**e**, **f**) polyps of low contrast with the intestinal folds. Green outlines denote the ground truth of polyp boundaries and the segmentation output is shown by blue outlines and masks

convolution [38] to extract multi-scale, denser feature maps but had less favorable results when encountering overexposed regions and intestinal folds (Fig. 11.15d–f).

PLPNet showed no such errors and achieved results that were very close to the experts' annotations in these challenging cases. This indicates that the PLPNet framework enables accurate polyp segmentation across a large range of scales by using the pyramid components and skip schemes. It highlights the key effect of the feature sharing strategy, where parameters transferred from the proposal stage allow the model to recognize the lesion area as a whole, and thus eliminate regional discontinuities in the segmentation masks.

PLPNet is superior to baseline and advanced semantic segmentation frameworks for pixel-accurate polyp recognition.

Running Time PLPNet has an inference time of ∼166 ms per input image on a single *NVIDIA GeForce GTX TITAN Xp*. As a reference, the challenge's benchmark VGG-16-FCN [35] takes ∼85 ms. PLPNet increases inference time due to the computational cost generated by its deeper network backbone and extra pyramid layers, but gives clear gains in accuracy. The baseline model, ResNet-50-FCN [37], takes ∼280 ms and is slower than PLPNet due to the lighter-weighted FCN heads in PLPNet and despite the extra costs of pyramid layers. The PSPNet and DeepLab v3+ models took ∼370 ms and ∼310 ms per image, respectively.

11.5 Conclusion and Future Perspectives

In this chapter, we focused on addressing the problem of automated recognition of lesions in endoscopic images by developing effective deep learning strategies for computer-aided diagnosis. Deep learning-driven methods were applied on two representative tasks in endoscopic image analysis, recognizing gastrointestinal hemorrhages in wireless capsule endoscopy images and recognizing colorectal polyps in colonoscopy images. Extensive experiments demonstrated superior performance of the proposed methods relative to current practice.

Deep learning-based methods are leading to improved diagnostic accuracy in medical image analysis. In the majority of tasks presented in this chapter, deep learning models were shown to be extremely effective for automated screening of diseases, in this case recognizing lesions in endoscopic images. We suggest three key issues for the further development of deep learning models to analyze endoscopic images.

Weakly Supervised Learning for a Constantly Growing Dataset Weakly supervised detection is the problem of learning object detectors using only image-level labels. Every endoscopy procedure results in large quantities of data, which are time-consuming to annotate manually. Learning complex concepts using only light supervision would significantly reduce the cost of data annotation for computerized lesion detection tasks. Weakly supervised networks [48] could ease the training of deep learning models to recognize lesions in the constantly growing datasets of endoscopic images.

Incorporation of Temporal Information for Sequence Recognition Recurrent Neural Networks (RNNs), particularly the Long Short Term Memory (LSTM) RNNs [49], have been used to extract temporal dependencies in the analysis of surgical videos [50]. The incorporation of spatial and temporal information considerably improved performance, and a similar benefit might apply if LSTM-based CNN models were applied to endoscopy videos.

System Integration and Optimization for Real-Time Diagnosis The requirement for computational efficiency in diagnostic software increases due to the real-time constraints posed by clinical workflows. Deep learning models could be used for online processing of endoscopy images if performance could be boosted. Work on system integration and optimization, e.g., (a) video pre-processing to remove frame redundancy and enhance visual quality, (b) video-based localization, to facilitate diagnosis and follow-up examinations, (c) extension of deep models to multi-lesion recognition tasks [51, 52], and (d) clinical trials on various patient cohorts, are likely to be beneficial. An integrated system would be expected to perform real-time diagnosis and detection with high accuracy and efficiency.

Acknowledgements This work was supported in part by the National Key R&D program of China under Grant 2019YFB1312400, the Hong Kong Research Grants Council (RGC) Collaborative Research Fund (CRF) Project under Grant C4063-18G, and the Shenzhen Science and Technology Innovation Project under Grant JCYJ20170413161503220.

References

1. Karkanis SA, Iakovidis DK, Maroulis DE, Magoulas GD, Theofanous NG (2000) Tumor recognition in endoscopic video images using artificial neural network architectures. In: Proceedings of the 26th Euromicro Conference. EUROMICRO 2000. Informatics: inventing the future, vol. 2. IEEE, New York, pp 423–429
2. Karkanis SA, Iakovidis DK, Maroulis DE, Karras DA, Tzivras M (2003) Computer-aided tumor detection in endoscopic video using color wavelet features. In: IEEE Transactions on information technology in biomedicine, vol 7. IEEE, New York, pp 141–152
3. Iakovidis DK, Koulaouzidis A (2015) Software for enhanced video capsule endoscopy: challenges for essential progress. In: Nature reviews gastroenterology & hepatology, vol 12. Nature Publishing Group, Berlin, pp 172–186
4. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. Curran Associates, Inc., Red Hook, NY, pp 1097–1105
5. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, pp 3431–3440
6. Lei H, Han T, Zhou F, Yu Z, Qin J, Elazab A, Lei B (2018) A deeply supervised residual network for hep-2 cell classification via cross-modal transfer learning. In: Pattern recognition, vol 79. Elsevier, Amsterdam, pp 290–302
7. Sarikaya D, Corso JJ, Guru KA (2017) Detection and localization of robotic tools in robot-assisted surgery videos using deep neural networks for region proposal and detection. In: IEEE transactions on medical imaging, vol 36. IEEE, New York, pp 1542–1549
8. Xu Y, Li Y, Liu M, Wang Y, Lai M, Eric I, Chang C (2016) Gland instance segmentation by deep multichannel side supervision. In: International conference on medical image computing and computer-assisted intervention. Springer, Berlin, pp 496–504
9. Jia X, Meng MQ-H (2016) A deep convolutional neural network for bleeding detection in wireless capsule endoscopy images. In: 2016 38th annual international conference of the IEEE engineering in medicine and biology society (EMBC). IEEE, Orlando, FL, pp 639–642
10. Jia X, Meng MQ-H (2017) Gastrointestinal bleeding detection in wireless capsule endoscopy images using handcrafted and cnn features. In: 2017 39th annual international conference of the IEEE engineering in medicine and biology society (EMBC). IEEE, South Korea, pp 3154–3157
11. Jia X, Meng MQ-H (2017) A study on automated segmentation of blood regions in wireless capsule endoscopy images using fully convolutional networks. In: 2017 IEEE 14th international symposium on biomedical imaging (ISBI). IEEE, Melbourne, pp 179–182
12. Jia X, Cai L, Liu J, Dai W, Meng MQ-H (2016) GI bleeding detection in wireless capsule endoscopy images based on pattern recognition and a MapReduce framework. In: 2016 IEEE international conference on real-time computing and robotics (RCAR). IEEE, Cambodia, pp 266–271
13. Jia X, Mai X, Cui Y, Yuan Y, Xing X, Seo H, Xing L, Meng MQ-H (2020) Automatic polyp recognition in colonoscopy images using deep learning and two-stage pyramidal feature prediction. In: IEEE transactions on automation science and engineering. IEEE
14. Jia X, Xing X, Yuan Y, Xing L, Meng MQ-H (2019) Wireless capsule endoscopy: a new tool for cancer screening in the colon with deep-learning-based polyp recognition. In: Proceedings of the IEEE, vol 108. IEEE, pp 178–197
15. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. Curran Associates, Inc., Red Hook, NY, pp 91–99
16. Girshick R (2015) Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. IEEE, Cambridge, MA, pp 1440–1448
17. Bovik AC (2010) Handbook of image and video processing. Academic press, Cambridge
18. Iddan G, Meron G, Glukhovsky A, Swain P (2000) Wireless capsule endoscopy. Nature 405:417. Nature Research

19. Van Gossum A, Munoz-Navas M, Fernandez-Urien I, Carretero C, Gay G, Delvaux M, Lapalus MG, Ponchon T, Neuhaus H, Philipper M, et al (2009) Capsule endoscopy versus colonoscopy for the detection of polyps and cancer. N Engl J Med 361:264–270. Mass Medical Soc
20. Hwang S (2011) Bag-of-visual-words approach to abnormal image detection in wireless capsule endoscopy videos. In: International symposium on visual computing. Springer, Berlin, pp 320–327
21. Yu M (2002) M2A$^{\text{TM}}$ capsule endoscopy: a breakthrough diagnostic tool for small intestine imaging. Gastroenterol Nurs 25:24–27. LWW
22. Fisher L, Krinsky ML, Anderson MA, Appalaneni V, Banerjee S, Ben-Menachem T, Cash BD, Decker GA, Fanelli RD, Friis C, et al (2010) The role of endoscopy in the management of obscure GI bleeding. Gastrointest Endosc 72:471–479. Elsevier
23. Fu Y, Zhang W, Mandal M, Meng MQ-H (2014) Computer-aided bleeding detection in WCE video. IEEE J Biomed Health Inf 18:636–642. IEEE
24. Mathew M, Gopi VP (2015) Transform based bleeding detection technique for endoscopic images. In: 2015 2nd international conference on electronics and communication systems (ICECS). IEEE, Piscataway, pp 1730–1734
25. Ghosh T, Bashar SK, Alam MS, Wahid K, Fattah SA (2014) A statistical feature based novel method to detect bleeding in wireless capsule endoscopy images. In: 2014 international conference on informatics, electronics & vision (ICIEV). IEEE, Dhaka, pp 1–4
26. Yuan Y, Meng MQ-H (2015) Automatic bleeding frame detection in the wireless capsule endoscopy images. In: 2015 IEEE international conference on robotics and automation (ICRA). IEEE, Seattle, pp 1310–1315
27. Yuan Y, Li B, Meng Q (2015) Bleeding frame and region detection in the wireless capsule endoscopy video. IEEE J Biomed Health Inf 20:624–630. IEEE
28. Cancer Facts & Figures (2019) American cancer society. Atlanta, GA, USA
29. Colorectal Cancer Facts & Figures 2017–2019 (2017) American cancer society, Atlanta, GA, USA
30. Silva J, Histace A, Romain O, Dray X, Granado B (2014) Toward embedded detection of polyps in wce images for early diagnosis of colorectal cancer. Int J Comput Assist Radiol Surg 9:283–293. Springer
31. Bernal J, Sánchez FJ, Fernández-Esparrach G, Gil D, Rodríguez C, Vilariño F (2015) Wm-dova maps for accurate polyp highlighting in colonoscopy: validation vs. saliency maps from physicians. Comput Med Imaging Graph 43:99–111. Elsevier
32. Park SY, Sargent D (2016) Colonoscopic polyp detection using convolutional neural networks. In: Medical imaging 2016: computer-aided diagnosis. International society for optics and photonics, vol 9785, p 978528
33. Shin Y, Qadir HA, Aabakken L, Bergsland J, Balasingham I (2018) Automatic colon polyp detection using region based deep CNN and post learning approaches. IEEE Access 6:40950–40962. IEEE
34. Yu L, Chen H, Dou Q, Qin J, Heng PA (2017) Integrating online and offline three-dimensional deep learning for automated polyp detection in colonoscopy videos. IEEE J Biomed Health Inf 21:65–75. IEEE
35. Vázquez D, Bernal J, Sánchez FJ, Fernández-Esparrach G, López AM, Romero A, Drozdzal M, Courville A (2017) A benchmark for endoluminal scene segmentation of colonoscopy images. J Healthcare Eng 2017. Hindawi
36. Zhang L, Dolwani S, Ye X (2017) Automated polyp segmentation in colonoscopy frames using fully convolutional neural network and textons. In: Annual conference on medical image understanding and analysis. Springer, Berlin, pp 707–717
37. Brandao P, Zisimopoulos O, Mazomenos E, Ciuti G, Bernal J, Visentini-Scarzanella M, Menciassi A, Dario P, Koulaouzidis A, Arezzo A, et al (2018) Towards a computed-aided diagnosis system in colonoscopy: automatic polyp segmentation using convolution neural networks. J Med Robot Res 3:1840002. World Scientific
38. Chen L-C, Papandreou G, Schroff F, Adam H (2017) Rethinking atrous convolution for semantic image segmentation. arXiv:1706.05587

39. Badrinarayanan V, Kendall A, Cipolla R (2017) Segnet: a deep convolutional encoder-decoder architecture for image segmentation. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 39. IEEE, pp 2481–2495
40. Xiao W-T, Chang L-J, Liu W-M (2018) Semantic segmentation of colorectal polyps with DeepLab and LSTM networks. In: 2018 IEEE international conference on consumer electronics-Taiwan (ICCE-TW). IEEE, pp 1–2
41. Wang P, Xiao X, Brown JRG, Berzin TM, Tu M, Xiong F, Hu X, Liu P, Song Y, Zhang D, et al (2018) Development and validation of a deep-learning algorithm for the detection of polyps during colonoscopy. Nature Biomed Eng 2:741. Nature Publishing Group
42. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, pp 770–778
43. Lin T-Y, Dollár P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, vol 1. IEEE, p 4
44. Bernal J, Tajkbaksh N, Sánchez FJ, Matuszewski BJ, Chen H, Yu L, Angermann Q, Romain O, Rustad B, Balasingham I, et al (2017) Comparative validation of polyp detection methods in video colonoscopy: results from the miccai 2015 endoscopic vision challenge. In: IEEE transactions on medical imaging, vol 36. IEEE, pp 1231–1249
45. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A (2016) Learning deep features for discriminative localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, Las Vegas, NV, pp 2921–2929
46. Zhao H, Shi J, Qi X, Wang X, Jia J (2017) Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, Honolulu, HI, pp 2881–2890
47. Chen L-C, Zhu Y, Papandreou G, Schroff F, Adam H (2018) Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV). Springer, Berlin, pp 801–818
48. Zhou B, Li Y, Wang J (2018) A weakly supervised adaptive densenet for classifying thoracic diseases and identifying abnormalities. arXiv:1807.01257
49. Donahue J, Anne Hendricks L, Guadarrama S, Rohrbach M, Venugopalan S, Saenko K, Darrell T (2015) Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, Boston, MA, pp 2625–2634
50. Yueming J, Qi D, Hao C, Yu L, Jing Q, Fu C-W, Pheng-Ann H (2017) Sv-rcnet: workflow recognition from surgical videos using recurrent convolutional network. IEEE Trans Med Imag. IEEE
51. Xing X, Yuan Y, Meng MQ-H (2020) Zoom in lesions for better diagnosis: attention guided deformation network for WCE image classification. IEEE Trans Med Imag
52. Guo X, Yuan Y (2020) Semi-supervised WCE image classification with adaptive aggregated attention. Med Image Anal 64:101733

Part III
Physiology

Chapter 12
A Dynamic Evaluation Mechanism of Human Upper Limb Muscle Forces

Qing Tao, Zhaobo Li, Quanbao Lai, Shoudong Wang, Lili Liu, and Jinsheng Kang

Abstract Dynamic evaluation mechanisms of the human upper limb are of great value for research and applications in upper limb rehabilitation, especially for the development of robotic upper limb rehabilitation systems. This paper proposes a muscle force prediction method based on the Hill muscle model. The proposed approach, which combines sEMG signals and kinematic data, provides a deep understanding of the dynamic motion mechanisms and parameters that characterize the upper limbs of the human body. The study provides a theoretical benchmark for the evaluation of rehabilitation training practices and for improved designs of upper limb rehabilitation robots that are used for upper limb neuro-rehabilitation. Specifically, the system collected motion data and sEMG signals from the upper limbs of the human body through a high-speed infrared motion capture system and skin sEMG sensors. By applying human kinematics and dynamics theories, real-time joint angle and torque information was obtained and imported into OpenSim. This platform can simulate the real-time muscle force values produced by the upper limbs during movements. The myoelectric signals were first filtered to remove noise, and an exponential model was then used to obtain the muscle activation. These data were then entered into the Hill-type prediction model to determine an individual's muscle forces. In this paper, grasping movements commonly used in everyday situations were taken as a testing case. The results of the experiments showed that an individual's muscle forces can be predicted using a Hill-type model. The results are consistent with those from simulated muscle force models and can reflect the real forces experienced during upper limb exercises.

Q. Tao (✉) · Z. Li · Q. Lai · S. Wang · L. Liu
School of Mechanical Engineering, Xinjiang University, Urumqi 830047, China
e-mail: xjutao@qq.com

Q. Tao
State Key Laboratory for Manufacturing Systems Engineering, Jiaotong University, Xi'an, China

J. Kang
College of Engineering, Design and Physical Sciences, Brunel University London, Uxbridge UB8 3PH, UK

T. D. Pham et al. (eds.), *Advances in Artificial Intelligence, Computation, and Data Science*, Computational Biology 31,
https://doi.org/10.1007/978-3-030-69951-2_12

Keywords Muscle force · Upper limb movement · Human–robot interaction · Biomechanics analysis · sEMG signals

12.1 Introduction

Cardio-cerebrovascular disease, more commonly referred to as a stroke or heart attack, is the second most common cause of death and the eighth most common cause of severe disability in the elderly population worldwide [1, 2]. One of the most serious symptoms of stroke is hemiparesis. It usually leads to the loss of motor function in the upper limb, which is important for activities of daily living, such as eating, bathing, and getting dressed independently. Studies have shown that rehabilitation robots can effectively improve the motor function of stroke patients because the device is designed to have multiple DOFs to mimic various limb movements and accommodate all types of exercises [3]. Rehabilitation machines are becoming increasingly important for stroke rehabilitation, and they have many potential advantages over traditional rehabilitation therapies in treating motor dysfunction in stroke patients [4].

At present, upper limb rehabilitation robots can be divided into three main types based on their structure [5–8]. The first type of robots is called rehabilitation robots with end-guided structures. The subject's body is placed on a separate structure, while a separate robotic mechanism guides the movement of the forearm or the hand to train and rehabilitate the affected limb [9–11]. The second type of robots is called exoskeleton rehabilitation robots, where the robotic arms closely follow the form and function of the affected limb and can consequently achieve a variety of rehabilitation training actions [12–15]. The third type of robots is called compound rehabilitation training robots, which have a combination of features from the first two types of rehabilitation robots and are used to complete rehabilitation training [16–21]. Regardless of the type of upper limb rehabilitation robot used, it is necessary to conduct a quantitative evaluation of the patient's upper limb impairment and provide personalized rehabilitation treatment. Since the mechanism of human upper limb movement is very complicated and the range of functional movements achievable is wide, the sports medicine field has not developed a standard for assessing upper limb movement patterns. Currently, the most immediate challenge is to develop objective indicators that are related to upper limb motion and can be applied to assess upper limb function under everyday conditions.

To address the challenges mentioned above, a model simulating the output force from muscle fibers was established, and predictive models of muscle forces for individuals with different demographics were obtained in this paper. We combined the musculoskeletal model of an upper extremity with the movement and EMG-assisted method to estimate the individual muscle forces of the musculoskeletal structure. The EMG signals were considered inputs to the musculoskeletal model to estimate the muscle activation information. A simulation tool commonly used in biomechanical analyses, OpenSim, was used to analyze the agreement in muscle force estimations. Through our method, therapists can more accurately describe the power of stroke patients' upper limbs during daily activities and can design better rehabilitation techniques. In addition, the research can also be useful for controlling an upper limb exoskeleton.

The rest of this paper is organized as follows. Section 12.2 presents the related works and muscle estimation growing trend. Section 12.3 describes experiment details and data procession and muscle estimation model. Section 12.4 compares the estimated muscle force by our NMS (Neuromuscular Subjective) model with force simulated in OpenSim. Section 12.5 evaluates the performances of NMS model and discusses. Section 12.6 summarizes this article.

12.2 Related Work

Three main approaches including clinical scales, movement evaluations, and surface electromyography (sEMG) analyses, are widely applied in objective evaluations of the upper extremities [22, 23]. Clinical scales are inherently subjective due to their reliance on a physician's visual assessment of a movement and generally. It is difficult to summarize a movement with a single score, especially when several aspects, such as the speed and amplitude of the movement, have to be taken into account in the evaluation of functional tasks [24]. Movement evaluation methods mainly include motion capture systems, which represent the gold standard in human movement analysis [25–27]. Motion capture systems can accurately assess the kinematics of upper limb movements during daily activities in stroke patients [28]. However, although they can overcome the limitations of clinical scales, which lack the ability to monitor a patient's movements, motion capture systems cannot assess the internal features of a patient's muscles.

Surface electromyography (sEMG) is a popular research tool that is used extensively in sports medicine and rehabilitation sciences. Based on sEMG analysis, researchers have attempted to draw conclusions concerning the neuro and electrophysiological mechanisms of force production and make hypotheses about potential muscular force adaptation rates and hypertrophy [29]. sEMG signal decomposition algorithms can be roughly divided into two categories. The first type of sEMG signal decomposition involves waveform detection and pattern recognition. The second type involves a blind source separation method or system identification so that the sEMG signal can be described and interpreted. Recent experiments have suggested that the central nervous system can spontaneously follow certain optimization criteria to overcome the motion uncertainty caused by kinematic mechanism positional redundancy [30, 31]. Although sEMG signal analysis has value in certain applications in upper limb function evaluations, quantitative evaluations cannot be conducted due to the lack of deep muscle activation information. Surface EMG signals can be easily obtained, but the activations of deep muscles cannot be measured by non-invasive methods [32–34]. The identification of force profiles of individual muscles during upper limb movements may help provide a better understanding of the functional roles of these muscles as well as the neuro-musculoskeletal impairments, leading to a better understanding of how these factors affect movement [35, 36].

