Mona Singh (Ed.)

Research in Computational Molecular Biology

LNBI 9649

20th Annual Conference, RECOMB 2016
Santa Monica, CA, USA, April 17–21, 2016
Proceedings

EXTRAS ONLINE

Lecture Notes in Bioinformatics 9649

Subseries of Lecture Notes in Computer Science

More information about this series at http://www.springer.com/series/5381

Mona Singh (Ed.)

Research in Computational Molecular Biology

20th Annual Conference, RECOMB 2016
Santa Monica, CA, USA, April 17–21, 2016
Proceedings

Springer

Editor
Mona Singh
Princeton University
Princeton, NJ
USA

The Paper "10 Years of the International Conference on Research in Computational Molecular Biology (RECOMB)", initially published in LNBI 3909, pp. 546–562, DOI 10.1007/11732990_45, is reprinted in this volume.

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Bioinformatics
ISBN 978-3-319-31956-8 ISBN 978-3-319-31957-5 (eBook)
DOI 10.1007/978-3-319-31957-5

Library of Congress Control Number: 2016934663

LNCS Sublibrary: SL8 – Bioinformatics

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG Switzerland

Preface

This year marked the 20th Annual International Conference on Research in Computational Molecular Biology. RECOMB 2016 was held in Santa Monica, CA, during April 17–21, 2016. This volume contains the 35 extended or short abstracts selected for oral presentation at RECOMB 2016 by the Program Committee (PC). Each of the 172 submissions consisted of a full paper, and was assigned to at least three PC members and reviewed with the help of many external reviewers. Following the initial reviews, final decisions were made after an extensive discussion of the submissions among the members of the PC. In 2016, RECOMB allowed parallel submission to the proceedings as well as to a journal. Papers accepted for oral presentation that had simultaneously been submitted to a journal are published as short abstracts. Parallel submissions that had not appeared in a journal by the time of the conference were to be deposited in the preprint server arxiv.org. All other papers that were accepted for RECOMB 2016 were invited for submission to an edited journal version of a special issue of the *Journal of Computational Biology*. In addition to the paper presentations, RECOMB 2016 featured six invited keynote talks by leading scientists worldwide. The keynote speakers were Karen Adelman (National Institute of Environmental Health Sciences at the National Institutes of Health), Phil Bradley (Fred Hutchinson Cancer Research Center), Peter S. Kim (Stanford University), Rob Knight (University of California, San Diego), Leonid Kruglyak (University of California, Los Angeles), and Teresa Przytycka (National Center for Biotechnology Information at the National Institutes of Health). Following the tradition started at RECOMB 2010, RECOMB 2016 also featured highlight talks presenting computational biology papers that were published in journals during the last 18 months. There were 29 highlight submissions, six of which were selected for oral presentation at the main conference.

The success of RECOMB depends on the effort, dedication, and devotion of many colleagues. I especially thank the Organizing Committee chair, Eleazar Eskin, for hosting the 2016 conference and marshalling the entire endeavor, including the Mike Waterman Symposium, three satellite meetings, and the main conference; Ting Chen for substantial organizational help; Danielle Everts for administrative support; the Steering Committee and especially its chair, Bonnie Berger, for help, advice, and support throughout the process; Teresa Przytycka (Program Chair of RECOMB 2015) for answering my many questions and sharing her experiences with me; Donna Slonim for chairing the highlights track; Fabio Vandin for chairing the posters track; Fengzhu Sun for organizing the Mike Waterman Symposium; Carl Kingsford, Alex Schoenhuth, Fabio Vandin, and Barbara Engelhart for chairing the satellite meetings; the main conference and satellite PC members and external reviewers for their timely reviews of assigned papers despite their busy schedules; the authors of the papers, highlights, and posters for their scientific contributions; and all the attendees for their enthusiastic participation in the conference. We also thank the International Society of Computational Biology (ISCB) and the National Science Foundation for student support.

Finally, on this occasion of the 20^{th} anniversary of RECOMB, the entire RECOMB community thanks Sorin Istrail, Pavel Pevzner, and Michael Waterman for having the vision to start this conference series back in 1997.