Hill-type or Hill-type modified models are vastly used in muscle estimation. All the models take sEMG as the only one input parameter in order to simplify compute procession. Muscle forces are calculated just like a black box. But it involves non-linear relationships in the expressions like muscle fiber length verse muscle force curve or muscle contraction velocity verse muscle force curve, which makes the computational process very complex. In order to acquire universality, the relationship between muscle contraction velocity and fiber length verse muscle force must be scaled [35, 39, 41]. In contrast to Hill-type model compute muscle force using sEMG, there is another method called inverse dynamics. The inverse dynamics method takes position, velocity, and acceleration as input parameters to calculate the jet moment. Then static or dynamic optimization is used to obtain individual muscle force. This method greatly simplifies the calculation procession but suffers from the problem of imprecision.

12.3 Materials and Methods

The experimental goal for this paper was to simultaneously capture data about a human subject's upper limb motion and about his or her limb surface EMG signals. These data were then analyzed and processed using optimizing calculations to obtain accurate values predicting muscle forces during exercise. The human upper limb musculoskeletal model was established by OpenSim, and then the kinematic data gathered by motion capture were used to simulate the muscle forces during the upper limb movements. The surface EMG signal data were then imported into a Hill-type model. By adjusting the parametric coefficients to achieve model predictive values close to the simulated muscle force values, we established a method of predicting muscle force directly from the surface EMG signal data (Fig. 12.1).

12.3.1 Data Collection and Preprocessing

Four healthy male subjects (age: 23.5 ± 1.2 years old, height: 171.3 ± 3.5 cm, weight: 72 ± 6.5 kg) volunteered to participate in the experiment and were included in the study after they signed written informed consent forms. The research project was pre-approved by the Research Ethics Committee of Xinjiang University. While the subjects sat in a chair, they autonomously moved their arm from the natural relaxed state to grasp a raised ball that was suspended directly in front of them at head height, released the ball and put their arm down. The body motion data from the subjects were collected using a VICON optical motion capture system, and the surface electromyography data from the relevant muscles of the upper limbs of the subjects were collected using a Neuracle 16 channel electromyography signal

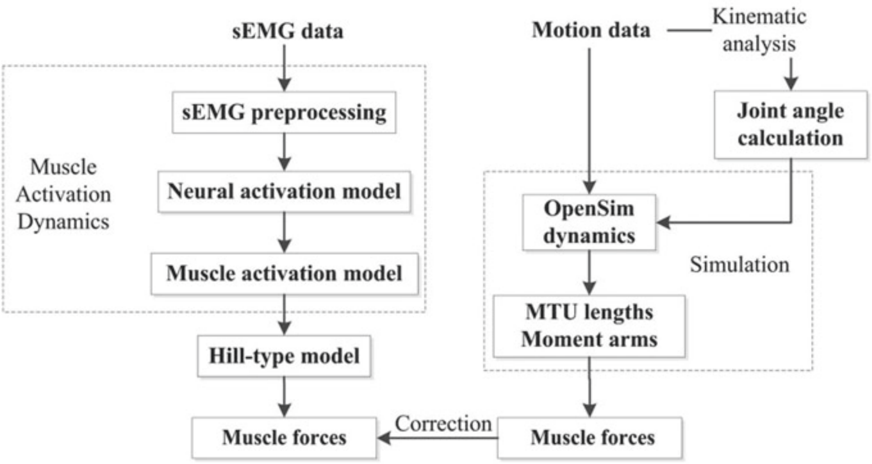


Fig. 12.1 The flowchart of our muscle force prediction method

acquisition system. The signals from 7 muscles were recorded: the short head of the biceps brachii (BICshort), long head of the biceps brachii (BIClong), brachialis (BRA), long head of the triceps brachii (TRIlong), lateral head of the triceps brachii (TRIlat), medial head of the triceps brachii (TRImed), and anconeus (ANC). The electrodes were placed longitudinally along the muscles in the direction of the muscle fibers and on the relevant part of each muscle according to the recommendations of the SENIAM (surface electromyography for the non-invasive assessment of muscles) project. A ground electrode was placed on the elbow joint (Fig. 12.2).

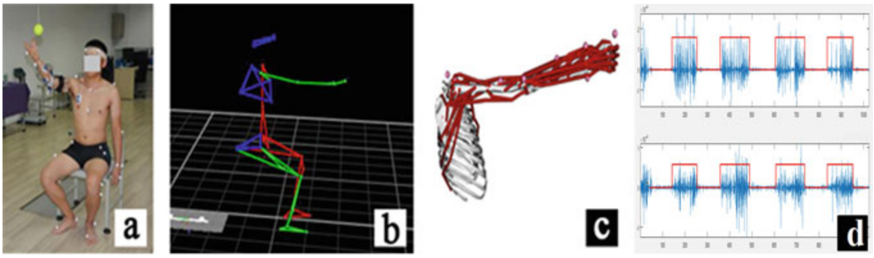


Fig. 12.2 Experimental setup. **a** Experimental setup for the actual tests, **b** Diagram from the motion capture system interface, **c** OpenSim skeletal model, **d** Raw EMG Signal

12.3.2 Joint Angle Estimation

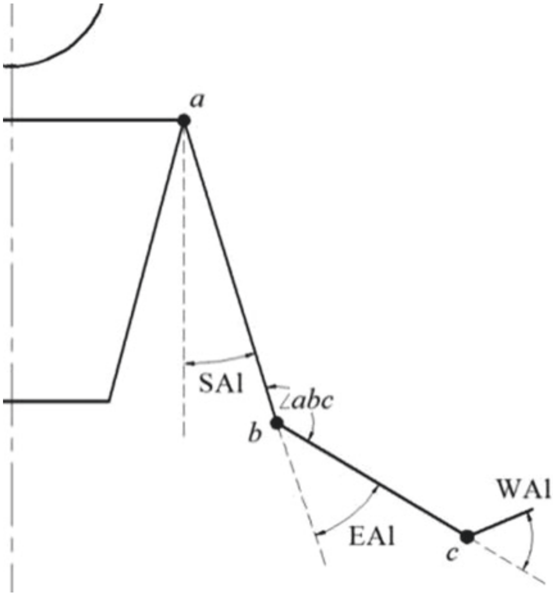
There are specific challenges concerning the collection of point data of human motion in an experiment. Although a human's limbs rotate around a single point within the skeletal structure, it is relatively difficult to maintain the joint at the zero-point throughout the testing process. To compensate for displacement of the joint, the space vector method was used to calculate the relative position in space and relative angular difference between two dependent point-lines. A space vector is a relative coordinate system, and the variation of the zero-point position can therefore be ignored. The human body parts were simplified to form a stick model and calculate the angle of the joint. A unique 3D coordinate positional system was established, and the value for n points for volunteer m at time t was collected. One vector segment was defined by two points in space, and the angle was obtained by measuring the relative positions of the two vector line segments (Fig. 12.3).

The angle of the left shoulder joint (SAl), the angle of the left elbow joint (EAl), and the angle of the left wrist joint (WAl) are shown in this diagram.

Suppose that a, b, c are three points in space; then, ∠abc represents the angle of joint b, and two vectors ending at a, and c are defined, which correspond to two marker points on volunteer m. The solution to the angle is as follows:

$$\cos\theta = \frac{\bar{M}^m l(a-b) \times \bar{M}^m l(c-b)}{\left|\bar{M}^m l(a-b)\right| * \left|\bar{M}^m l(c-b)\right|} \tag{12.1}$$

Fig. 12.3 Upper limb joint angle calculation definition



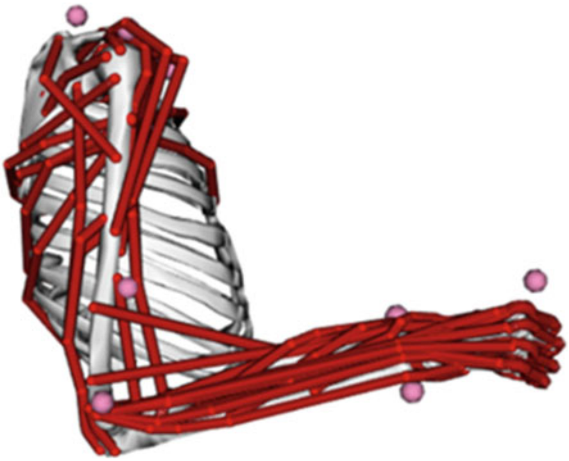


Fig. 12.4 Human upper limb musculoskeletal model generated in OpenSim

$$\Rightarrow \theta = ar\cos\theta.$$

12.3.3 OpenSim Simulation

OpenSim [37] skeletal muscle simulation software was used to generate a dynamic simulation. An OpenSim upper limb musculoskeletal model [38] developed by Saul et al. consisting of 7 body segments and 32 muscles was used to generate a simulation relative to the kinematic data, and muscle kinematics parameters, such as the musculotendon unit (MTU) lengths and moment arms, were obtained. This upper limb musculoskeletal model is shown in Fig. 12.4. First, scaling was carried out to calibrate the model to the subject according to the subject's anthropometric parameters. Inverse kinematics was then used to reconcile the differences in values between the actual 3D coordinates and the simulated virtual marker points. This process was achieved by a weighted least-squares method, which reduced the values to the minimum values possible. Last, dynamic optimization was carried out on the muscle forces in the main muscle group during the upper limb movement, which was performed in the simulation.

12.3.4 Muscle Activation Dynamics

(1) Data preprocessing

First, the original sEMG signal was preprocessed. The preprocessing phase mainly included three steps: (a) 50 Hz notch filtering to remove power

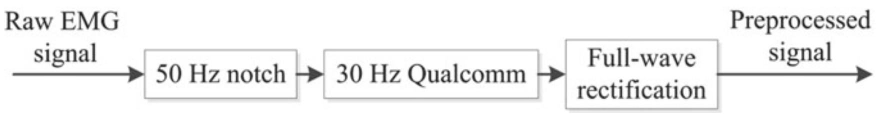


Fig. 12.5 Preprocessing phase for an sEMG signal

frequency interference; (b) 30 Hz zero-phase high-pass filtering to remove motion artefacts; and (c) full-wave rectification, which involves taking the absolute value of the signal (Fig. 12.5).

(2) Low-pass filter
The low-pass filter used was a 5 Hz zero-phase low-pass filter, which is a low-pass filter commonly used to smooth muscle signals.

(3) Normalization
The same method (data preprocessing → low-pass filter) was used to process the sEMG signal at maximal voluntary contraction (MVC) and identify the maximum value of the sEMG signal at MVC, which was considered 100% of the magnitude of the muscle activation signal. The normalized signal e(t) was obtained by dividing the processed myoelectric signal (data preprocessing → low-pass filter) recorded during normal motion by the maximum value.

(4) Neural activation model
EMG is a measure of the electrical activity that spreads across the muscle, causing it to activate. This process results in the production of a muscle force. However, it takes time for the force to be generated—it does not happen instantaneously. Thus, we adopted a second-order discrete linear model [39] to model the neural activation from muscle excitation obtained through preprocessing in the form of a recursive filter:

$$u(t) = \alpha e(t-d) - (c_1 + c_2)u(t-1) - c_1 c_2 u(t-2), \tag{12.2}$$

where *e(t)* is the muscle excitation at time *t*, *u (t)* is the neural activation, α is the muscle gain, c_1 and c_2 are recursive coefficients, and *d* is the electromechanical delay.

(5) Muscle activation model

The neural activations were then adjusted to account for either a linear or non-liasdnear EMG-force relationship [40]:

$$a(t) = \frac{e^{Au(t)} - 1}{e^{A} - 1}, \tag{12.3}$$

where *a(t)* is the muscle activation, *u(t)* is the neural activation, and *A* is the non-linear shape factor.

After obtaining the muscle activation *a(t)*, we computed the muscle force by integrating a Hill-type muscle model consisting of two elements: a contractile element

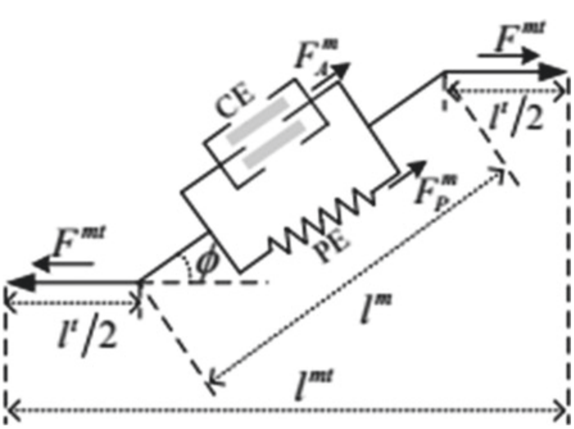


Fig. 12.6 Analysis of the Hill-type model mechanics

producing the active muscle force F_A^m and a parallel elastic element producing the passive force F_P^m. As shown in Fig. 12.6 [41], l^m is the muscle fiber length, l^t is the total length of the tendons, and φ is the pennation angle. Thus, the musculotendon length l^{mt} can be expressed as follows.

The muscle-tendon force (F^{mt}) is calculated as

$$F^{mt} = \left(f_A(l) \cdot f_v(v) \cdot a(t) + f_p(l)\right) \cdot F_o^m \cos(\phi), \tag{12.4}$$

where $l = l^m / l_o^m$,$v = v^m / v_o^m$, *a(t)* is the muscle activation, F_o^m is the maximum isometric muscle force, l_o^m represents the optimal fiber length, v_o^m is the maximum muscle contraction velocity, l is the normalized muscle fiber length, and v is the normalized muscle fiber velocity. $f_A(l)$, $f_V(v)$, and $f_P(l)$ define the normalized active force-length relationship, force-velocity relationship, and the normalized passive elastic force-length relationship, respectively.

12.4 Results

Using the above formula, we can calculate the relative values of the extension angles between the wrist, elbow, and shoulder joints when the upper limb of the human body performs the exercise. The joint angle data and point data were imported into OpenSim, the steps and parameters of the model described in the previous section were followed, and the upper limb model was run to simulate the motion and the changes in muscle forces in the short head of the biceps brachii (BICshort), long head of the biceps brachii (BIClong), brachialis (BRA), long head of the triceps brachii (TRIlong), lateral head of the triceps brachii (TRIlat), medial head of the triceps brachii (TRImed), and anconeus (ANC) (Fig. 12.7).

A surface electromyography signal acquisition device was used, and the sampling frequency was 1000 Hz. The surface electromyography signals of the seven muscles

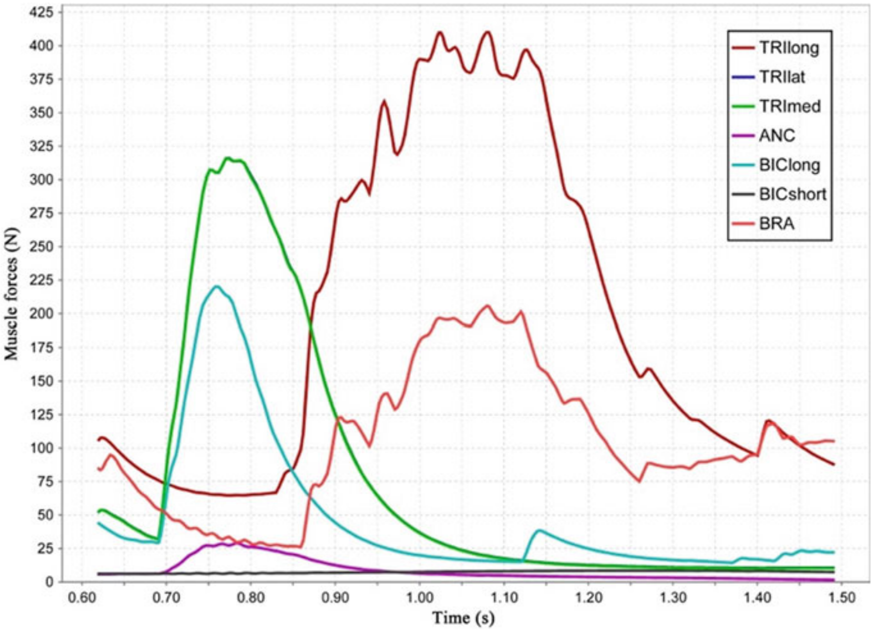


Fig. 12.7 Human upper limb muscle forces simulation generated by OpenSim

(the short head of the biceps brachii (BICshort), long head of the biceps brachii (BIClong), brachialis (BRA), long head of the triceps brachii (TRIlong), lateral head of the triceps brachii (TRIlat), medial head of the triceps brachii (TRImed), and anconeus (ANC)) involved in the movement of the upper arm of the human body were collected. The original signals were preprocessed and substituted into the muscle activation values obtained by formula (12.3), the muscle force predictive values were obtained by substituting the muscle activations into formula (12.4), and the above data were calculated with MATLAB R2014b. Figure 12.8 shows the changes in the muscle force of the brachialis.

We compared the force of the same muscles with the predicted values calculated by OpenSim and sEMG, and the curves were very close, as shown in Fig. 12.9. With a statistical analysis, the muscle force values of the other six muscles were also compared, showing a strong correlation ($P < 0.05$). The comparative trial in this paper also verified the feasibility of predicting muscle forces by sEMG.

12.5 Discussion

Calculating joint angles and muscle forces from motion capture data is a simple process. There are many formulas and simulation software available, but the shortcomings of motion capture experiments are that the space required for experiments

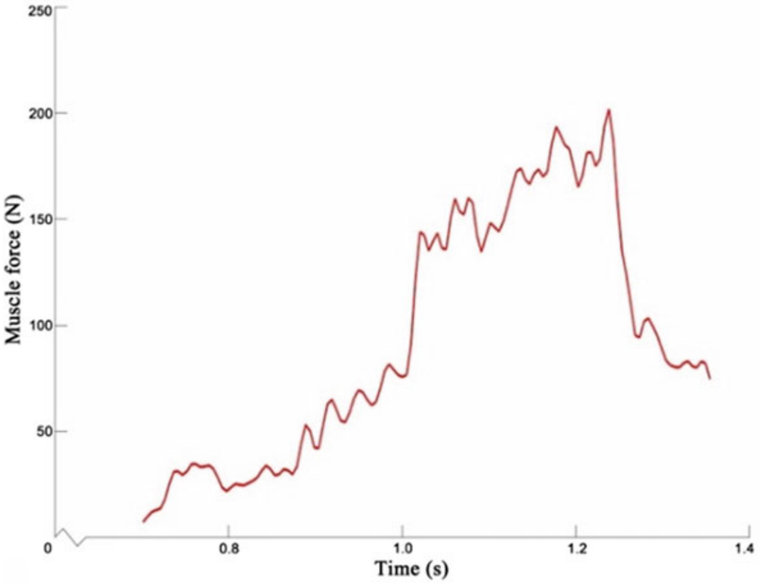


Fig. 12.8 Muscle force of the brachialis according to the sEMG prediction

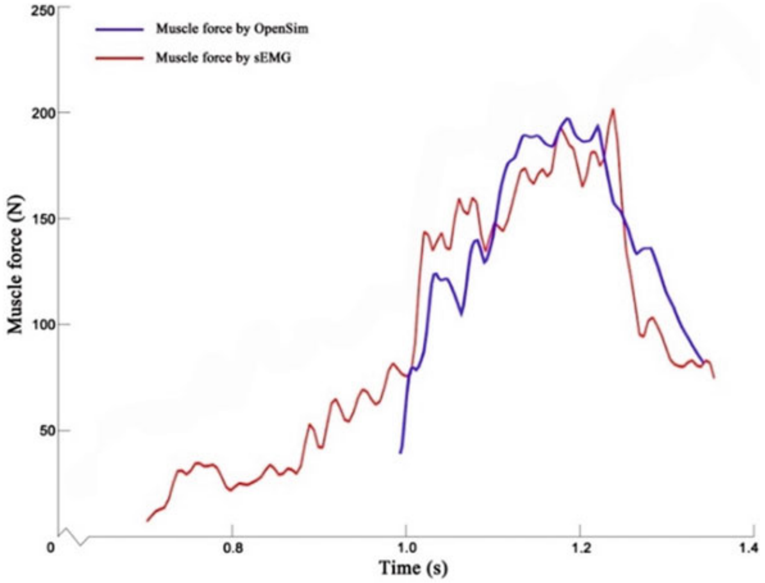


Fig. 12.9 Comparison of the muscle force values obtained from OpenSim and sEMG

is too large and the experimental process is cumbersome. Motion capture data can be used for offline scientific experiments but are unreliable when used to generate real-time or online active control signals, such as those applied to an upper limb robotic exoskeleton. The advantages of surface EMG include the facts that the acquisition process is simple, the associated experimental equipment is small and portable, and real-time signals can be acquired and generated for control, but EMG signals are weak, and the process of calculating and processing the signals is complicated. Therefore, an experiment for the synchronous acquisition of motion capture data and sEMG signals was carried out to verify the sEMG signal calculation results obtained by using the calculated motion capture data. The final verification results also suggest the feasibility of using EMG signals to calculate muscle forces.

As upper extremity robotic exoskeletons and rehabilitation robots continue to develop, pattern recognition, which is an offline control method, does not meet the needs of practical applications. A control source signal requires real-time acquisition control, which requires the acquisition process to be simple and easy to perform and the signal to be stable and continuous. sEMG signals can meet these demands. This paper studies the dynamic evaluation mechanism of human upper limb movement, which was designed to convert offline motion capture calculations to online electromyography calculations. Real-time calculations of muscle force can provide more accurate control of robotic exoskeletons and real-time evaluations of upper limb motion states. In the future, sEMG signal changes and upper limb joint angles should be analyzed more deeply to detect changes in the joint through the surface EMG signal; then, robotic exoskeletons or rehabilitation robots can be controlled in real time by surface EMG signals.

Several limitations should be noted. First, uncertain noises in EMG signals still existed, even if we tried to avoid it, such as cross-talk from other muscles, baseline noise, and artifact. Due to difference of individual physiologic and electrode positions, the outputs will be a little bit different. But it can still serve as a reference in rehabilitation.

12.6 Conclusions

Multi-parameter human–computer interaction technology is an important new development in the field of human physics and neuro-rehabilitation. This study proposes a set of upper limb kinematic analysis methods, which include muscle force prediction methods. In addition, this work can provide reference values for the evaluation of upper limb motor function and the auxiliary control of upper limb rehabilitation robots. Accurate muscle force prediction methods can be used to assess an individual's ability to generate limb movements, which can promote a deeper understanding of the condition of the patient's nervous system. This knowledge can be used to guide the selection of rehabilitative treatments and to design better rehabilitation robots that can assist people with upper extremity dyskinesia during upper limb tasks.

Acknowledgements Supported by National Natural Science Foundation of China (Grant No. 51865056), and Xi'an Jiaotong University State Key Laboratory for Manufacturing Systems Engineering (Grant No. sklms2018006).

References

1. World Health Organization. World health statistics 2018, Available online. http://www.who.org/
2. Perez R, Costa U, Torrent M, Solana J, Opisso E, Caceres C, Tormos JM, Medina J, Gomez EJ (2017) Upper limb portable motion analysis system based on inertial technology for neurorehabilitation purposes. Sensors 10:10733–10751
3. Rahman MH, Rahman MJ, Cristobal OL, Saad M, Kenné JP, Archambault PS (2015) Development of a whole arm wearable robotic exoskeleton for rehabilitation and to assist upper limb movements. Robotica 33:19–39
4. Albert CL, Peter DG, Lorie GR, Jodie KH, George FW, Daniel GF, Robert JR, Todd HW, Hermano IK, Bruce TV, Christopher TB, Dawn MB, Pamela WD, Barbara HC, Alysia DM, Stephen EN, Susan SC, Janet MP, Grant DH, Peter P (2010) Robot-assisted therapy for long-term upper-limb impairment after stroke. New England J Med 362:1772–1783
5. Farulla GA, Pianu D, Cempini M, Cortese M, Russo LO, Indaco M, Nerino R, Chimienti A, Oddo CM, Vitiello N (2016) Vision-based pose estimation for robot-mediated hand telerehabilitation. Sensors 16:208
6. Borboni A, Maddalena M, Rastegarpanah A, Saadat M, Aggogeri F (2016) Kinematic performance enhancement of wheelchair-mounted robotic arm by adding a linear drive. 2016 IEEE international symposium on medical measurements and applications (MeMeA)
7. Zhang L, Zheng Z, Li G, Sun Y, Jiang G, Kong J, Tao B, Xu S, Yu H, Liu H (2018) Tactile sensing and feedback in SEMG hand. Int J Comput Sci Math 9(4):365–376
8. Rzyman G, Szkopek J, Redlarski G, Palkowski A (2020) Upper limb bionic orthoses: general overview and forecasting changes. Appl Sci 10(15):5323
9. Meattini R, Chiaravalli D, Palli G, Melchiorri C (2020) sEMG-based human-in-the-loop control of elbow assistive robots for physical tasks and muscle strength training. IEEE Robot Autom Lett 5(4):5795–5802
10. Xu G, Song A, PanL, Li H, Liang Z, Zhu S, Xu B, Li J (2012) Adaptive hierarchical control for the muscle strength training of stroke survivors in robot-aided upper-limb rehabilitation. Int J Adv Robot Syst, https://doi.org/10.5772/51035
11. Hogan N, Krebs HI, Charnnarong J, Srikrishna P, Sharon A (1993) MIT-MANUS: a workstation for manual therapy and training II[P]. Other Conferences
12. Gassert R, Dietz V (2018) Rehabilitation robots for the treatment of sensorimotor deficits: a neurophysiological perspective. J NeuroEng Rehab, https://doi.org/10.1186/s12984-018-0383-x
13. Lenzi T, De Rossi SMM, Vitiello N, Carrozza MC (2012) Intention-based EMG control for powered exoskeletons. IEEE Trans Bio-Med Eng 59(8)
14. Shing LH, Quan XS (2011) Exoskeleton robots for upper-limb rehabilitation: state of the art and future prospects[J]. Med Eng Phys 34(3)
15. Hao L, Jun T, Pan L (2018) Human-robot cooperative control based on sEMG for the upper limb exoskeleton robot. Robot Auton Syst
16. He L, Xiong C, Liu K, Huang J, He C, Chen WB (2018) Mechatronic design of a synergetic upper limb exoskeletal robot and wrench-based assistive control. J Bionic Eng 15:247–259
17. Nelson CA, Nouaille L, Poisson G (2020) A redundant rehabilitation robot with a variable stiffness mechanism. Mech Mach Theory 150

18. Leiyu Z, Jianfeng L, Ying C, Mingjie D, Bin F, Pengfei Z (2020) Design and performance analysis of a parallel wrist rehabilitation robot (PWRR). Robot Auton Syst 125(C)
19. Nelson Carl A, Laurence N, Gérard P (2019) A redundant rehabilitation robot with a variable stiffness mechanism. Mechan Mach Theory 150
20. Ning Y, Xu W, Huang H, Li B, Liu F (2019) Design methodology of a novel variable stiffness actuator based on antagonistic-driven mechanism. Proc Inst Mech Eng Part C: J Mech Eng Sci 233(19–20)
21. Meshram DA, Patil DD (2020) 5G Enabled tactile internet for tele-robotic surgery. Procedia Comput Sci 171
22. Platz T, Pinkowski C, Van WF, Wijck FV, Kim IH, Bella PD, Johnson G (2005) Reliability and validity of arm function assessment with standardized guidelines for the Fugl-Meyer test, action research arm test and box and block test: a multicentre study. Clin Rehab 19:404–411
23. Mehrholz J, Pohl M, Platz T, Kugler J, Elsner B (2018) Electromechanical and robot-assisted arm training for improving activities of daily living, arm function, and arm muscle strength after stroke. Cochrane Database Syst Rev, https://doi.org/10.1002/14651858.cd006876.pub5
24. Proud EL, Miller KJ, Bilney B, Balachandran S, McGinley JL, Morris ME (2015) Evaluation of measures of upper limb functioning and disability in people with Parkinson disease: a systematic review. Arch Phys Med Rehab 96:540–551
25. Boser QA, Valevicius AM, Lavoie EB, Chapman CS, Pilarski PM, Hebert JS, Vette AH (2018) Cluster-based upper body marker models for three-dimensional kinematic analysis: comparison with an anatomical model and reliability analysis. J Biomech 72:228–234
26. Caimmi M, Guanziroli E, Malosio M, Pedrocchi N, Vicentini F, Molinari Tosatti L, Molteni F (2015) Normative data for an instrumental assessment of the upper-limb functionality. Biomed Res Int 484131
27. Hebert JS, Lewicke J, Williams TR, Vette AH (2014) Normative data for a modified box and blocks test measuring upper limb function via motion capture. J Rehab Res Dev 51:919–932
28. Sohn WJ, Sipahi R, Sanger TD, Sternad D (2019) Portable motion-analysis device for upper-limb research, assessment, and rehabilitation in non-laboratory settings. IEEE J Trans Eng Health Med 7:1–14
29. Vigotsky AD, Halperin I, Lehman GJ, Trajano GS, Vieira TM (2018) Interpreting signal amplitudes in surface electromyography studies in sport and rehabilitation sciences. Front Physiol 108:227–237
30. Goble JA, Zhang Y, Shimansky Y, Sharma S, Dounskaia NV (2007) Directional biases reveal utilization of arm's biomechanical properties for optimization of motor behavior. J Neurophysiol 98:1240–1252
31. William HJ, Shivam P, Priyanshu A, Sadie HR, Bowie RL, Michael DB, Marcia KO'M (2018) Toward improved surgical training: Delivering smoothness feedback using haptic cues. 2018 IEEE haptics symposium (HAPTICS), pp 241–246
32. Zhang LL, Zhou J, Zhang XA, Wang CT (2011) Upper limb musculo-skeletal model for biomechanical investigation of elbow flexion movement. J Shanghai Jiaotong Univ (Science) 16:61–64
33. Ma R, Zhang L, Li G, Jiang D, Xu S, Chen D (2020) Grasping force prediction based on sEMG signals. Alexandria Eng J 59(3):1135–1147
34. Dai C, Hu X (2019) Extracting and classifying spatial muscle activation patterns in forearm flexor muscles using high-density electromyogram recordings. Int J Neural Syst, https://doi.org/10.1142/s0129065718500259
35. Ursula T, Hermann S, Richard B, Nathalie A (2019) Muscle force estimation in clinical gait analysis using AnyBody and OpenSim. J Biomech 86:55–63
36. Malesevic N, Björkman A, Andersson GS, Matran-Fernandez A, Citi L, Cipriani C, Antfolk C (2020) A database of multi-channel intramuscular electromyogram signals during isometric hand muscles contractions. Scientific Data, https://doi.org/10.1038/s41597-019-0335-8
37. Delp SL, Anderson FC, Arnold AS, Loan P, Habib A, John CT, Guendelman E, Thelen DG (2007) OpenSim: open-source software to create and analyze dynamic simulations of movement. IEEE Trans Biomed Eng 54(11):1940–1950

38. Saul KR, Hu X, Goehler CM, Vidt ME, Daly M, Velisar A, Murray WM (2015) Benchmarking of dynamic simulation predictions in two software platforms using an upper limb musculoskeletal model. Comput Methods Biomech Biomed Eng 18:1–14
39. David GL, Thor FB (2003) An EMG-driven musculoskeletal model to estimate muscle forces and knee joint moments in vivo. J Biomech 36:765–776
40. Christian F, Günter H (2008) A human–exoskeleton interface utilizing electromyography. IEEE Trans Rob 24:872–882
41. Jiateng H, Yingfei S, Lixin S, Bingyu P, Zhipei H, Jiankang W, Zhiqiang Z (2018) A pilot study of individual muscle force prediction during elbow flexion and extension in the neurorehabilitation field. Sensors 16:2–15

Chapter 13
Resting-State EEG Sex Classification Using Selected Brain Connectivity Representation

Jean Li, Jeremiah D. Deng, Divya Adhia, and Dirk De Ridder

13.1 Introduction

Electroencephalography (EEG) is a widely used non-invasive technique to measure multi-channel potentials that reflect the electrical activity of the brain. Over the last few decades, EEG analysis has been an intensively explored research topic due to its potentials in being applied to the diagnosis of neurological diseases, such as epilepsy, brain tumors, head injury, sleep disorders, and dementia [19]. Despite many advances made in recent years, EEG signal analysis remains a challenging task. In addition to being non-stationary, EEG signals often have high noise-to-information ratios, and they can be significantly affected by various artifacts, demonstrating characteristics that differ from signals generated by activities in the brain [21]. Common artifacts include eye movements, jaw tension, and muscle contractions. To make effective signal analysis even more challenging, EEG signals are highly individual-specific, and cross-subject pattern identification can be elusive.

In a more proactive approach, EEG can also be applied to biofeedback training as an operant conditioning technique to reinforce or inhibit specific forms of EEG activities. It has been used in anxiety and addiction treatment, also employed for attentional, cognitive, and psychosocial functioning improvement [17]. It is noted that

J. Li · J. D. Deng (✉)
Department of Inforamtion Science, University of Otago, Dunedin, New Zealand
e-mail: jeremaih.deng@otago.ac.nz

J. Li
e-mail: jean.li@postgrad.otago.ac.nz

D. Adhia · D. De Ridder
Department of Surgical Sciences, University of Otago, Dunedin, New Zealand
e-mail: divya.adhia@otago.ac.nz

D. De Ridder
e-mail: dirk.deridder@otago.ac.nz

T. D. Pham et al. (eds.), *Advances in Artificial Intelligence, Computation, and Data Science*, Computational Biology 31,
https://doi.org/10.1007/978-3-030-69951-2_13

popular EEG biofeedback treatment is largely based on sex-neutral protocols [17]. Our proposition is that if there are innate sex differences found in EEG signals, then it is possible for new sex-differentiated EEG biofeedback protocols to be developed to potentially enhance many neurological treatments.

This study focuses on using machine learning techniques to explore the evidence of sex effects on EEG signals. Rather than just examining the between-sex statistic differences, we attempt to construct an effective classification model to predict the sex of a subject through their EEG signals.

This chapter is organized as follows. In Sect. 13.2, we review related work and discuss some common approaches to represent EEG signals. Section 13.3 introduces the dataset used in this study, and the data preprocessing procedures, feature extraction and selection, and the classification methods. Then in Sect. 13.4, we present the results of sex classification. We conclude the chapter in Sect. 13.5.

13.2 Related Work

Within the neuroscience literature, there has been an ongoing interest in sex effects in cognitive performance and the underlying neural mechanisms [7]. While a number of studies failed to find sex differences in cognitive performance and hemispheric asymmetry [2, 16], some evidence of the sex effects on EEG has been found.

In [4], the EEG signals of 80 individuals between the ages of 8 and 12 years were analyzed. Differences between sex were found in this study group, with males having less theta but more alpha frequency components than females. Females were also found to have a developmental lag in the EEG compared to males. An earlier work [5] also reported sex differences in EEG asymmetry during self-generated cognitive and affective tasks. However, both of these studies did not validate the findings on external test individuals, therefore, the generality of the findings may be limited.

The study in [3] investigated the effects of age and sex on sleep EEG power spectral density of individuals of age ranging 20–60 years. The average power density within the 4-sec epochs was calculated. It was found that females show significantly higher spectral power density in some power bands than males. Though significant effects of age on sleep EEG spectral power density were found, the study did not find any interaction between age and sex. This study performed robust statistical analysis for longitudinal data. However, no external data validation was performed to demonstrate the generalization ability on new subjects.

Although EEG signal classification has been widely explored for different purposes, there have been few studies that investigate sex differences in EEG and attempt EEG-based sex classification [11]. For classification purposes in general, the EEG signals are typically preprocessed by band-pass filters and spatial filters for feature extraction. Most commonly, frequency band power features and time-point features are employed to represent EEG signals. Band power features represent the average energy level of EEG signals within a certain frequency range over a given time window called an “epoch”. Band power features need to be extracted, respectively, in each channel. Time-point features are a concatenation of EEG signals from all channels, and they are typically used for event-related potential classifications [12]. To cope with the non-stationarity of EEG signals, band power features are usually extracted from a reasonably short epoch. For example, in [3], epochs are extracted using a sliding window with a length of 2 s.

Spatial filters were also applied in other studies for EEG feature extraction. These can be obtained in a supervised manner, such as Common Spatial Patterns (CSP). CSP projects the signals into another matrix space that maximizes the distance between 2 classes. Reference [12] discusses the effectiveness of this approach and has proven it to be useful. Spatial filters can also be obtained through an unsupervised way such as Independent Component Analysis (ICA). In addition to the above, other EEG representing methods are also studied, which include sparse representation and deep learning. The sparse representation-based classification (SRC) method has shown a robust classification performance [18]. In deep learning, the features and the classifier are jointly learned directly from the EEG signals. The convolutional neural networks and restricted Boltzmann machines are the two most popular deep learning methods for EEG-based Brain-Computer Interface (BCI) studies [12].

Correlation between EEG channels, also known as coherence, was evaluated in time domain to analyze the connectivity patterns in dystonia patients [1]. Coherences have the advantage of yielding the possibility to recognize motor-imagery-related activation even without typical activation observed, often giving small standard deviations [13]. To better handle non-stationarity, connectivity has been modeled by coherence obtained from spectral features obtained from FFT [8, 13–15]. Connectivity based on spectral coherence is also found to be an effective biometric feature [10].

More recently, [20] utilized deep learning to predict sex through EEG signals. With a large data size (1000 adults) and deep convolutional nets, an accuracy of 81% was achieved. It shows that the beta band provides the most important features in predicting sex. Another deep-learning-based study [22] assesses gender differences in emotion processing EEG data and reports a classification accuracy as high as 95% using gamma-band features. The dataset contains however only 60 subjects.

13.3 Data and Methods

13.3.1 Dataset Description

The dataset used for this study is part of the data collected for a previous work [21], containing a raw resting-state EEG stream set of 241 healthy individuals only. The raw EEG signal was collected through a standard Mitsar amplifier with 19 channels (Fp1, Fp2, F7, F3, Fz, F4, F8, T7, C3, Cz, C4, T8, P7, P3, Pz, P4, P8, O1, and O2) with a sampling rate of 250 Hz. The studied population consists of 150 females and 91 males aged between 17 and 89 years. The distribution regarding the age and sex

Table 13.1 Age and sex distribution of the sample set

Age (years)	Females	Males
17–25	35	20
26–35	17	9
36–45	36	19
46–55	34	21
56–65	16	15
66–75	8	5
76–89	4	2
Sum	150	91

of the population can be found in Table 13.1. Note that there is a notable imbalance between the two sexes across several age groups.

13.3.2 Preprocessing

Since EEG signals cannot be segmented into physiologically relevant units, the conventional approach of segmenting the EEG streams into epochs according to a time interval is adopted in this study. In this study, we segmented the raw EEG streams into 2-s epochs. The first 5 s of every EEG recording were discarded to avoid possible noise.

13.3.3 Signal Representation

In this study, the raw EEG signals of each subject are represented by the spectral connectivity between channel pairs. The MNE package[1] was applied to compute the frequency-domain connectivity measures. In particular, for every pair of channels, the coherence across all epochs in each frequency was computed, as in Eq. 13.1:

$$C = \frac{|E[S_{xy}]|}{\sqrt{E[S_{xx}] \times E[S_{yy}]}} \tag{13.1}$$

where S_{xy} is the cross-spectral density between x and y, and $E[]$ denotes average over epochs.

To shorten the feature vector length, the mean value of the connectivity within 5 major brain wavebands is adopted. These frequency bands are 0–4 Hz (delta), 4–8

[1] Available from https://mne.tools/stable/generated/mne.connectivity.spectral_connectivity.html.

Hz (theta), 8–13 Hz (alpha), 13–30 Hz (beta), and 30–45 Hz (gamma). This gives us a feature vector of length 855 (171 channel pairs × 5 bands).

13.3.4 Feature Analysis

Compared with the relatively small sample size of 241, the feature vector of length 855 may cause a potential overfitting problem. To further shorten the connectivity representation, training subject splitting and feature selection were adopted.

For each training subject, we split the subject's entire EEG recordings into several 30 s long (15 epochs) sections, and use each section as an independent training sample. Considering the imbalance of the dataset, we take 5 s from each female and 8 from each male to form a more balanced training set. This approach results in a training set with a 5 to 8 times larger sample size.

Feature selection is carried out using XGBoost [6]. In each of the 50 trials, we randomly selected 90% of the subjects, split the recordings as described above, and fit these data points into an XGBoost binary classifier. We then rank all 855 features according to the feature importances given by XGBoost and stack the ranks from the 50 trials together. Since the feature importance ranking varies across trials, we generate our own feature importances through the 855 × 50 matrix. We iterate from the most to least important XGBoost ranks, the earlier a feature appears in all 50 trials, the more important it is marked. We then select the top 34 connectivity features according to our ranking as the final representation. Details of the feature selection outcome will be demonstrated in Sect. 13.4.1.

13.3.5 Classification

Four different classifiers are applied separately using the chosen representation. They are XGBoost classifier (XGB), multi-layer perceptron (MLP), support vector machine (SVM), and the random forest classifier (RF).

Validation scores were generated by running 50 trials. In each of the 50 trials, we randomly select 90% of the subjects and split these recordings as described in Sect. 13.3.4 to form a reasonably balanced training set. The other 10% of subjects are used for validation. For the subjects in the validation set, EEG streams are not split (i.e. no repeated validation subjects). To increase the stability, we take a longer recording length of 3 min (90 epochs) to compute the connectivity and choose the 34 features we selected as the final representations of the validation subjects.