February 2016 Mona Singh

Organization

Steering Committee

Bonnie Berger	MIT, USA
Serafim Batzoglou	Stanford University, USA
Michal Linial	The Hebrew University of Jerusalem, Israel
Martin Vingro	Max Planck Institute for Informatics, Germany
Sorin Istrail	Brown University, USA
Vineet Bafna	University of California, San Diego, USA

Program Chair

Mona Singh	Princeton University, USA

Organizing Committee Chair

Eleazar Eskin	University of California, Los Angeles, USA

Poster Chair

Fabio Vandin	University of Padua, Italy

Highlights Chair

Donna Slonim	Tufts University, USA

Program Committee

Tatsuya Akutsu	Kyoto University, Japan
Can Alkan	Bilkent University, Turkey
Rolf Backofen	Albert-Ludwigs-Universität Freiburg, Germany
Chris Bailey-Kellogg	Dartmouth College, USA
Nuno Bandeira	University of California, San Diego, USA
Ziv Bar-Joseph	Carnegie Mellon University, USA
Alexis Battle	Johns Hopkins University, USA
Niko Beerenwinkel	ETH Zurich, Switzerland
Panayiotis Takis Benos	University of Pittsburgh, USA
Bonnie Berger	Massachusetts Institute of Technology, USA
Jadwiga Bienkowska	Biogenidec, USA
Mathieu Blanchette	McGill University, Canada
Philip Bradley	Fred Hutchinson Cancer Research Center, USA

Michael R. Brent	Washington University, USA
Tony Capra	Vanderbilt University, USA
Kevin Chen	Rutgers University, USA
Lenore Cowen	Tufts University, USA
Colin Dewey	University of Wisconsin – Madison, USA
Bruce Donald	Duke University, USA
Dannie Durand	Carnegie Mellon University, USA
Nadia El-Mabrouk	University of Montreal, Canada
Barbara Engelhardt	Princeton University, USA
Eleazar Eskin	University of California, Los Angeles, USA
Dario Ghersi	University of Nebraska at Omaha, USA
Anna Goldenberg	SickKids Research Institute, Canada
Casey Greene	Geisel School of Medicine at Dartmouth, USA
Eran Halperin	University of California, Berkeley, USA
Tao Jiang	University of California, Riverside, USA
Igor Jurisica	Ontario Cancer Institute, Canada
Tamer Kahveci	University of Florida, USA
Olga Kalinina	Max Planck Institute for Informatics, Germany
Carl Kingsford	Carnegie Mellon University, USA
Mehmet Koyuturk	Case Western Reserve University, USA
Smita Krishnaswamy	Yale University, USA
Rui Kuang	University of Minnesota Twin Cities, USA
Jens Lagergren	KTH Royal Institute of Technology, Sweden
Christina Leslie	Sloan Kettering, USA
Michal Linial	The Hebrew University of Jerusalem, Israel
Tobias Marschall	Saarland University/Max Planck Institute for Informatics, Germany
Paul Medvedev	Pennsylvania State University, USA
Tijana Milenkovic	University of Notre Dame, USA
Satoru Miyano	University of Tokyo, Japan
Bernard Moret	EPFL, Switzerland
Sara Mostafavi	University of British Columbia, Canada
Vincent Moulton	University of East Anglia, UK
Chad Myers	University of Minnesota, USA
Laxmi Parida	IBM T.J. Watson Research Center, USA
Itsik Peer	Columbia University, USA
Jian Peng	University of Illinois at Urbana – Champaign, USA
Nico Pfeifer	Max Planck Institute for Informatics, Germany
Natasa Przulj	Imperial College London, UK
Teresa Przytycka	NIH, USA
Ben Raphael	Brown University, USA
Knut Reinert	FU Berlin, Germany
Maga Rowicka	University of Texas Medical Branch, USA
Marie-France Sagot	Inria Grenoble Rhône-Alpes and Université de Lyon 1, Villeurbanne, France
S. Cenk Sahinalp	Simon Fraser University, Canada