For binary classification, depending on the classification algorithm, a crisp classification decision is made (such as in decision trees) or a 0.5 threshold value is used, e.g. in the output node of an MLP. We use a unified probabilistic framework here, by estimating the probability of a classification outcome, and using a relaxed threshold value to decide the classification outcome c:

$$c = \begin{cases} 1, \text{ if } y > \theta \\ 0, \text{ otherwise} \end{cases}, \tag{13.2}$$

where y is the predicted probability, and θ is the probability threshold which can be optimized through cross-validation.

13.4 Results

13.4.1 Feature Selection

Figure 13.1 demonstrates the 50 trial validation result from a binary XGBoost classifier using different numbers of features. The classifier reached the best performance when using the top 34 features. As shown in the figure, the mean score displays an increasing trend until the number of features reaches 34.

The chosen 34 features are illustrated in topographical maps in Fig. 13.2. The brain connectivity in beta and gamma bands seems more prominent than those in other bands when performing sex classification tasks, as more connectivity pairs in these two bands are marked as important features. The gamma-band connectivity between channels Fp1 and Fz is ranked as the most important feature. The connectivity between channels F7 and F8 in the beta band is the sixth important feature. These two important pairs are in agreement with a previous neuroscience study [9],

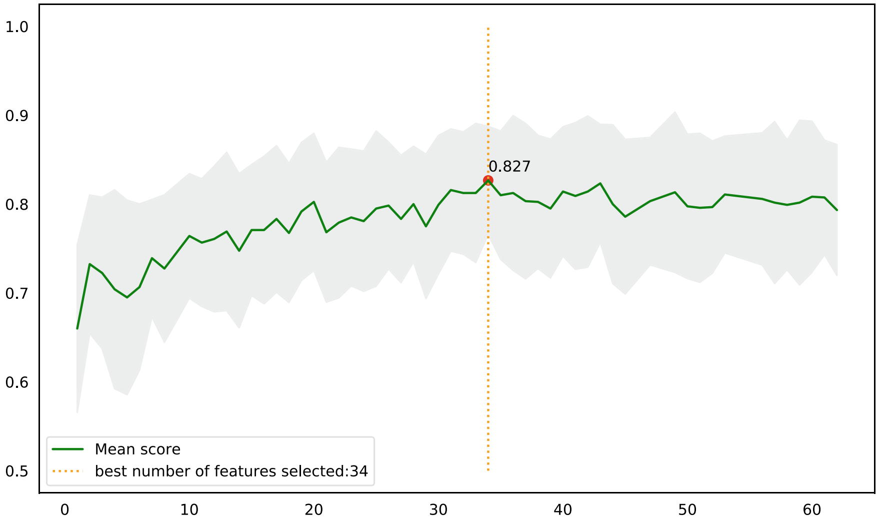


Fig. 13.1 Validation result of XGBoost using increasing numbers of chosen features

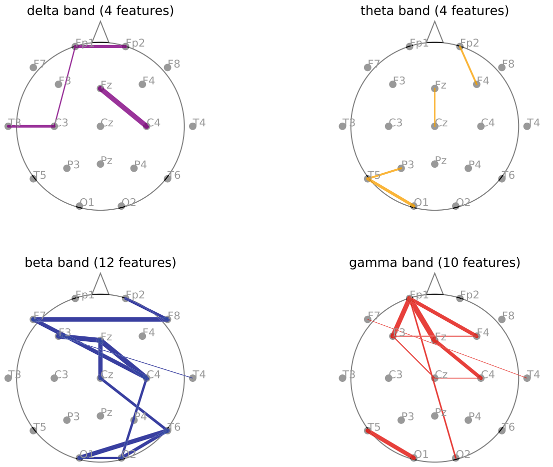


Fig. 13.2 Chosen connectivity features in five frequency bands. Thicker lines between the channels suggest that the connectivity values between these channel pairs are more important features

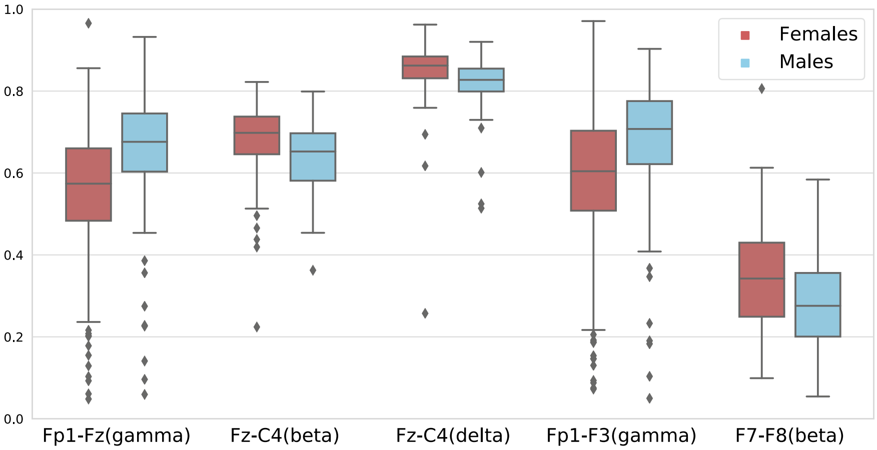


Fig. 13.3 Box plots of some top connectivity features display promising separability of the two sex classes

in that females have more latero-lateral interhemispheric connectivity (F7–F8) and males more antero-posterior intrahemispheric (Fp1–Fz) connectivity. The connectivity between channels Fz and C4 appears to be important in three different bands: beta (ranked 2nd), delta (ranked 3rd), and gamma (ranked 13th). We speculate that the Fz–C4 connectivity in beta could be related to the known sex differences in the anterior cingulate cortex (Fz) and sensorimotor area (C4). Figure 13.3 shows that some of the top connectivity features display promising separability of the two sex classes.

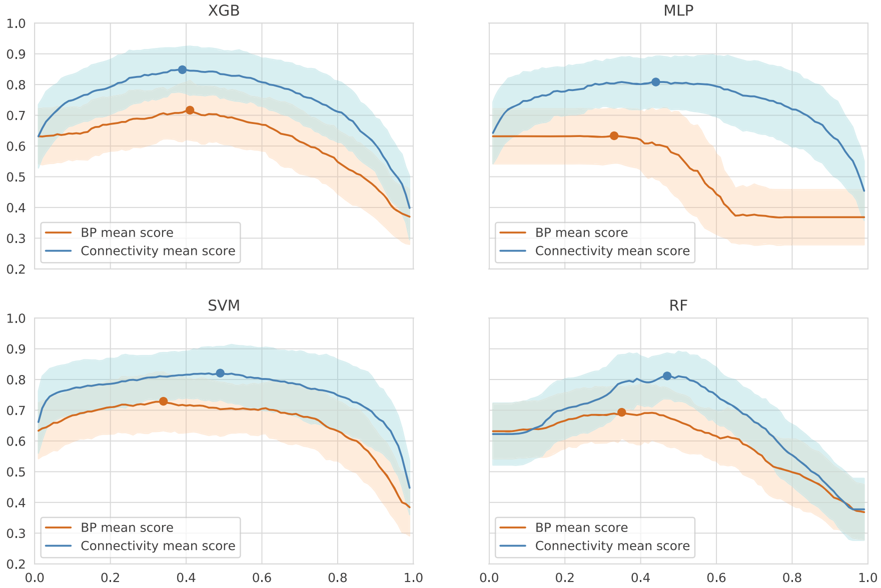


Fig. 13.4 Validation accuracy versus the threshold ranged in (0, 1). Shaded areas indicate the accuracy values within 1 stdev of the corresponding mean score

13.4.2 Validation Results

Figure 13.4 illustrates the 50 trial validation scores across increasing thresholds when using XGB, MLP, SVM, and RF to predict the probability of the female class. The connectivity scores are generated using the chosen 34 connectivity features. The band power scores are generated by using 31 band power features which are selected in a similar way to the connectivity features. All classifiers demonstrate that using connectivity features outperforms using band power features. When using the selected connectivity features, all four classifiers reach their best performances when the threshold of the probability is set around 0.4–0.5. XGB gives the highest accuracy score of 0.848 using a threshold of 0.39.

The mean area under the curve (AUC) of XGB using selected connectivity features across 50 trials is 0.89 (±0.07), as shown in Fig. 13.5. The mean AUCs of other classifiers are SVM—0.88 (±0.07), MLP—0.86 (±0.08), and RF—0.88 (±0.08). The high AUCs from classifiers suggest the generally good quality of the chosen features.

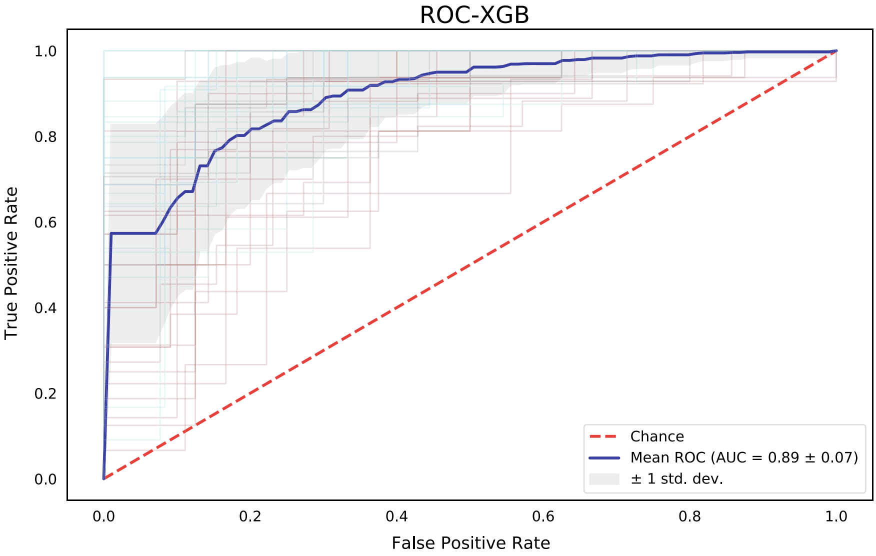


Fig. 13.5 Average ROC curves obtained through cross-validations of XGB

13.4.3 Test Results

An independent set comprised of 40 subjects (25 females and 15 males) was used for testing. The test scores across increasing thresholds are shown in Fig. 13.6. MLP has the best score of 0.825 using a threshold of 0.47, and the best scores of XGB, SVM, and RF are all 0.8.

13.5 Conclusions

We have conducted a preliminary study on the potential sex differences in resting-state EEG signals using a machine learning approach. Instead of using a data-driven deep learning method which demands more subject data to avoid overfitting, we chose to focus on assessing the effects of band power and connectivity features in sex classification using classifier ensembles and feature analysis methods. In particular, it is found that female and male groups show different brain connectivity patterns which are most prominent in beta and gamma bands. The connectivity between channels Fp1 and Fz in the gamma band shows the greatest sex discrepancy. The connectivity between channels Fz and C4 appears to be different between sexes in the delta, beta, and gamma bands. The initial band power features are not included in the classification models because they are found to be collectively less important than

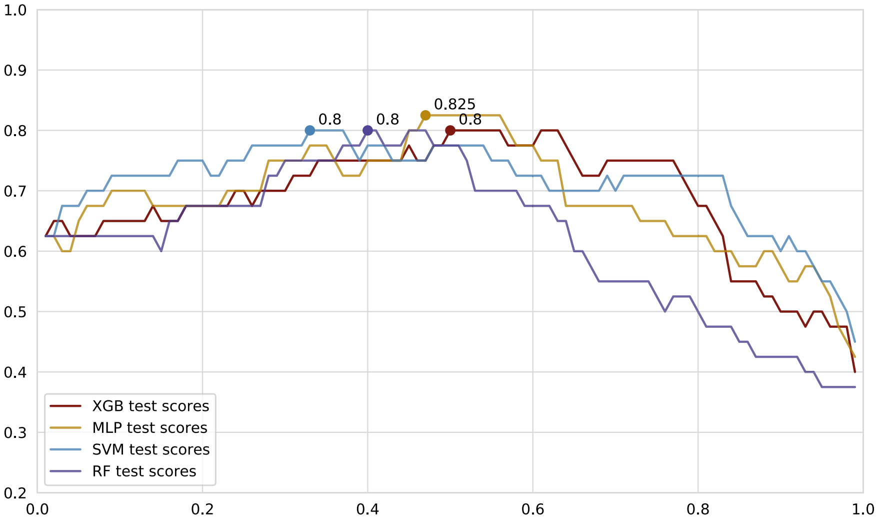


Fig. 13.6 Test accuracy versus the threshold ranged in (0,1). Best accuracy positions indicated by colored dots

connectivity features, even though they may contribute positively to the discriminant analysis on sexes.

Due to limited data availability, we have concentrated on examining healthy subjects' EEG signals. For future work, we would like to investigate sex differences on EEG connectivity with signals obtained under different pathological settings and explore its possible application in biofeedback therapy.

References

1. Baltazar CA, Machado BS, de Faria DD, Paulo AJM, Silva SMCA, Ferraz HB, de Carvalho Aguiar P (2020) Brain connectivity in patients with dystonia during motor tasks. J Neural Eng 17(5):056039
2. Beste C, Heil M, Konrad C (2010) Individual differences in erps during mental rotation of characters: lateralization, and performance level. Brain Cogn 72(2):238–243
3. Carrier J, Land S, Buysse DJ, Kupfer DJ, Monk TH (2001) The effects of age and sex on sleep eeg power spectral density in the middle years of life (ages 20–60 years old). Psychophysiology 38(2):232–42
4. Clarke AR, Barry RJ, McCarthy R, Selikowitz M (2001) Age and sex effects in the eeg: development of the normal child. Clin Neurophysiol 112(5):806–814
5. Davidson RJ, Schwartz GE, Pugash E, Bromfield E (1976) Sex differences in patterns of eeg asymmetry. Biol Psychol 4(2):119–137
6. Gramfort A, Luessi M, Larson E, Engemann D, Strohmeier D, Brodbeck C, Goj R, Jas M, Brooks T, Parkkonen L, Hämäläinen M (2013) Meg and eeg data analysis with mne-python. Front Neurosci 7:267

7. Hirnstein M, Hugdahl K, Hausmann M (2019) Cognitive sex differences and hemispheric asymmetry: a critical review of 40 years of research. Laterality 24(2):204–252 PMID: 29985109
8. Höller Y, Bergmann J, Thomschewski A, Kronbichler M et al (2013) Comparison of eeg-features and classification methods for motor imagery in patients with disorders of consciousness. PLoS One 8(11):e80479
9. Ingalhalikar M, Smith A, Parker D, Satterthwaite TD, Elliott MA, Ruparel K, Hakonarson H, Gur RE, Gur RC, Verma R (2014) Sex differences in the structural connectome of the human brain. Proc Natl Acad Sci 111(2):823–828
10. La Rocca D, Campisi P, Végső B, Cserti P, Kozmann G, Babiloni F, De Vico Fallani F (2014) Human brain distinctiveness based on eeg spectral coherence connectivity. IEEE Trans Bio-med Eng 61:2406–2412
11. Li J, Deng JD, Ridder DD, Adhia D (2020) Gender classification of EEG signals using a motif attribute classification ensemble. In: 2020 international joint conference on neural networks, IJCNN 2020, Glasgow, United Kingdom, July 19–24, 2020. IEEE, pp 1–8
12. Lotte F, Congedo M, Lécuyer A, Fabrice L, Arnaldi B (2007) A review of classification algorithms for eeg-based brain-computer interfaces. J Neural Eng 4:07
13. Markovic A, Kaess M, Tarokh L (2020) Gender differences in adolescent sleep neurophysiology: a high-density sleep eeg study. Sci Reports 10:15935
14. Mumtaz W, Ali SS, Yasin MAM, Malik A (2017) A machine learning framework involving eeg-based functional connectivity to diagnose major depressive disorder (mdd). Med Biol Eng 56:233–246
15. Murias M, Webb SJ, Greenson J, Dawson G (2007) Resting state cortical connectivity reflected in eeg coherence in individuals with autism. Biol Psychiatry 62(3):270–273. Mechanisms of Circuit Dysfunction in Neurodevelopmental Disorders
16. Papousek I, Murhammer D, Schulter G (2011) Intra- and interindividual differences in lateralized cognitive performance and asymmetrical EEG activity in the frontal cortex. Brain Cogn 75(3):225–231
17. Scott WC, Kaiser D, Othmer S, Sideroff SI (2005) Effects of an eeg biofeedback protocol on a mixed substance abusing population. Am J Drug Alcohol Abuse 31(3):455–69
18. Shin Y, Lee S, Ahn M, Cho H, Jun SC, Lee H-N (2015) Noise robustness analysis of sparse representation based classification method for non-stationary eeg signal classification. Biomed Signal Process Control 21:8–8
19. Puthankattil Subha D, Joseph PK, Rajendra AU, Lim CM (2010) Eeg signal analysis: a survey. J Med Syst 34(2):195–212
20. van Putten M, Olbrich S, Arns M (2018) Predicting sex from brain rhythms with deep learning. Sci Rep 8:12
21. Vanneste S, Song J, De Ridder D (2018) Thalamocortical dysrhythmia detected by machine learning. Nature Commun 9:1103
22. Zhang J, Xu H, Zhu L, Kong W, Ma Z (2019) Gender recognition in emotion perception using eeg features. In: 2019 IEEE international conference on bioinformatics and biomedicine (BIBM), pp 2883–2887

Part IV
Innovation in Medicine and Health

Chapter 14
Augmented Medicine: Changing Clinical Practice with Artificial Intelligence

Giovanni Briganti

Abstract Smart medical technologies are augmenting clinical practice by offering the patient an increased autonomy and the clinician more advanced tools to predict, detect, monitor, and treat diseases. Augmented Medicine, a new framework of techniques that extends to clinical practice from the applied medical research aiming to introduce and improve tools, is rapidly gaining popularity. In this chapter, we will outline the principles of Augmented Medicine and its main applications in clinical practice, as well as future directions of this promising field.

14.1 Introduction

Artificial Intelligence (AI) has revolutionized medical technologies and can be commonly understood as the part of computer science that is able to deal with complex problems with many applications in areas with huge amount of data but little theory. AI is defined as *perceiving*, *reasoning*, and *acting* computation: in medicine, the two most commonly popular domains of AI are *machine learning* and *deep learning*. Machine learning can be defined as a computation that can improve from experience and is divided into *supervised learning*, when the data input for prediction is labeled, and *unsupervised learning* when the data input is unlabeled [1]. The two techniques that stem from machine learning and are common in medicine are *random forests*, understood as decision trees, and *Artificial Neural Networks* (ANN), understood as network with hidden layers and *n* neurons per layer, each assigned with a specific role. Deep learning stems from machine learning but requires much lower computational resources as the number *n* of neurons per layer decreases. Deep learning is optimized for big and temporal data sets.

There is a manifold of differences between the AI framework and the more widely known traditional statistical techniques. Statistics is mostly based on theory and

G. Briganti (✉)
Université libre de Bruxelles, Bruxelles, Belgium
e-mail: giovanni.briganti@ulb.be

T. D. Pham et al. (eds.), *Advances in Artificial Intelligence, Computation, and Data Science*, Computational Biology 31,
https://doi.org/10.1007/978-3-030-69951-2_14

hypothesis testing, and is often employed with low-dimensional data sets as input (that is, the number of subjects exceeds by far the number of variables). On the other hand, AI is mostly based on data and predictability of variables, and is often employed with high-dimensional data sets (that is, the number of variables exceeds the number of subjects) and requires very big data sets. An example of high-dimensional data is input stemming from genome analyses, where thousands of genes stem from only one individual.

Medical technologies are defined as tools that can enable physicians to perform early diagnosis, reduce complications, optimize treatment and/or provide less invasive options, and reduce the length of hospitalization. Before the mobile era, medical technologies were mainly known as classic medical devices (e.g. prosthetics, stents, and implants), although in the last decade, the emergence of smartphones, wearables, sensors, and communication systems has revolutionized medicine with the capability of containing AI-powered tools (such as applications) in very small sizes [2]: these new tools can be customarily defined as smart medical technologies. Smart medical technologies enjoy increasing success and can be easily adopted by patients, since they increase their autonomy in various domains, such as keeping electronic personal records, monitoring vital functions with biosensors, and reaching optimal therapeutic compliance.

The development of intelligent medical technologies is enabling the rise of a new field in medicine: Augmented Medicine (AM), i.e. the use of new medical technologies to improve different aspects of clinical practice. The origins of AM can be traced back to the start of a series of approval by the Food and Drug Administration (FDA) of several AI-based technologies, although smart technologies that do not necessarily employ AI are also understood as a part of AM [3]: two examples of such technologies are surgical navigation systems for computer-assisted surgery [4] and virtuality-reality continuum tools for surgery, pain management, and psychiatric disorders [5–7].

Although patients seem to be interested in AM, a certain physician resistance arose against it: this can be traced back to four main reasons. First, most practising physicians around the world did not have appropriate education on AM [8], since it is very recent: even currently, medical students are not thinking about it, even though several AM courses around the world are proposed as alternative medical curricula. Second, AM is naturally related to the field of medical informatics, which in the past two decades has dealt with the early digitization process in health care, including the shift to electronic health records (EHR): the administrative burden that came with this shift [9] is nowadays recognized as an important cause of physician burnout [10]. Third, physicians are worried about losing the battle against AI when it comes to compared performance: this is secondary to the fact that most medical literature on AI is focused on comparing performances between the human and the machine: this has been one of the main pitfalls of AI literature to this day, with consequences on physicians' perception that are difficult to assess and overthrow. Fourth, the current worldwide lack of a legal framework [11] that defines the concept of liability in the case of adoption or rejection of algorithm recommendations leaves the physician exposed to potential legal outcomes when using AI. One of the only proposed concepts that can be applied in clinical practice is the French attitude of *garantie humaine* (human warranty), that is, there must always be a physician supervising the AM framework.

The important physician resistance to AM is one of the reasons why current medical literature on AI tools in clinical practice involves little to no medical expertise: such tools are often developed by private companies without important partnerships within the hospital framework, to the exception of widely known tech companies. This translates to poor reliability of the study designs that have emerged in a meta-analysis of deep learning tools in the field of medical imaging [12].

This chapter serves as an introduction to the practising physician or the medical technologist interested in what the main applications of AM are that can be easily integrated into clinical practice, and provides the main problems to overcome in its current state as well as potential solutions.

14.2 Implementation of Augmented Medicine in Clinical Practice: An Overview

AM has different domains of application: we will hereby overview some of the popular ones.

14.2.1 Monitoring with Wearable Technology

The domain of parameter monitoring is certainly the most widely used application belonging to AM. There has always been a knowledge gap of what happens to patients between contacts with a healthcare professional (for instance, you can go an entire year without consulting a general practitioner or a specialist for a given disease): that is why physicians hospitalize patients when they want to closely monitor their status. However, in selected cases, monitoring can now be outsourced with the appropriate medical technologies. Several critiques of wearable and portable ECG technologies have been addressed [13], highlighting limitations to their use, such as the false positive rate that originates from movement artifacts, and barriers in the adoption of wearable technology in elderly patients that are more likely to suffer from atrial fibrillation. On the other hand, several studies have reported the benefits of outsourced monitoring. To illustrate such benefits, let us review three use cases: the monitoring of atrial fibrillation, diabetes, and epilepsy.

Patients with an abnormal heart rhythm can suffer from complications where rapid medical intervention is crucial for survival. The early detection of atrial fibrillation (abnormal heart rhythm characterized by an irregular beating of the atria) was one of the first applications of AI in medicine. The recent REHEARSE-AF study [14] showed that remote ECG monitoring in ambulatory patients is more likely to identify atrial fibrillation than routine care. On a more general basis, wearable technology can allow for the remote monitoring of heart rhythm.

People living with diabetes (high glucose levels over a prolonged period of time) and treated with anti-diabetic drugs are affected with hypoglycemic episodes (that is, the glucose levels are too strongly corrected) presenting as loss of consciousness, confusion, or seizures. Continuous glucose monitoring (CGM) enables patients with diabetes to view real-time interstitial glucose readings and provides information on the direction and rate of change of blood glucose levels [15]: wearable technology focused on CGM is able to prevent hypoglycemic episodes based on a repeated measurement that can ideally be shared with clinicians. CGM can enable patients to optimize their blood glucose control and reduce the stigma associated with hypoglycemic episodes; however, a study focusing on patient experience with glucose monitoring reported that participants, while expressing confidence in the notifications, also declared feelings of personal failure to regulate glucose level [15]. However, thanks to CGM technology, patients can now be more aware while in social gatherings or other occasions, which is a clear benefit to the quality of life.

Children and adults with epilepsy can experience unexpected seizures, which can lead to a significant decrease in quality of life and can seriously complicate learning and leading a professional life as well as being included in social activities.

Intelligent seizure detection devices are promising technologies that have the potential to improve seizure management through permanent ambulatory monitoring. Wearables associated with electrodermal captors can detect generalized epilepsy seizures and report to a mobile application that is able to alert close relatives and trusted healthcare providers with complementary information about patient localization [16]. A report focused on patient experience revealed that, in contrast to heart monitoring wearables, patients suffering from epilepsy had no barriers in the adoption of seizure detection devices, and reported high interest in wearable usage [17].

14.2.2 AI for Diagnosis

Far from being the most widely known application of AM by patients, diagnostic AI applications are however widely known by medical professionals.

Diagnostic AI ranges in a variety of specialties in the medical field. Input data not only can be body parameters (such as heart rhythm, ECG, glucose levels, and blood pressure) but can also be pictures from medical imaging (such as radiographies, CT-scans, and MRI), a video input from an endoscopy procedure, or histopathology slides.

Recent medical technologies also allow for a smart and outsourced medical history taking, that is, the patient inputs the symptoms directly on a smartphone, while the systems keeps providing new questions related to the symptoms at hand to achieve a differential diagnosis or simply to order the history information for the clinician to export into the electronic medical record and analyze.

The workflow of two specialties of medicine are revolutionized with AI: histopathology and diagnostic radiology, since they are two specialties where most of the workflow is focused on diagnosis. For instance, deep learning algorithms have been proven to be at least as good as trained radiologists in a recent meta-analysis [12] that covered most papers published on the topic.

The domain of deep learning for diagnostics is young but counts already a solid number of publications in the literature (over 1 million publications found in Google Scholar): however, several issues exist in this domain that need to be addressed in the coming decade to greatly improve the clinical benefit of diagnostic AI; we will hereby describe three of them.

The first issue is the lack of clinical validation of the core concepts and tools of diagnostic AI: this is mainly due to a lack of primary replication, that is, once algorithms are trained and tested (on a respective training and testing set), they are seldom applied to other data sets from the same population, and even more rarely applied to samples from other populations [3]. A solution to this first issue is moving toward open data and science settings for the development of algorithms.

The second issue is the problem of overfitting: when applied to other data sets, algorithms do not perform as good as with training and testing data sets, because often the algorithm is built so as to closely fit the source data and therefore does not replicate. This is a potentially crucial barrier for the adoption of diagnostic AI in clinical practice: it would be a great problem if a given hospital adopted an algorithm and that poorly performs on the population because it overfits its source data. There are two possible ways forward for the issue of overfitting: to reevaluate and recalibrate the performance of algorithms after institutional adoption, and to develop from scratch algorithms that fit larger communities while taking into account subgroups.

The third issue is the study the performance of diagnostic AI in comparison with physicians' performance: such studies constitute the vast majority of the rationale for diagnostic AI works (so as to prove a software's worth), but are however not optimized for the settings in which such diagnostic AI will be deployed (that is, with a physician's supervision). The clear solution for this issue is publishing studies that analyze the combined force of AI and physicians. This solution not only solves the issue of performance but is also a good way for discovering how practising AM revolutionizes the workflow in selected specialties and therefore the way we treat patients.

14.2.3 Machine Learning for Prediction

Electronic medical records (EMR) are a precious source of information. Such information can now be transformed into actionable data with medical informatics languages such as SNOMED-CT, which associates each medical term with a specific code, if it is either directly input by the practitioner while filling the EMR or if there is an underlying Natural Language Processing software that analyzes the EMR.

This way, machine learning (ML) applications are able to predict a given event (such as a failure or a given complication) based on the EMR. This has been proven useful in the field of internal medicine. For instance, ML has been proven useful for the prediction of cardiovascular risk [18], the prediction of the decline of glomerular filtration rate in patients with polycystic kidney disease [19], and several gastrointestinal disorders [3].

The adoption of such ML prediction, should they become more performant and with high replicability, could be straightforward for hospitals: integrated with the EMR, they could constitute a *risk dashboard* that could at all times alert physicians of possible complications and constitute a great clinical decision support system personalized for any given patients.

14.3 Conclusions

This chapter aimed at describing the main applications of artificial intelligence in medicine which leads to augmented medicine. The implementation of artificial intelligence in clinical practice is a promising area of development that rapidly evolves together with the other modern fields such as precision medicine, genomics, and teleconsultation. While scientific progress should remain rigorous and transparent in developing new solutions to improve modern health care, health policies should now be focused on tackling the ethical and financial issues associated with this cornerstone of the evolution of medicine.

References

1. Yonghong Peng, Yufeng Zhang, Lipo Wang (2010) Artificial intelligence in biomedical engineering and informatics: an introduction and review. Artif Intell Med 48(2–3):71–73
2. Steinhubl SR, Muse ED, Topol EJ (2015) The emerging field of mobile health. Sci Trans Med 7(283):283rv3
3. Giovanni B, Olivier LM (2020) Artificial intelligence in medicine: today and tomorrow. Front Med 7:27
4. SamuelC Overley, SamuelK Cho, AnkitI Mehta, PaulM Arnold (2017) Navigation and robotics in spinal surgery: where are we now? Neurosurgery 80(3S):S86–S99
5. OrenM Tepper, HayeemL Rudy, Aaron Lefkowitz, KatieA Weimer, ShelbyM Marks, CarrieS Stern et al (2017) Mixed reality with hololens: where virtual reality meets augmented reality in the operating room. Plast Reconstr Surg 140(5):1066–1070
6. MatthewC Mishkind, AaronM Norr, AndreaC Katz, GregM Reger (2017) Review of virtual reality treatment in psychiatry: evidence versus current diffusion and use. Curr Psychiatry Rep 19(11):80
7. KevinM Malloy, LeonardS Milling (2010) The effectiveness of virtual reality distraction for pain reduction: a systematic review. Clin Psychol Rev 30(8):1011–1018
8. Martin H, Christoph I, Fischer MR, German Medical Education Society (GMA), Committee Digitization—Technology-Assisted Learning and Teaching, Joint working group Technology-enhanced Teaching and Learning in Medicine (TeLL) of the German Association for Medical

Informatics, Biometry and Epidemiology (gmds) and the German Informatics Society (GI) (2018) Digital teaching and digital medicine: a national initiative is needed. GMS J Med Educat 35(3):Doc43

9. Chaiyachati KH, Shea JA, Asch DA, Liu M, Bellini LM, Dine CJ, et al (2019) Assessment of inpatient time allocation among first-year internal medicine residents using time-motion observations. JAMA Internal Med 179(6):760–767
10. West CP, Dyrbye LN, Shanafelt TD (2018) Physician burnout: contributors, consequences and solutions. J Internal Med 283(6):516–529
11. Price WN, Gerke S, Cohen IG (2019) Potential liability for physicians using artificial intelligence. JAMA 322(18):1765–1766
12. Xiaoxuan Liu, Livia Faes, AdityaU Kale, Wagner SiegfriedK Fu, DunJack Bruynseels Alice et al (2019) A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis. Lancet Digit. Health 1(6):e271–e297
13. Raja JM, Elsakr C, Roman S, Cave B, Pour-Ghaz I, Nanda A, et al (2019) Apple watch, wearables, and heart rhythm: where do we stand? Ann Trans Med 7(17):417
14. JulianPJ Halcox, Kathie Wareham, Antonia Cardew, Mark Gilmore, JamesP Barry, Ceri Phillips et al (2017) Assessment of remote heart rhythm sampling using the alivecor heart monitor to screen for atrial fibrillation: the rehearse-af study. Circulation 136(19):1784–1794
15. Lawton J, Blackburn M, Allen J, Campbell F, Elleri D, Leelarathna L, et al (2018) Patients' and caregivers' experiences of using continuous glucose monitoring to support diabetes self-management: qualitative study. BMC Endocr. Disord. 18:12
16. Giulia Regalia, Francesco Onorati, Matteo Lai, Chiara Caborni, RosalindW Picard (2019) Multimodal wrist-worn devices for seizure detection and advancing research: Focus on the empatica wristbands. Epilepsy Res. 153:79–82
17. Bruno E, Simblett S, Lang A, Biondi A, Odoi C, Schulze-Bonhage A, et al (2018) Wearable technology in epilepsy: the views of patients, caregivers, and healthcare professionals. Epilepsy Behav 85:141–149
18. Dorado-Díaz PI, Sampedro-Gómez J, Vicente-Palacios V, Sánchez P (2019) Applications of artificial intelligence in cardiology. The future is already here. Revista Española de Cardiología (English Edition) 72(12):1065–1075
19. Olivier Niel, Charlotte Boussard, Paul Bastard (2018) Artificial intelligence can predict GFR decline during the course of ADPKD. Am J Kidney Dis: Off J Natl Kidney Found 71(6):911–912

Chapter 15
Environmental Assessment Based on Health Information Using Artificial Intelligence

Jannik Fleßner, Johannes Hurka, and Melina Frenken

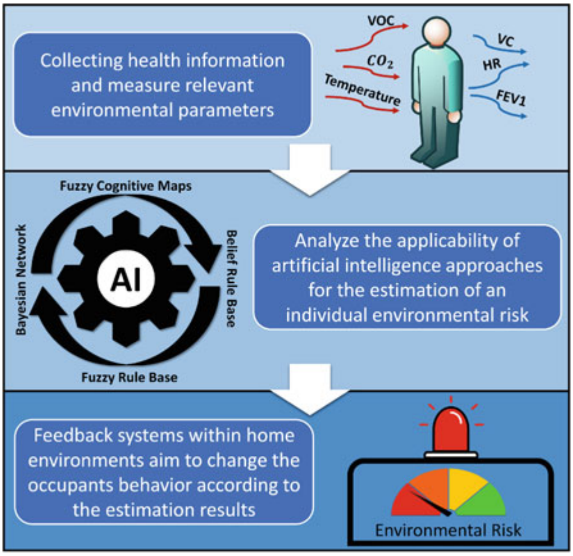
Abstract A holistic care system which enables extensive medical care even outside the hospital brings significant benefits for health care. The application of novel communication and computation technologies is essential in order to accomplish such a system. In the presented chapter, a conceptual system is described which links environmental parameters measured by building automation and control systems with data from electronic health records. The system's purpose is to provide medical personnel with interpreted data about possible adverse health effects of the indoor environment with respect to the patient's health condition. Additionally, the patient receives real-time feedback about the environmental parameters and their potential health effects. The purpose of this feedback is to inspire behavior changes in the patient, which results in a more health-friendly environment. A special focus of the chapter lies on the analysis of possibly applicable artificial intelligence approaches for the estimation of the individual environmental risk factor. These are necessary because the system combines knowledge about the adverse health effect of environmental parameters and knowledge about health parameters for the environmental assessment. This knowledge is often incomplete, ambiguous, and is linked to uncertainty, which makes the interpretation of the raw data non-trivial and would overstrain the occupant as well as the medical personnel.

15.1 Introduction

Digitization is already part of many applications and business processes today. Technological progress enables new approaches to support and improve activities in various areas of daily life. One of these areas is health care. In particular, one goal is to establish a holistic care system that enables comprehensive medical care outside of the hospital.

J. Fleßner (✉) · J. Hurka · M. Frenken
Jade University of Applied Sciences Wilhelmshaven/Oldenburg/Elsfleth, Ofenerstr. 16/19, Oldenburg, Germany
e-mail: jannik.flessner@jade-hs.de

T. D. Pham et al. (eds.), *Advances in Artificial Intelligence, Computation, and Data Science*, Computational Biology 31,
https://doi.org/10.1007/978-3-030-69951-2_15


Collecting health information and measure relevant environmental parameters
VOC
CO2
Temperature
VC
HR
FEV1
Fuzzy Cognitive Maps
Bayesian Network
AI
Belief Rule Base
Fuzzy Rule Base
Analyze the applicability of artificial intelligence approaches for the estimation of an individual environmental risk
Feedback systems within home environments aim to change the occupants behavior according to the estimation results
Environmental Risk

During the last years, several scientific papers have been focused on the development of telemedical applications in order to provide medical care within the home environment. A common approach is the use of building automation and control systems (BACS) technologies to build telemedical systems [41]. In general, the communication technologies developed for BACS are often used to connect various medical devices and sensors. A usual application is emergency detection within the home environment. The data is transmitted to healthcare personnel who are monitoring it in hospitals or service centers.

In this context, the centralized storage of patient information and to enable instant access for authorized users are the purposes of electronic health records (EHRs). Therefore, the development and integration of EHRs offer the possibility of collecting more comprehensive data on the course of various diseases of patients treated with different methods in different hospitals. Accordingly, the analysis of EHR data must meet high standards of data protection and data security, but has the potential to determine the most efficient treatment for certain diseases. Moreover, EHRs enable the combination of information obtained from hospital treatment with data recorded by telemedical applications in the home environment. However, today EHRs are not widely implemented within medical institutions and the development of EHR platforms like openEHR [48] still continues.

A quite unnoticed opportunity of the recent technological developments is the provision of information about environmental parameters within the home environment for healthcare applications. Parameters like air temperature, humidity, air pollution, and illumination significantly affect the well-being of the occupant and may also affect the progression of specific diseases. In this context, Fong and Fong [19] designed a wireless telemedical network to monitor indoor air pollution, which is associated with chronic respiratory diseases. Such efforts focused on the relation of indoor environmental parameters, and diseases are rather rare. Nevertheless, there is extensive scientific evidence that indoor as well as outdoor environmental parameters influence the mortality, morbidity, and prevalence of specific disease symptoms. Consequently, the medical treatment may benefit from the inclusion of measured indoor environmental parameters with relevance for the particular disease.

However, the additional data needs to be interpreted before transmission to the EHR in order to facilitate their handling. Therefore, approaches to estimate the environmental risk based on knowledge about the health effect of environmental parameters and knowledge about the patient's health data are required. In addition, it is of importance to provide the patient with adequate feedback about the environmental condition based on the estimation outcomes. An intuitive human-system interface improves the acceptability of the system and may lead to behavior change with the purpose of improving the current environmental condition.

In Sect. 15.2, an overview of the effects of environmental parameters on various disease symptoms is given. Afterwards, a system concept for the environmental risk estimation based on BACS and EHR data is presented in Sect. 15.3. In this context, a closer look at the requirements and possible approaches for the environmental risk estimation using artificial intelligence is given in Sect. 15.4.

15.2 Environmental Parameters and Health

In the past years, a lot of effort was made investigating the effects of indoor environmental parameters on occupants' health. As one outcome, the sick building syndrome was defined, which associates different symptoms like headache, tiredness, and breathlessness with environmental parameters.

Particularly, there is scientific evidence regarding the effects of environmental parameters on pulmonary and cardiovascular diseases. Cardiovascular and pulmonary diseases are among the most common causes of death and are therefore scientific topics of particular interest. Thus, an overview of studies investigating the effect of environmental parameters on the pulmonary and cardiovascular systems is given in the following. The purpose of the overview is to underline the available knowledge, which possibly can be used for estimating the environmental health risk.

The available scientific studies are subject to certain limitations due to the complexity of the parameter interactions. A higher temperature, for example, may enhance the occurrence of harmful air particles, while a certain relative air humidity may lower their occurrence at the same time. In order to measure all these possible interactions, an expensive and time-consuming study design is required, which is not possible at any time.

A brief excerpt of scientific findings starting with the effects of air pollutants is given in the following. The effects of weather-related parameters are described afterwards. In the end, the effect of illumination and some implications for health-related BACS applications are presented.

15.2.1 Air Pollution

Without any doubt, air pollution affects the health of the population negatively. WHO is aware of these adverse health effects and suggests standards for selected air pollutants [70, 71]. However, these standards might not be sufficient for occupants with higher sensitivity caused by particular diseases. In fact, air pollution is seen as a major cause of several diseases, especially concerning the pulmonary and cardiovascular systems.

Air pollution is caused by various gases, particulate matter (PM), and volatile organic compounds (VOC), which are emitted into the environment by different sources. In general, higher concentrations of these particles lower the air quality and lead to a higher risk of disease symptom occurrence.

Argacha et al. [4] give a comprehensive overview of studies investigating the effect of different air components with a negative effect on the cardiovascular system. A large part of the presented studies investigates the effect of PM on the cardiovascular system. PM is classified in particles with a diameter between 2.5 and 10 μm (PM_{10}) and particles with a diameter of 2.5 μm or less ($PM_{2.5}$). In addition, particles with a size of less than 0.1 μm are defined as ultrafine particles. PM_{10} and $PM_{2.5}$ are both linked with higher risk of cardiovascular mortality and occurrence of cardiovascular disease symptoms [52]. However, Franck et al. [20] concluded in their work that the negative effect of ultrafine particles is even higher in comparison to PM_{10} and $PM_{2.5}$.

Yode et al. [75] investigated the effect of short-term exposure to PM on the pulmonary system of healthy adolescents. They found a strong relation between the decrease of pulmonary function and the concentration of PM. Especially, $PM_{2.5}$ seems to have an adverse health effect concerning the pulmonary system. The adverse health effect of $PM_{2.5}$ was confirmed by Haberzettl et al. [24]. Their experiments showed that a short-term exposure leads to vascular insulin resistance and inflammation in mice caused among others by pulmonary oxidative stress.

In recent years, the association among diseases of the pulmonary system like asthma, chronic obstructive pulmonary disease (COPD), and respiratory infections was focused on by several scientific workgroups. Kurt et al. [38] give an overview about the health effect of common air pollutants on these diseases. Heinrich and Schikowski [25] concluded in their systematic review about the vulnerability of COPD patients toward air pollution that they are more vulnerable to gaseous pollutants and PM compared to healthy people. In this context, To et al. [65] concluded in their work that the exposure to higher concentrations of $PM_{2.5}$ and O_3 increases the risk to develop the asthma-chronic obstructive pulmonary disease overlap syndrome. However, such reviews addressing the vulnerability of patients with certain diseases are rare but of high importance for the establishment of a holistic care system. The adverse health effect of short-term exposure of COPD patients to air pollution was confirmed by Gao et al. [21]. Their results show that COPD patients are even more vulnerable to gaseous pollutants like NO_2 and SO_2 compared with PM.

These gaseous pollutants and their health effects were the object of several scientific investigations as well. Hong et al. [29] conducted an epidemiological study about the association of gaseous pollutants like SO_2, CO, NO_2, O_3, and PM_{10} with a reduced pulmonary function. Their results suggest that there exists an adverse health effect of gaseous pollutants on the pulmonary system, especially for SO_2, CO, and O_3. In addition, Smith et al. [62] focused their research on the effect of air pollution on the occurrence of pulmonary tuberculosis. They found an elevated occurrence of pulmonary tuberculosis when exposed to higher concentrations of CO and NO_2.

Besides PM and gaseous pollutants, VOC concentrations often are related to disease symptoms. A recent scientific effort is spent on the detection of cancer based on the VOC composition in exhaled breath [9]. However, it was shown that the concentrations of these air pollutants are often higher in the indoor air than outdoors. Therefore, it is of importance to know how VOC may affect the occupants' health, even though the exact biological reasons remain unclear to some extent. The study of Cakmak et al. [8] suggests that the inhalation of VOC within the indoor environment has an adverse health effect. Their investigation showed that an increase in VOC leads to a significant decrease in measures of lung function. Rumchev et al. [55, 56] confirm the adverse effect of elevated VOC concentrations on the pulmonary system. Accordingly, the works of Boenisch et al. [6] and Wang et al. [69] show that exposure to VOC leads to increased airway inflammation in mice. The effect of VOC exposure on the pulmonary function of elderly people was investigated by Yoon et al. [76]. Their results suggest that VOC exposure leads to oxidative stress and a reduced pulmonary function in elderly people. Furthermore, the relation between long-term exposure to ambient VOC and mortality was investigated by Villeneuve et al. [68]. The results suggest that VOC concentration is positively associated with cancer mortality.

15.2.2 Weather-Related Parameters

Besides air pollution, the air temperature is a highly regarded parameter which is associated with the occurrence of disease symptoms. Specifically, cardiovascular disease symptoms seem to be affected by the ambient air temperature.

The air temperature as well as the diurnal temperature range is associated with cardiovascular diseases [35]. Especially, low temperatures and elevated diurnal temperature ranges are linked with adverse health effects. The finding of a negative health effect of the short-term exposure to air temperatures below an optimum was confirmed by Dahlquist et al. [15]. Additionally, the works of Shiue et al. [59] and Chen et al. [12] suggest a U-shaped relationship between temperature and cardiovascular diseases. Accordingly, extremes in temperature lead to stress of the cardiovascular system causing the occurrence of disease symptoms. The relationship of the physiologically equivalent temperature, which is determined using the temperature, wind speed, humidity, sky cover, and solar radiation, among others, and specific disease symptoms was investigated by Shiue et al. [60, 61]. They found slightly varying temperature ranges with the highest adverse health effect on particular diseases. An epidemiological study by Hensel et al. [26] found an adverse effect of cold temperature, low humidity, and high airspeed on the occurrence of heart diseases.

15.2.3 Illumination

Illumination is affected by natural light and artificial light sources controlled by the occupant. An important aspect of exposure to light is its effect on the circadian rhythm. The natural light-dark cycle is a major trigger of the inner clock [53]. Furthermore, it was shown that even single body functions and organic systems have their own circadian rhythm. In this context, several studies show that a disturbance of the circadian rhythm is associated with adverse health effects. A comprehensive review about the negative effects of the disturbance of the circadian rhythm is given by Khaper et al. [34].

15.2.4 Implications for Health-Related BACS

As shown before, there is plenty of evidence about the potentially adverse health effects of environmental parameters. These can be regulated by BACS with the purpose to create health-friendly environments. Consequently, a BACS needs to be built which is able to measure various air pollutants like SO_2, CO, NO_2, O_3, VOC, $PM_{2.5}$, and PM_{10}. Moreover, environmental parameters like air temperature, humidity, and illumination should be observed by the BACS as well.

It was shown that specific health conditions lead to a higher sensitivity to certain environmental parameters. Consequently, a system which brings together the regulation of the environmental parameters and the medical information about the patient would be an important step toward a holistic care approach.

Such a system should provide information about possibly harmful conditions within the patient's home environment for the medical personnel. Based on these information, the medical personnel are able to suggest adequate behavior to mitigate potential risk factors.

Additionally, real-time feedback within the home environment can be used to inform the patient about probable health risks originating from the current conditions. The purpose of the feedback is to change the occupant's behavior in favor of a healthier indoor environment.

15.3 System Concept for Health based Environmental Assessment

The system concept and feedback systems as the main part of the system are described in the following section. The feedback systems are basically expert systems providing information about the home environment for the occupants and the medical personnel. These feedback systems retrieve data from BACS and EHR to determine feedback signals. Accordingly, the feedback systems combine knowledge about the environmental parameters to estimate possible environmental health effects and knowledge about health parameters to estimate the occupant's sensitivity. The combination of the possible environmental health effects and the occupant's sensitivity is used to estimate an individual environmental health risk. As stated before, interpreting the raw data on the background of compiled knowledge requires artificial intelligence approaches because it would overstrain the occupant as well as the medical personnel.

In the following, the general concept of the whole system (Sect. 15.3.1) and some specifications of the feedback signals for the medical personnel (Sect. 15.3.2) and the occupants (Sect. 15.3.3) are described. A short description of the challenging communication between BACS and EHR is given in Sect. 15.3.4. In Sect. 15.4, special attention is paid to possible artificial intelligence approaches for the implementation of the feedback systems.

15.3.1 System Components and Their Interactions

The general concept of the system and the relations between system components and persons are depicted in Fig. 15.1. The medical personnel perform medical checkups with the patient in order to gain information about the patient's health condition and document the results in the EHR. Especially, medical scores are a relevant information format because these scores already contain medical knowledge about the severity of a particular health condition [47]. Nevertheless, also measured parameters like heart rates and vital capacity are useful data to model the patient's sensitivity.

These data are transmitted to the feedback systems, which receive information about the measured environmental parameters from the BACS as well. The BACS observes and potentially regulates the environmental parameters which the occupant is exposed to. As summarized before, exposure to these environmental parameters may affect the occupant's health positively or negatively. The occupant perceives the environmental condition and compares the perception with her or his preferences. Based on the result of this inner process, the occupant interacts with the BACS and changes its configuration or tries to manipulate the environmental condition manually.

One purpose of the feedback systems is to present the environmental condition within the home environment with respect to the health condition of the occupant.

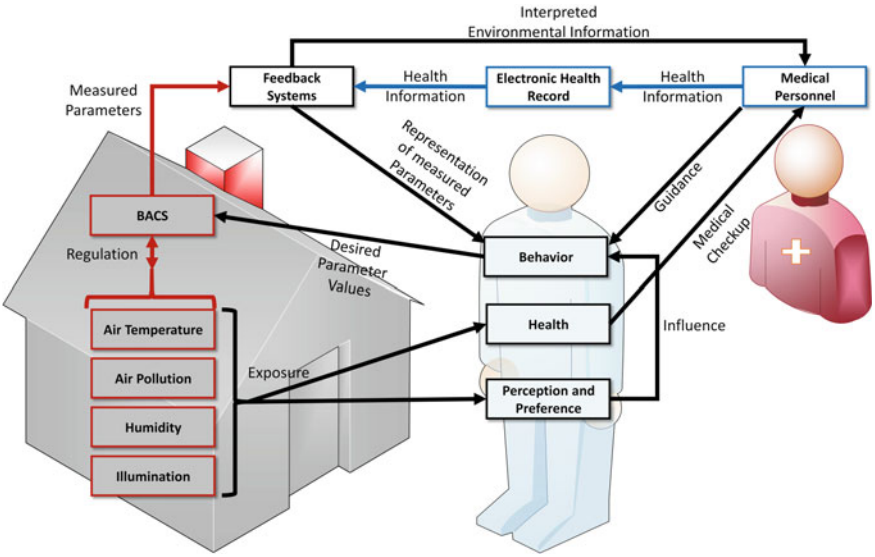


Fig. 15.1 An abstract overview of the general concept and the interactions between the involved persons and systems. The red marked objects on the left side are the environment-related system components. The blue marked components at the top are related to medical information. The black marked symbols in the middle describe the interactions between the occupant and other persons and systems

This feedback aims to influence the occupant's behavior positively without taking their self-determination.

Another purpose is to inform the medical personnel about the possibly adverse health effects of the environment in which the patient is living. The medical personnel are able to define suggestions based on these information and may draw conclusions concerning the development of the patient's health.

A closer look at the data procession steps within the feedback systems is given in Fig. 15.2. The assessment of the environment for a chosen period of time will be done based on the data retrieved from the BACS, the EHR, and a knowledge base about the health effects of environmental parameters. The information reduction process filters the needless information with respect to the health condition of the particular patient. With the filtered information, the desired feedback format such as scores or classes will be determined and transmitted to the medical personnel via the EHR or visualized for the occupant via an interface. Both types of feedback are described in more detail in the following.

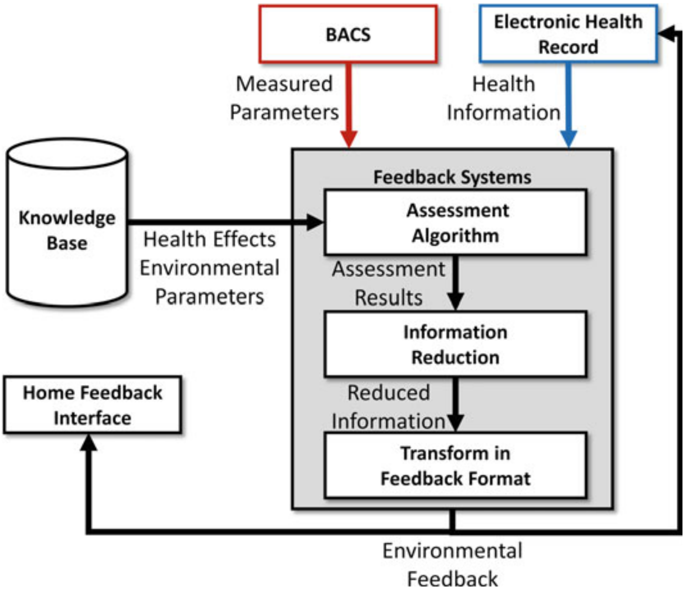


Fig. 15.2 The process to determine feedback based on measured environmental parameters and health information about the occupant. An algorithm assesses the situation based on measured environmental parameters, health information about the occupant, and information retrieved from a knowledge base. Afterwards, only the important information is used to generate a feedback signal

15.3.2 Data Interpretation for Medical Staff

An important aspect of feedback systems linking medical information and measured environmental parameters is to provide the medical personnel with information about the patient's indoor environment. As an example, a patient suffering from COPD might complain about a more frequent occurrence of disease symptoms. In consequence, the medical personnel check the relevant physiological parameters. If there are no physiological anomalies, the medical personnel might have a closer look at the provided environmental information and give advice to reduce the concentrations of PM and gaseous air pollutants within the home environment. However, in order to be able to quickly decide whether the environmental condition might be a cause of the frequent occurrence of disease symptoms, a data interpretation is necessary to support the decision-making.

Several expert systems supporting the medical personnel during the search for the correct diagnosis or treatment of different diseases were developed in recent years [1]. A lot of these systems use algorithms which belong to the artificial intelligence research like fuzzy logic or artificial neural networks.

A similar approach seems to be suitable for an expert system interpreting the environmental condition considering the health status of the occupant. The complexity resulting from different effects of different environmental parameters at the same time leads to a more complex set of rules, which have to be defined in order to take into account all possible combinations. Therefore, the application of artificial intelligence approaches will be necessary. Moreover, the available data about the health effects of environmental parameters is quite heterogeneous and sometimes ambiguous. The ambiguity is partly caused by differences in study design or by varying analysis methods. Consequently, one challenge for the development of an expert system will be to determine reliable feedback information based on the data. The system needs to be able to determine appropriate feedback even though the amount of data is quite limited. Today, there are not enough data available to rely on a big database. However, since feedback about the possible health effects of environmental conditions is an additional information, which might be useful but not crucial for medical personnel, accuracy requirements might be lower compared to other medical applications like finding the right diagnosis or suggesting the optimal treatment.

Another aspect which needs to be addressed during the development of the feedback systems is the reduction of information in an adequate manner. Not every information recorded might be important for the medical personnel. Consequently, decisions have to be made which information is transmitted according to the particular situations. A possible solution is a representation using nested data: a superior information reflects the probable overall effect of the environmental condition on the patients' health, whereas the data about the environmental condition is presented in more detail on the secondary level. These more detailed data contain the statistically analyzed concentrations of the different environmental parameters and time series of the concentrations. Using such a nested representation enables the medical personnel to decide on the basis of the superior information whether a closer look into the data is needed. Another possibility commonly applied for medical checkups is to determine a score. In order to find the most useful way of presenting the results, more investigation is needed.

15.3.3 Feedback Systems for Patients

The feedback systems for patients are in general quite similar to the ones for medical personnel. However, the biggest difference is the need for a specific interface to represent the probable harmfulness of the current environmental condition.

Moreover, the feedback for patients should reflect only the current environmental condition and its short-term changes. Consequently, there is even more reduction of information to keep it as simple as possible.

The feedback needs to be simple and requires to be interpreted intuitively. The purpose of the device is to influence the patient's behavior without drawing too much attention to the interface. It needs to be integrated unobtrusively into the patient's indoor environment. A device which is not perceived as a medical device increases the acceptability and avoids the stigmatization of the occupant. A possible solution is the development of an application for mobile devices, which informs the patient and sends a notification if the environmental condition worsens. However, this solution requires that the mobile device be in reach for the patient at any time. In addition, an activity by the user is required to receive the desired information. Another possibility is the development of a decorative device, which represents the feedback via a color-coded signal [18].

15.3.4 Communication Between BACS and EHR

A challenging aspect of the system development is the establishment of the communication between BACS and EHRs. Fortunately, the transfer between different medical institutions and the storage of healthcare data were topics of recent scientific efforts. Therefore, communication standards and frameworks like HL7 FHIR [28] and openEHR [48] were developed to provide opportunities to share, organize, and store relevant data. Based on their developer-friendly architecture, these tools might be appropriate for the described purpose.

Nevertheless, a connection between the BACS network and the openEHR or FHIR server needs to be achieved. Therefore, a gateway is required, which receives the measured environmental parameters and transmits them to the openEHR or FHIR server. In order to minimize the amount of system components, the gateway should also be enabled to execute data procession steps.

15.4 Approaches for Environmental Assessment

In this chapter, possible approaches of artificial intelligence for the environmental risk assessment are described in detail. However, the organization of data retrieved from the previously described studies has to be done before the approaches are implemented. Therefore, the organization of data and derived constraints for the feedback system implementation are described firstly. Afterwards, the approaches for the design of expert systems, also known as knowledge-based systems, for the assessment of the environment with respect to the occupant's health condition are discussed.

15.4.1 Data Organization

The data published in epidemiological studies differ in form and scope. Therefore, it is necessary to organize the data adequately in order to reveal incompleteness, ambiguity, and uncertainty. The following structure for database entries is suggested in order to organize the data appropriately (Table 15.1). Using such a structure facilitates the search for specific information and gives a clear overview of the collected information.

Table 15.1 Suggested structure for the organization of collected information. At first, the particular environmental parameter is defined for every entry. Afterwards, the form of the results and the results itself are listed. These information are followed by the health status of the subjects. At last, the possibility to add some comments is included

Environmental parameter	Result type	Results	Health status	Comments
Temperature	Optimal range	18 °C–22 °C	Healthy	Short-term exposure
PM10	Percentage value	12 % per $10\,\mu g/m^3$	Cardiovascular symptoms	None
VOC	Odds ratio	1.45	COPD patients	Long-term exposure

The study results contain statements for optimal conditions (e.g. *a risk minimum at a temperature between* 18 °C *and* 22 °C), percentage values for risk increase or decrease caused by changes of conditions (e.g. *a* $10\,\mu g/m^3$ *increase in PM10 increases health events by* 12 %), and odds ratios (e.g. *the odds ratio for COPD symptoms and VOC exposure is* 1.45). All these information are useful for designing an expert system. However, a direct comparison of these different types of study results is obviously impossible. Therefore, methods for the design of the expert system require to be able to handle different forms of data or need to be able to generalize the different forms.

However, the available database leads to some constraints for the implementation of the feedback systems, which are described in the following.

15.4.2 Derived Constraints

The described data are quite different from data usually used for the development of expert systems and medical decision support systems. In general, data contain a command variable like correct diagnoses, which can be compared to predictions of the model. The outcome of the comparison is used to make a statement about the accuracy and analyze the need for model adaptions. However, the purpose of the models in our case is the estimation of the individual environmental risk based on the environmental condition and the occupant's health information. Such an individual environmental risk factor is not content of the data, which complicates issuing a statement about the accuracy of the model. This leads to the assumption that supervised machine learning approaches, which adapt the model based on the difference between command variable and model prediction, are not suitable without further adjustments.

Another limiting factor is the small amount of available data. In consequence, methods need to be chosen which enable the environmental assessment based on a smaller batch of data. According to this, deep learning methods seem to be inapplicable at first glance. However, the creation of synthetic data based on the characteristics of the available data might be an approach capable of enabling deep learning algorithms. Nevertheless, such deep learning approaches are often considered as *black box* solutions, which are hardly comprehensible for the user and should therefore preferably not be applied for the design of decision support systems or feedback systems.

In the next section, applicable methods which meet the requirements will be discussed in further detail. Moreover, the positive and negative aspects of the different methods will be described.

15.4.3 Appropriate Methods for Environmental Risk Estimation

In general, expert systems like the described feedback systems contain a knowledge base usually in the form of *if… then* rules, which are derived from expert knowledge. However, in cases involving a large amount of components and relationships, the definition of a complete rule set is quite complex. Therefore, a visualization of all components and relations represented by a finite state machine might facilitate the knowledge base creation [17]. In addition, visualization using finite state machines provides an overview of the knowledge base, alleviating the search for possible mistakes and the system adaption.

The other component of an expert system is the inference machine, which defines the system behavior when the *if-condition* of a particular rule is fulfilled. The inference machine defines how the activated rules are handled and is dependent on the desired outcome of the system. In our case, the outcome might be a score or a color-coded signal representing the probable health risk. Accordingly, the inference machine needs to determine one of these outputs.

However, the determination of simple *if… then* rules will not be appropriate to represent the ambiguity and uncertainty of the available information. Accordingly, the applicable methods are limited to those able to reflect the data characteristics. This leads to the assumption that an approach using methods of the artificial intelligence range might be the solution. Consequently, approaches using a belief rule base, fuzzy rule base, fuzzy cognitive maps, and Bayesian networks are presented in the following.

15.4.3.1 Belief Rule Base Systems

The extension of simple *if… then* rules with so-called *belief* factors is a possibility often used to represent additional information within an expert system. These belief rule bases enable the inclusion of knowledge about uncertainty. In general, a belief rule base can be designed using qualitative and/or quantitative data. Consequently, the approach can be applied for many different applications. It was shown that a belief rule base can be designed for several different applications like classification [11], disease diagnosis [30–32, 45, 50, 77], and risk assessment [36]. Moreover, the approach of belief rule bases was used for activity recognition with BACS sensors as well [10].

An often applied approach is the generic rule-base inference methodology using evidential reasoning (RIMER). A belief rule base system following the RIMER approach contains k rules of the form [73]:

$$\begin{aligned} R_k : if\, A_1^k \wedge A_2^k \wedge \cdots \wedge A_{T_k}^k, \\ then\{(D_1, \overline{\beta}_{1k}), \ldots, (D_N, \overline{\beta}_{Nk})\}, \\ \left(\sum_{i=1}^{N} \overline{\beta}_{ik} \leq 1\right) \end{aligned} \tag{15.1}$$

where A_i^k represents the referential values of antecedent attributes and D_n represents the consequent with the particular degree of belief $\beta_{n,k}$. As an example, a derived rule might be

$$\begin{aligned} R_k : if\ & \textbf{temperature}\ is\ \textit{hot} \wedge \\ & \textbf{VOC}\ is\ \textit{slightly high} \wedge \\ & \textbf{heart rate}\ is\ \textit{high}, \\ then\ & \textbf{health risk}\ is\ \{(\textit{harmless}, 0), (\textit{slightly harmful}, 0), \\ & (\textit{harmful}, 0.2), (\textit{harmful}, 0.8)\} \quad . \end{aligned} \tag{15.2}$$

The addition of weighting factors δ_{ki} for the antecedent attributes and weighting factors θ_k for particular rules enables more possibilities to represent knowledge and give certain attributes more importance than others. Therefore, different grades of sensitivity caused by particular health conditions can be modeled.

Furthermore, the RIMER approach can be extended using the belief structure for the possible antecedent terms as well [43]. Accordingly, the above example might be extended like this:

$$\begin{aligned} R_k : if\ & \textbf{temperature}\ is\ \{(\textit{cold}, 0), (\textit{medium}, 0.3)(\textit{hot}, 0.7)\} \wedge \\ & \textbf{VOC}\ is\ \{(\textit{low}, 0), (\textit{slightly low}, 0), (\textit{medium}, 0), \\ & (\textit{slightly high}, 0.6), (\textit{high}, 0.4)\} \wedge \\ & \textbf{heart rate}\ is\ \{(\textit{low}, 0), (\textit{medium}, 0), (\textit{high}, 1)\} \\ then\ & \textbf{health risk}\ is\ \{(\textit{harmless}, 0), (\textit{slightly harmful}, 0), \\ & (\textit{harmful}, 0.2), (\textit{harmful}, 0.8)\} \quad . \end{aligned} \tag{15.3}$$

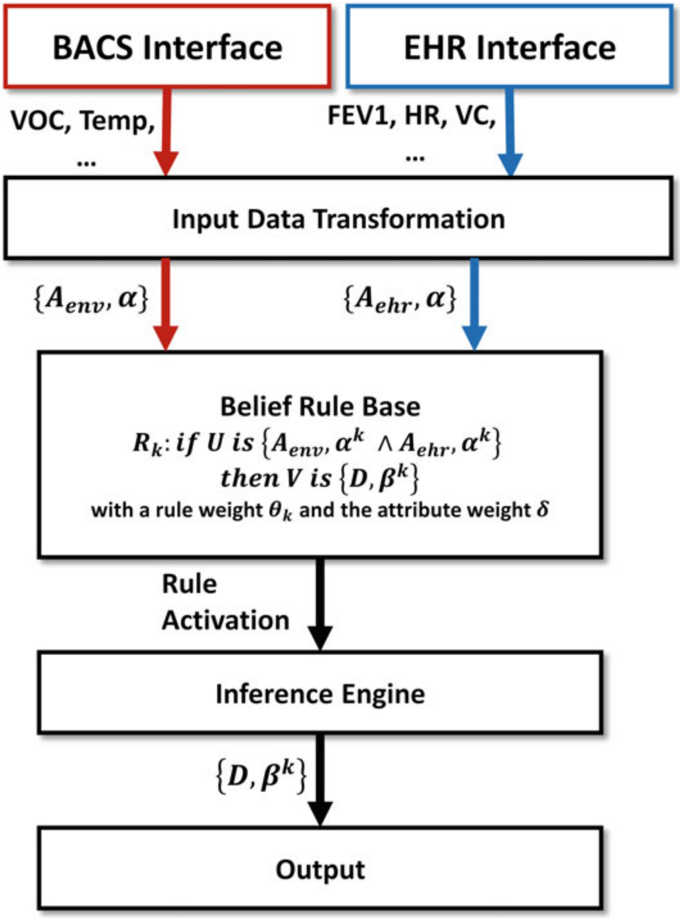

BACS Interface
EHR Interface
VOC, Temp, …
FEV1, HR, VC, …
Input Data Transformation
$\{A_{env}, \alpha\}$
$\{A_{ehr}, \alpha\}$
Belief Rule Base
R_k: if U is $\{A_{env}, \alpha^k \wedge A_{ehr}, \alpha^k\}$
then V is $\{D, \beta^k\}$
with a rule weight θ_k and the attribute weight δ
Rule Activation
Inference Engine
$\{D, \beta^k\}$
Output

Fig. 15.3 General approach for the environmental assessment using a belief rule base. The process starts with the input data transformation in order to relate the input value with the referential values of the antecedent attributes and a particular belief factor. These transformed input data lead to different activation values for each rule in the belief rule base. The rule activation values are used within the inference engine to calculate the resulting belief values for each referential value of the consequent

The process from the measurement of environmental parameters and the request of health information from the EHR to the determination of an output value using the belief rule base approach is depicted in Fig. 15.3.

The transformation of measured environmental parameters and the health information into referential values of the antecedents with assigned degrees of belief is done in the first step. Information and thresholds from relevant international standards and from WHO might be used as a starting point for the determination of belief degrees for the environmental parameters. Information about the effect of environmental parameters on human health from the knowledge base are useful to adapt the belief degrees. Moreover, the information from EHRs are transformed into the referential values of the antecedents and assigned degrees of belief as well.

Afterwards, the transformed information is processed by the belief rule base. The belief rule base consists of rules similar to (15.3) and represents the complex relations between the different parameters. The uncertainty and ambiguity of the data are represented by the degrees of belief. Each rule produces an activation value based on the input data, the rule weight, and the attribute weights. These rule activation values are processed by an inference engine, which, e.g. uses the evidential reasoning method to determine the degrees of belief for each referential value of the consequent.

In summary, the different approaches for the design of belief rule bases provide various opportunities to represent the characteristics of available data. For example, uncertainty of whether a certain air pollutant concentration should be classified rather as medium or high can be represented by particular belief factors of the antecedent attributes. Similarly, uncertainty about the probable health risk caused by a specific condition of environmental parameters can be represented by the belief degrees of the consequent.

15.4.3.2 Fuzzy Rule Base Systems

Another opportunity to model expert systems is to make use of the fuzzy logic concept. The usage of fuzzy logic is beneficial in cases when the assignment of an input parameter to an output parameter is not possible by implication. The basic concept of fuzzy logic is the linkage of input values to linguistic variables. The linguistic variables represent every attribute the input parameter might be assigned to. Therefore, a membership value within the interval $[0 - 1]$ for each linguistic variable is assigned to the input value. For example, an input temperature of 17 °C might be assigned to the attributes *cold*, *neutral*, or *hot* with the membership values $(0.3, 0.7, 0.0)$. Input membership functions are defined for the membership assignment. These functions are usually triangular or trapezoidal functions but may have every imaginable form. The transformation of input values into membership values for each linguistic variable is called fuzzification.

The results of fuzzification are afterwards processed within a rule base. The rule base represents knowledge about the application and defines relations between input linguistic variables and output linguistic variables. Each rule has a different amount of effect on the outcome based on the membership values. According to the example, a rule containing the linguistic variable *neutral* has the highest effect on the outcome.

The last part of the process is called defuzzification. Output membership functions for each output linguistic variable are used to determine a crisp output value. Usually, the output membership functions are truncated based on the particular membership values. Based on the example, the y-coordinates of an output membership function which is related to the *neutral* input membership value does not exceed the value 0.7. This is done for each output membership function. Afterwards, there are different approaches to calculate the output value with the modified output membership functions. One possibility is to use the center of gravity of the merged output membership functions. However, the usage of the position of maxima or minima of the merged output membership functions is a possible solution as well.

Fuzzy logic is the basis of several different medical decision support systems. An overview of the application of fuzzy logic for the development of medical support systems is given by Gorgulu and Akilli [23]. Samuel et al. [57] used a web-based approach to develop a medical decision support system with fuzzy logic. The purpose of their system is to support the diagnosis of typhoid fever and the estimation of the severity of the illness. An example of how fuzzy logic can be applied to implement an early warning system for the identification of patients in need of urgent medical care is given in Al-Dmour et al. [2]. In their work, mobile devices are used to observe the systolic blood pressure, heart rate, oxygen saturation, body temperature, and blood sugar. A fuzzy rule base was developed to determine one of 15 possible risk groups based on the input parameter. The comparison with the usually used Modified Early Warning Scoring (MEWS) shows that the fuzzy rule base system leads to quite similar results.

These examples show that the usage of fuzzy logic for the development of medical expert systems is a promising approach. Accordingly, a fuzzy rule base system for the assessment of the environmental condition with respect to the health information of the occupant might be an appropriate approach. The general process for the environmental risk and occupant's sensitivity assessment is depicted in Fig. 15.4.

The approach contains two more or less independent fuzzy rule bases for the BACS parameters and the EHR parameters. Adequate linguistic variables and input membership functions need to be created based on the available knowledge in order to transform the input values into degrees of membership.

Two sets of fuzzy rules are generated based on the available knowledge as well. Hence, rule activation values for the environmental risk caused by environmental parameters and for the occupant's sensitivity are calculated independently from each other. This separation leads to shorter rules and facilitates a clear arrangement. Moreover, adjustments concerning only the BACS or EHR part can be done without consequences for the other part.

The defuzzification process leads to crisp values for the general environmental risk and the occupant's sensitivity. The question about which method is the most useful for the defuzzification in this case needs to be investigated further. However, results for the environmental risk and the occupant's sensitivity might be combined afterwards to yield an estimation of the environmental risk for the occupant's specific health condition.

In summary, fuzzy rule bases provide several opportunities to model uncertain and ambiguous data. The membership functions enable the modeling of uncertain knowledge about the input parameters, the probable environmental risk, and the occupant's sensitivity. Furthermore, the definition of rules and the choice of the defuzzification method offer additional possibilities to model knowledge and adapt the system behavior.

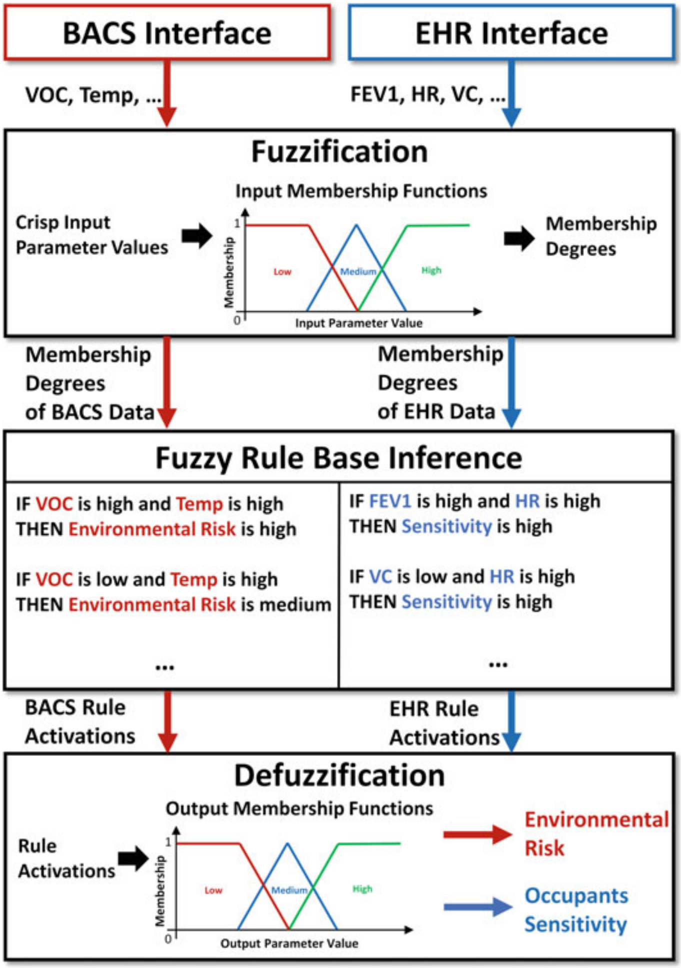

BACS Interface
EHR Interface
VOC, Temp, ...
FEV1, HR, VC, ...
Fuzzification
Input Membership Functions
Crisp Input Parameter Values
Membership Degrees
1
0
Membership
Low
Medium
High
Input Parameter Value
Membership Degrees of BACS Data
Membership Degrees of EHR Data
Fuzzy Rule Base Inference
IF VOC is high and Temp is high THEN Environmental Risk is high
IF VOC is low and Temp is high THEN Environmental Risk is medium
...
IF FEV1 is high and HR is high THEN Sensitivity is high
IF VC is low and HR is high THEN Sensitivity is high
...
BACS Rule Activations
EHR Rule Activations
Defuzzification
Output Membership Functions
Rule Activations
1
0
Membership
Low
Medium
High
Output Parameter Value
Environmental Risk
Occupants Sensitivity

Fig. 15.4 Suggested process for environmental risk estimation based on a fuzzy rule base system. The approach starts with the fuzzification of the input parameters retrieved from the BACS and EHR interfaces. The input membership functions used for the fuzzification are the first part representing the knowledge about the health effect of environmental parameters. Afterwards, the resulting membership degrees lead to the activation of rules from the fuzzy rule base. The fuzzy rule base is the second part of the knowledge representation. Crisp values for the environmental health risk and the occupant's sensitivity are determined during the defuzzification. The choice of the defuzzification method and output membership function is the last part of the knowledge representation

15.4.3.3 Fuzzy Cognitive Maps

Another approach for the modeling of expert systems is the application of fuzzy cognitive maps (FCM). In general, FCM are graph-based models, which represent the relationships between the different system components [37]. These models consist of nodes, so called concepts, for each system component and weighted edges for each causal relationship between the concepts. The concepts are associated with values from the interval [0, 1] which represent the current activation or relevance. Consequently, the concepts and edges are used to represent the knowledge about the effect of a particular system component toward other components, and the strength of the effect is defined via the weighted edges. Thus, the fuzzy logic approach of using linguistic variables to represent uncertainty for the definition of the weights is suggested. Accordingly, three general steps are needed to develop a FCM [49]. First, identify all relevant concepts and assign a state value A_i for each concept based on the knowledge about the application. Second, identify the relationships among the concepts. Third, assign weights w_{ij} to the resulting edges in order to estimate the strength of the relationships.

All states and weights can be represented by a state vector A and a weight matrix W, respectively. New state values A^k of the concepts are incrementally calculated based on the initial state values A^{k-1} of each concept and the weights w_{ij} of the edges. The states of the concepts are determined by the following formula:

$$A^k = f(A^{k-1} + \sum A^{k-1} \cdot W). \tag{15.4}$$

The function f is used to transform the results of the calculation into the interval $[0 - 1]$. One possibility is to use the sigmoid threshold function:

$$f = \frac{1}{1 + e^{-\lambda A_i}}. \tag{15.5}$$

The usability of FCM for the development of medical decision support systems is proved by several scientific works. Stylios and Georgopoulos [64] successfully applied the FCM approach to design a model to distinguish between the diagnosis of Specific Language Impairment and Dyslexia based on multiple different symptoms. Moreover, Anninou et al. [3] modeled a FCM for the determination of specific knee injuries based on different symptoms. Papageorgiou [49] shows the possibility to combine FCM and data mining approaches for the data-driven adaption of the edge weights.

An example of how FCM can be used for the environmental assessment with respect to the occupant's sensitivity is depicted in Fig. 15.5. The presented FCM contains two kinds of concepts: EHR-based concepts and BACS-based concepts. The EHR-based ones represent the patient's sensitivity based on the data retrieved from the EHR. The other ones stand for the observed parameters by the BACS. The graph-based representation gives a clear overview of the relations between the particular concepts. Increased sensitivities of the patient toward specific environmental parameters can be modeled via edges between EHR concepts and BACS concepts. As an example, an increase in heart rate might possibly affect the sensitivity of the patient toward air temperature and a decreased vital capacity might be related to a higher sensitivity toward the VOC concentration and the air temperature as well. Different intensities of the effects can be modeled by modifying the edge weights. Accordingly, a decreased vital capacity might have a more intense effect on the sensitivity toward the VOC concentration compared to the air temperature, which can be modeled by different weight values.

In general, the information from the EHR and the BACS are used to define initial values for the concepts. The system behavior is highly dependent on the initial values of the edge weights. Consequently, an approach has to be applied which transforms the information from the knowledge base into adequate weight values.

However, several questions need to be answered before the start of the implementation:

- What is the most appropriate method to determine the initial state values of the concepts?
- Which methods can be applied to determine the edge weights?

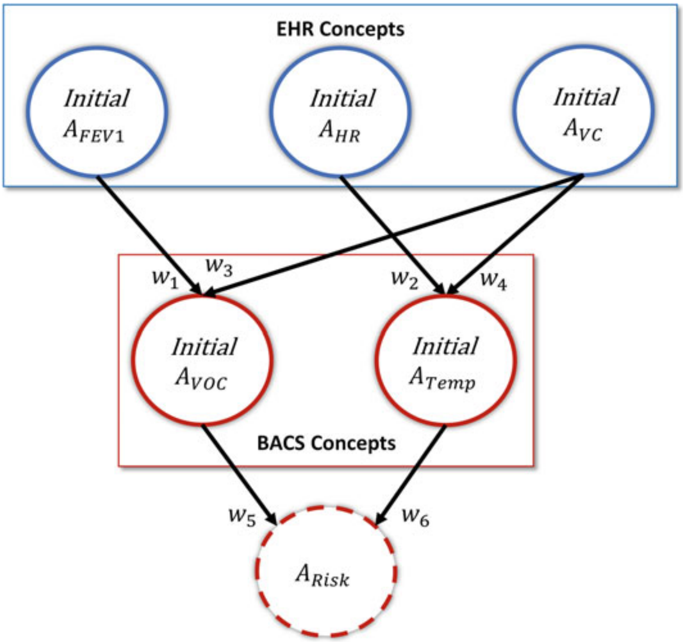


Fig. 15.5 Example of a FCM for health risk assessment based on data from BACS and EHR. The data is used to determine the initial values for particular concepts. In this example, initial values for the forced expiratory pressure in 1 s (A_{FEV1}), the heart rate (A_{HR}), and the vital capacity (A_{VC}) are used as parameters retrieved from the EHR. The VOC concentration (A_{VOC}) and the temperature (A_{Temp}) represent BACS parameters

- How many calculation increments are needed to get an adequate assessment result?

The estimation of initial state values for the concepts considering the environmental parameters is based on the actual values of the observed parameters. The comparison results of the measured values with data retrieved from scientific studies and suggestions within relevant standards could be supportive for a reasonable estimation. In consequence, these initial states are independent of patients' health status.

The initial state values of the EHR-related concepts are estimated based on the interpretation of the observations made by medical checkups. Therefore, available knowledge retrieved from scientific studies and interviewed medical experts might be needed to find reasonable values for the initial state values.

Similar to this, the estimation of edge weights might profit from additional expert knowledge. Nevertheless, data about the effect of environmental parameters toward health can be used as a basis for the definition of the edge weights. An approach to do so is to calculate the ratios between comparable data. As an example, the results of different studies investigating the effect of VOC concentration and air temperature toward the vital capacity might be used to calculate a ratio. This ratio is afterwards used to define the edge weights. However, studies often are designed differently in different aspects, so additional expert knowledge might be needed to refine the edge weights.

The computations using (15.4) and (15.5) lead to changes of the BACS-based concept state values based on the initial state values and edge weights estimated before. Therefore, the patient's sensitivity which is represented by the initial state of the EHR-based concepts has a major influence on the system outcomes.

A threshold for the difference between the previous state values and the actual state values can be used to limit the number of calculation cycles. Consequently, if the calculated state value for the health risk does not change significantly, the calculation process might be stopped. Another option is to set a maximum calculation time in order to provide nearly real-time feedback.

15.4.3.4 Bayesian Networks

In a Bayesian network [51], a set of variables x_i described by a joint probability distribution $P(x_1, x_{,2}, \ldots, x_n)$ is organized in a directed acyclic graph where each node stands for a variable x_i and the links from parent nodes x_p to x_i represent the parent's influence on the conditional probability distribution of the child $P(x_i | x_{p_1}, x_{p_2}, \ldots)$ or $P(x_i | S_i)$ where S_i denotes the set of direct parents of x_i. Nodes not directly connected are understood as not influencing each other (Markov condition). Thus, the joint probability distribution can be factorized over a set of lower-dimensional conditional probabilities:

$$P(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | S_i)$$

which are generally easier to compute.

Nodes without parents can be considered as input nodes, being frequently updated by measurements, or containing fixed probability distributions. Changes in the input nodes will then be propagated through their descendants, sequentially updating the network.

A thorough introduction into Bayesian networks and their use in medical context are given in [44]. Regarding air quality, work has been published ranging from global assessment [58] to mobile indoor sensing systems [33] and utilizing smart environments [16], and connecting air pollution with health risks [22]. Several examples use data from electronic patient records: [5, 27, 54].

Bayesian networks can be constructed automatically from data. Search and scoring algorithms [7, 14, 39, 40, 66, 72] are suitable for structure learning, which determines the optimal layout of the network, and parameter learning, optimizing the actual conditional probability distributions. These algorithms evaluate different structures with the goal to optimize a score which determines the feasibility of the network for the given data.

For structure learning, constraint-based algorithms [13, 46, 63, 67, 74] can be used as well. These apply conditional independence tests on any two nodes and determine where edges are required.

Networks can also be constructed manually, representing established or assumed causal relationships. Thus, it is possible to apply the logic of Bayesian networks in situations where there is no large base of consistent and reliable training data. A hypothetical network for the risk estimation based on environmental parameters and health information is given in Fig. 15.6. An example of how such a Bayesian network can be constructed on the basis of a limited dataset containing mortality and morbidity rates is presented by Liu et al. [42]. However, in our case, their approach needs to be adapted to include and estimate the patient's sensitivity.

Generally, the input parameters of the BACS and the EHR are transformed into discrete values. Therefore, the value range of each parameter is classified into a desired number of classes. Afterwards each class of each parameter is associated with a probability value representing the possibility of the occurrence of disease symptoms which is derived from the available epidemiological data. In case of missing required data, Experts should be involved to estimate adequate values. This is done to determine probabilities of the patient's sensitivity and general environmental risk. The specific health risk for the patient is calculated with the sensitivity probability and the environmental risk probability in the final step.

15.5 Conclusion and Discussion

In this work a system concept combining measurable environmental parameters and health information for environmental assessment was described. Firstly, the system components and the interactions between the system and its user were described. Afterwards, the specifications and requirements of the feedback systems were dis-

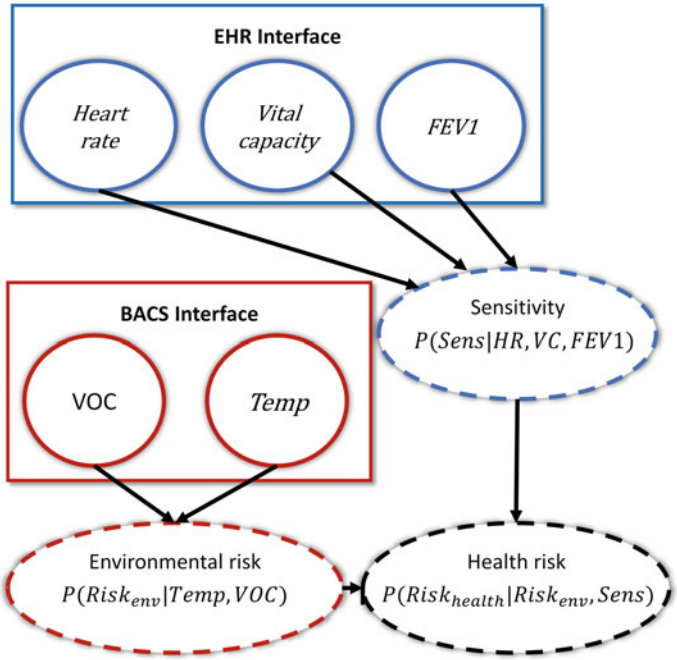


Fig. 15.6 Example of a simple Bayesian network with eight nodes. Comparable to Fig. 15.5, the input data consist of the forced expiratory pressure in 1 s ($FEV1$), heart rate (HR) and vital capacity (VC) from the EHR, and of temperature ($Temp$) and level of volatile organic compounds (VOC) from the BACS. These nodes each influence the conditional probabilities of the environmental risk $P(Risk_{env}|Temp, VOC)$ and the occupant's sensitivity $P(Sens|HR, VC, FEV1)$, respectively, indicated by arrows (edges). The probability of health risk $P(Risk_{health}|Risk_{env}, Sens)$ in turn is influenced by both the sensitivity and the environmental risk, and could be used to trigger feedback actions depending on its value

cussed. The main focus of this work was set on the analysis of possible approaches of artificial intelligence to estimate an individual environmental risk factor.

The estimation of an environmental health risk factor with respect to the occupant's health condition is a quite novel research topic. The system is required to combine knowledge from different disciplines in order to estimate the health risk based on the measured environmental parameters and the individual health information. This risk factor estimation differs from usual decision support and classification systems. In fact, the lack of a measurable target value for the comparison of the predicted value by the expert system and the actual value complicates the system design and evaluation.

Nevertheless, every approach described is able to model ambiguous and uncertain data. The approaches differ in the amount of possibilities to adapt the model on the basis of available data. Approaches like fuzzy cognitive maps and Bayesian networks are quite similar in the structure but are based on different computation methods.

Modeling the knowledge using belief factors within a belief rule base is a quite simple way to model the expert system. In consequence, the possibility to model complex relationships between the parameters is limited using belief rule bases. Still, the concept behind the approach is easy to understand, allowing for quick development and adaption after system evaluations.

More options to model an expert system are given by the fuzzy rule base approach. Thereby, knowledge can be modeled by the definition of the input membership functions, the fuzzy rule base, the output membership functions, and the method to calculate crisp output values by defuzzification. The many ways to influence system behavior complicate the development process and may impede the creation of a reasonable system. Accordingly, a fuzzy rule base might be used in cases when belief rule bases don't offer enough design options.

Fuzzy cognitive maps provide a clear graph-based approach for the modeling of system components and their relations. The knowledge is thereby represented via state values and edge weights. An interesting research topic is the investigation about how the initial state values can be estimated from the measured environmental data and provided health information.

Acknowledgements Funded by the Lower Saxony Ministry of Science and Culture within the Lower Saxony "Vorab" of the Volkswagen Foundation and supported by the Center for Digital Innovations (ZDIN).

References

1. Abu-Nasser B (2017) Medical expert systems survey. Int J Eng Inf Syst (IJEAIS) 1(7):218–224
2. Al-Dmour JA, Sagahyroon A, Al-Ali AR, Abusnana S (2019) A fuzzy logic-based warning system for patients classification. Health Inform J 25(3):1004–1024
3. Anninou AP, Groumpos PP, Panagiotis P (2013) Modeling health diseases using competitive fuzzy cognitive maps, pp 88–95
4. Argacha JF, Bourdrel T, Van De Borne P (2018) Ecology of the cardiovascular system: a focus on air-related environmental factors. Trends Cardiovasc Med 28(2):112–126
5. Bandyopadhyay S, Wolfson J, Vock DM, Vazquez-Benitez G, Adomavicius G, Elidrisi M, Johnson PE, O'Connor PJ (2015) Data mining for censored time-to-event data: a bayesian network model for predicting cardiovascular risk from electronic health record data. Data Min Knowl Discov 29(4):1033–1069
6. Bönisch U, Böhme A, Kohajda T, Mögel I, Schütze N, von Bergen M, Simon JC, Lehmann I, Polte T (2012) Volatile organic compounds enhance allergic airway inflammation in an experimental mouse model. PLoS One 7(7)
7. Bouchaala L, Masmoudi A, Gargouri F, Rebai A (2010) Improving algorithms for structure learning in bayesian networks using a new implicit score. Expert Syst Appl 37(7):5470–5475
8. Cakmak S, Dales RE, Liu L, Kauri LM, Lemieux CL, Hebbern C, Zhu J (2014) Residential exposure to volatile organic compounds and lung function: results from a population-based cross-sectional survey. Environ Pollut 194:145–151
9. Calenic B, Miricescu D, Greabu M, Kuznetsov AV, Troppmair J, Ruzsanyi V, Amann A (2015) Oxidative stress and volatile organic compounds: interplay in pulmonary, cardio-vascular, digestive tract systems and cancer. Open Chem 1(open-issue)
10. Calzada A, Liu J, Nugent CD, Wang H, Martinez L (2014) Sensor-based activity recognition using extended belief rule-based inference methodology, pp 2694–2697

11. Chang L, Zhou ZJ, You Y, Yang L, Zhou Z (2016) Belief rule based expert system for classification problems with new rule activation and weight calculation procedures. Inf Sci 336:75–91
12. Chen R, Li T, Cai J, Yan M, Zhao Z, Kan H (2014) Extreme temperatures and out-of-hospital coronary deaths in six large chinese cities. J Epidemiol Community Health 68(12):1119–1124
13. Colombo D, Maathuis MH (2014) Order-independent constraint-based causal structure learning. J Mach Learn Res 15(1):3741–3782
14. Cooper GF, Herskovits E (1992) A bayesian method for the induction of probabilistic networks from data. Mach Learn 9(4):309–347
15. Dahlquist M, Raza A, Bero-Bedada G, Hollenberg J, Lind T, Orsini N, Sjögren B, Svensson L, Ljungman PL (2016) Short-term departures from an optimum ambient temperature are associated with increased risk of out-of-hospital cardiac arrest. Int J Hyg Environ Health 219(4–5):389–397
16. Deleawe S, Kusznir J, Lamb B, Cook DJ (2010) Predicting air quality in smart environments. J Ambient Intell Smart Environ 2(2):145–154
17. Flessner J, Frenken M (2018) High level modeling of building automation and control systems based on perceptual knowledge. In: Life sciences conference (LSC). IEEE
18. Flessner J, Frenken M (2019) Towards perceptual computing in bacs: an air quality assistant based on fuzzy logic and perceptual knowledge, pp 2921–2926
19. Fong ACM , Fong B (2012) Home telemedicine system for chronic respiratory disease surveillancean automated solution for disease control and management to combat the health impact of indoor air pollution. In: 2012 7th IEEE conference on industrial electronics and applications (ICIEA). IEEE, pp 472–476
20. Franck U, Odeh S, Wiedensohler A, Wehner B, Herbarth O (2011) The effect of particle size on cardiovascular disorders - the smaller the worse. Sci Total Env 409(20):4217–4221
21. Gao N, Li C, Ji J, Yang Y, Wang S, Tian X, Kai-Feng X (2019) Short-term effects of ambient air pollution on chronic obstructive pulmonary disease admissions in beijing, china (2013–2017). Int J Chronic Obstr Pulm Dis 14:297
22. Gore RW, Deshpande DS (2017) An approach for classification of health risks based on air quality levels. In: 2017 1st international conference on intelligent systems and information management (ICISIM). IEEE, pp 58–61
23. Gorgulu O, Akilli A (2016) Use of fuzzy logic based decision support systems in medicine. Stud Ethno-Med 10(4):393–403
24. Haberzettl P, OToole TE, Bhatnagar A, Conklin DJ (2016) Exposure to fine particulate air pollution causes vascular insulin resistance by inducing pulmonary oxidative stress. Environ Health Perspect 124(12):1830–1839
25. Heinrich J, Schikowski T (2018) Copd patients as vulnerable subpopulation for exposure to ambient air pollution. Curr Environ Health Rep 5(1):70–76
26. Hensel M, Geppert D, Kersten JF, Stuhr M, Lorenz J, Wirtz S, Kerner T (2018) Association between weather-related factors and cardiac arrest of presumed cardiac etiology: a prospective observational study based on out-of-hospital care data. Prehospital Emerg Care 22(3):345–352
27. Himes BE, Dai Y, Kohane IS, Weiss ST, Ramoni MF (2009) Prediction of chronic obstructive pulmonary disease (COPD) in asthma patients using electronic medical records. J Am Med Inform Assoc 16(3):371–379
28. HL7. Fast healthcare interoperable resources (FHIR). http://wiki.hl7.de/. Accessed: 2020-07-22
29. Hong E, Lee S, Kim G-B, Kim T-J, Kim H-W, Lee K, Son B-S (2018) Effects of environmental air pollution on pulmonary function level of residents in korean industrial complexes. Int J Environ Res Public Health 15(5):834
30. Hossain MS, Ahmed F, Andersson K et al (2017) A belief rule based expert system to assess tuberculosis under uncertainty. J Medical Syst 41(3):43
31. Hossain MS, Andersson K, Naznin S (2015) A belief rule based expert system to diagnose measles under uncertainty, pp 17–23
32. Hossain MS, Rahaman S, Mustafa R, Andersson K (2018) A belief rule-based expert system to assess suspicion of acute coronary syndrome (acs) under uncertainty. Soft Comput 22(22):7571–7586

33. Jiang Y, Li K, Tian L, Piedrahita R, Yun X, Mansata O, Lv Q, Dick RP, Hannigan M, Shang L (2011) Maqs: a personalized mobile sensing system for indoor air quality monitoring. In: Proceedings of the 13th international conference on Ubiquitous computing, pp 271–280
34. Khaper N, Bailey CDC, Ghugre NR, Reitz C, Awosanmi Z, Waines R, Martino TA (2018) Implications of disturbances in circadian rhythms for cardiovascular health: a new frontier in free radical biology. Free Radic Biol Med 119:85–92
35. Kim J, Kim H (2017) Influence of ambient temperature and diurnal temperature range on incidence of cardiac arrhythmias. Int J Biometeorol 61(3):407–416
36. Kong G, Dong-Ling X, Body R, Yang J-B, Mackway-Jones K, Carley S (2012) A belief rule-based decision support system for clinical risk assessment of cardiac chest pain. Eur J Oper Res 219(3):564–573
37. Kosko B et al (1986) Fuzzy cognitive maps. Int J Man-Mach Stud 24(1):65–75
38. Kurt OK, Zhang J, Pinkerton KE (2016) Pulmonary health effects of air pollution. Curr Opin Pulm Med 22(2):138
39. Lam W, Bacchus F (1994) Learning bayesian belief networks: an approach based on the mdl principle. Comput Intell 10(3):269–293
40. Larrañaga P, Poza M, Yurramendi Y, Murga RH, Kuijpers CMH (1996) Structure learning of bayesian networks by genetic algorithms: a performance analysis of control parameters. IEEE Trans Pattern Anal Mach Intell 18(9):912–926
41. Li KF (2013) Smart home technology for telemedicine and emergency management. J Ambient Intell Humaniz Comput 4(5):535–546
42. Liu KF-R, Lu C-F, Chen C-W, Shen Y-S (2012) Applying bayesian belief networks to health risk assessment. Stoch Environ Res Risk Assess 26(3):451–465
43. Liu J, Martinez L, Calzada A, Wang H (2013) A novel belief rule base representation, generation and its inference methodology. Knowl-Based Syst 53:129–141
44. Lucas PJ, Gaag LC, Abu-Hanna A (2004) Bayesian networks in biomedicine and health-care. Artif Intell Med 30:201–214
45. Malmir B, Amini M, Chang SI (2017) A medical decision support system for disease diagnosis under uncertainty. Expert Syst Appl 88:95–108
46. Margaritis D (2003) Learning bayesian network model structure from data. Technical report, Carnegie-Mellon Univ Pittsburgh Pa School of Computer Science
47. Nannan Panday RS, Minderhoud TC, Alam N, Nanayakkara PWB (2017) Prognostic value of early warning scores in the emergency department (ed) and acute medical unit (amu): a narrative review. Eur J Intern Med 45:20–3
48. OpenEHR. http://openehr.org/. Accessed: 2020-07-22
49. Papageorgiou EI (2011) A new methodology for decisions in medical informatics using fuzzy cognitive maps based on fuzzy rule-extraction techniques. Appl Soft Comput 11(1):500–513
50. Patwary MJA, Akter S, Mahmud T (2014) An expert system to detect uterine cancer under uncertainty. IOSR J Comput Eng (IOSR-JCE), e-ISSN, pp 2278–0661
51. Pearl J (1985) Bayesian netwcrks: a model cf self-activated memory for evidential reasoning. In: Proceedings of the 7th conference of the cognitive science society, University of California, Irvine, CA, USA, pp 15–17
52. Polichetti G, Cocco S, Spinali A, Trimarco V, Nunziata A (2009) Effects of particulate matter (pm10, pm2. 5 and pm1) on the cardiovascular system. Toxicology 261(1-2):1–8
53. Roenneberg T, Kantermann T, Juda M, Vetter C, Allebrandt KV (2013) Light and the human circadian clock, pp 311–331
54. Rotmensch M, Halpern Y, Tlimat A, Horng S, Sontag D (2017) Learning a health knowledge graph from electronic medical records. Sci Rep 7(1):1–11
55. Rumchev K, Brown H, Spickett J (2007) Volatile organic compounds: do they present a risk to our health? Rev Environ Health 22(1):39
56. Rumchev K, Spickett J, Bulsara M, Phillips M, Stick S (2004) Association of domestic exposure to volatile organic compounds with asthma in young children. Thorax 59(9):746–751
57. Samuel OW, Omisore MO, Ojokoh BA (2013) A web based decision support system driven by fuzzy logic for the diagnosis of typhoid fever. Expert Syst Appl 40(10):4164–4171

58. Shaddick G, Thomas ML, Green A, Brauer M, van Donkelaar A, Burnett R, Chang HH, Cohen A, Van Dingenen R, Dora C, Gumy S, Liu Y, Martin R, Waller LA, West J, Zidek JV, Prüss-Ustün A (2018) Data integration model for air quality: a hierarchical approach to the global estimation of exposures to ambient air pollution. J R Stat Soc Ser C (Appl Stat) 67(1):231–253
59. Shiue I, Perkins DR, Bearman N (2015) Inverted u-shape relationships of the weather as biometeorological and hospital admissions due to carcinoma in situ and benign neoplasm in germany in 2009–2011. Environ Sci Pollut Res 22(12):9378–9399
60. Shiue I, Perkins DR, Bearman N (2016) Hospital admissions due to diseases of arteries and veins peaked at physiological equivalent temperature- 10 to 10 c in germany in 2009–2011. Environ Sci Pollut Res 23(7):6159–6167
61. Shiue I, Perkins DR, Bearman N (2016) Relationships of physiologically equivalent temperature and hospital admissions due to i30–i51 other forms of heart disease in germany in 2009–2011. Environ Sci Pollut Res 23(7):6343–6352
62. Smith GS, Van Den Eeden SK, Garcia C, Shan J, Baxter R, Herring AH, Richardson DB, Van Rie A, Emch M, Gammon MD (2016) Air pollution and pulmonary tuberculosis: a nested case–control study among members of a northern california health plan. Environ Health Perspect 124(6):761–768
63. Spirtes P, Glymour CN, Scheines R, Heckerman D (2000) Causation, prediction, and search. MIT press
64. Stylios CD, Georgopoulos VC, Malandraki GA, Chouliara S (2008) Fuzzy cognitive map architectures for medical decision support systems. Appl Soft Comput 8(3):1243–1251
65. To T, Zhu J, Larsen K, Simatovic J, Feldman L, Ryckman K, Gershon A, Diane Lougheed M, Licskai C, Chen H et al (2016) Progression from asthma to chronic obstructive pulmonary disease. Is air pollution a risk factor? Am J Respir Crit Care Med 194(4):429–438
66. Tsamardinos I, Brown LE, Aliferis CF (2006) The max-min hill-climbing bayesian network structure learning algorithm. Mach Learn 65(1):31–78
67. Verma T, Pearl J (1991) Equivalence and synthesis of causal models. UCLA, Computer Science Department
68. Villeneuve PJ, Jerrett M, Su J, Burnett RT, Chen H, Brook J, Wheeler AJ, Cakmak S, Goldberg MS (2013) A cohort study of intra-urban variations in volatile organic compounds and mortality, toronto, canada. Environ Pollut 183:30–39
69. Wang F, Li C, Liu W, Jin Y (2012) Effect of exposure to volatile organic compounds (vocs) on airway inflammatory response in mice. J Toxicol Sci 37(4):739–748
70. World Health Organization and others (2010) WHO guidelines for indoor air quality: selected pollutants. WHO
71. World Health Organization et al (2006) Who air quality guidelines for particulate matter, ozone, nitrogen dioxide and sulfur dioxide: global update 2005: summary of risk assessment. World Health Organization, Technical report, Geneva
72. Wu Y, McCall J, Corne D (2010) Two novel ant colony optimization approaches for bayesian network structure learning. In: IEEE Congress on Evolutionary Computation. IEEE, pp 1–7
73. Yang J-B, Liu J, Wang J, Sii H-S, Wang H-W (2006) Belief rule-base inference methodology using the evidential reasoning approach-rimer. IEEE Trans Syst Man Cybern-Part A Syst Hum 36(2):266–285
74. Yaramakala S, Margaritis D (2005) Speculative markov blanket discovery for optimal feature selection. In: Fifth ieee international conference on data mining (ICDM'05). IEEE, pp 4–pp
75. Yoda Y, Takagi H, Wakamatsu J, Ito T, Nakatsubo R, Horie Y, Hiraki T, Shima M (2019) Stronger association between particulate air pollution and pulmonary function among healthy students in fall than in spring. Sci Total Environ 675:483–489
76. Yoon HI, Hong Y-C, Cho SH, Kim H, Kim YH, Sohn JR, Kwon M, Park SH, Cho MH, Cheong HK (2010) Exposure to volatile organic compounds and loss of pulmonary function in the elderly. Eur Respir J 36(6):1270–1276
77. Zhou Z-G, Liu F, Jiao L-C, Zhou Z-J, Yang J-B, Gong M-G, Zhang X-P (2013) A bi-level belief rule based decision support system for diagnosis of lymph node metastasis in gastric cancer. Knowl-Based Syst 54:128–136

Correction to: Advances in Artificial Intelligence, Computation, and Data Science

Tuan D. Pham, Hong Yan, Muhammad W. Ashraf, and Folke Sjöberg

Correction to:
T. D. Pham et al. (eds.),
***Advances in Artificial Intelligence, Computation, and Data Science*, Computational Biology 31,**
https://doi.org/10.1007/978-3-030-69951-2

The original version of this book was inadvertently published without a volume number. This has been corrected.

Volume number has been updated to '31' for this book.

The updated online version of the book can be found at
https://doi.org/10.1007/978-3-030-69951-2

T. D. Pham et al. (eds.), *Advances in Artificial Intelligence, Computation, and Data Science*, Computational Biology 31,
https://doi.org/10.1007/978-3-030-69951-2_16

Index

T. D. Pham et al. (eds.), *Advances in Artificial Intelligence, Computation, and Data Science*, Computational Biology 31,
https://doi.org/10.1007/978-3-030-69951-2