Marcel Schulz	Saarland University, Germany
Russell Schwartz	Carnegie Mellon University, USA
Amarda Shehu	George Mason University, USA
Mona Singh	Princeton University, USA
Donna Slonim	Tufts University, USA
Fengzhu Sun	University of Southern California, USA
Glenn Tesler	University of California, San Diego, USA
Fabio Vandin	University of Padua, Italy
Martin Vingron	Max Planck Institut für Molekulare Genetik, Germany
Olga Vitek	Northeastern University, USA
Jerome Waldispuhl	McGill University, Canada
Wei Wang	University of California, Los Angeles, USA
Tandy Warnow	The University of Illinois – Urbana Champaign, USA
Jinbo Xu	Toyota Technological Institute at Chicago, USA
Esti Yeger-Lotem	Ben-Gurion University, Israel

The following paper summarizes the first 10 years of the RECOMB conference and was published in 2006. We include it here in order to commemorate the first 20 years of the RECOMB conference. Later in this volume is a new manuscript summarizing years 11 through 20 of RECOMB.

10 Years of the International Conference on Research in Computational Molecular Biology (RECOMB)*

Sarah J. Aerni and Eleazar Eskin

The RECOMB 10th Year Anniversary Committee

The tenth year of the annual International Conference on Research in Computational Biology (RECOMB) provides an opportunity to reflect on its history. RECOMB has been held across the world, including 6 different countries spanning 3 continents (Table 1). Over its 10 year history, RECOMB has published 373 papers and 170 individuals have served on its various committees. While there are many new faces in RECOMB each year, a significant number of researchers have participated over many years forming the core of the RECOMB community.

Over the past ten years, members of the RECOMB community were key players in many of the advances in Computational Biology during this period. These include the sequencing and assembly of the human genome, advances in sequence comparison, comparative genomics, genome rearrangements and the HapMap project among others.

Table 1. The locations and dates of each year of RECOMB. The program and conference chair are listed for each conference in the final two columns.

	Location	**Dates**	**Program Chair**	**Conference Chair**
1997	Santa Fe, USA	January 20-23	Michael Waterman	Sorin Istrail
1998	New York, USA	March 22-25	Pavel Pevzner	Gary Benson
1999	Lyon, France	April 11-14	Sorin Istrail	Mireille Régnier
2000	Tokyo, Japan	April 8-11	Ron Shamir	Satoru Miyano
2001	Montreal, Canada	April 22-25	Thomas Lengauer	David Sankoff
2002	Washington, USA	April 18-21	Eugene Myers	Sridhar Hannenhalli
2003	Berlin, Germany	April 10-13	Webb Miller	Martin Vingron
2004	San Diego, USA	March 27-31	Dan Gusfield	Philip Bourne
2005	Boston, USA	May 14-18	Satoru Miyano	Jill Mesirov, Simon Kasif
2006	Venice, Italy	April 2-5	Alberto Apostolico	Concettina Guerra

10 Years of RECOMB Papers

Over RECOMB's 10 year history, 731 authors have published a total of 373 papers in the conference proceedings. These papers span the diversity of research areas in Computational Biology and present many new computational techniques for the analysis of biological data.

* reprint from RECOMB 2006, LNBI 3909, pp. 546–562, DOI 10.1007/11732990_45

It should be noted that some authors have variances in how names appear throughout the years, including differing first names, initials, and middle names. While every effort was made to normalize the names, any such error could lead to the skewing of data and there may be small errors in the reporting of individual participation throughout the paper.

As a preliminary analysis, we consider the number of papers for each researcher that has appeared throughout the 10 years of RECOMB in the proceedings. In such a measure, Richard Karp who has authored 12 different papers in RECOMB throughout the 10 years would be the top participant.

Using the graph in Figure 1, we can identify the most collaborative members of the RECOMB community (hubs in a protein network). The most collaborative authors are the individuals that have the most number of co-authors. Ron Shamir is the most collaborative RECOMB author with 22 co-authors (Table 3).

Similarly, we can identify which groups of authors have had the most success working together (complexes in protein networks). The team of Eric S. Lander, Bonnie

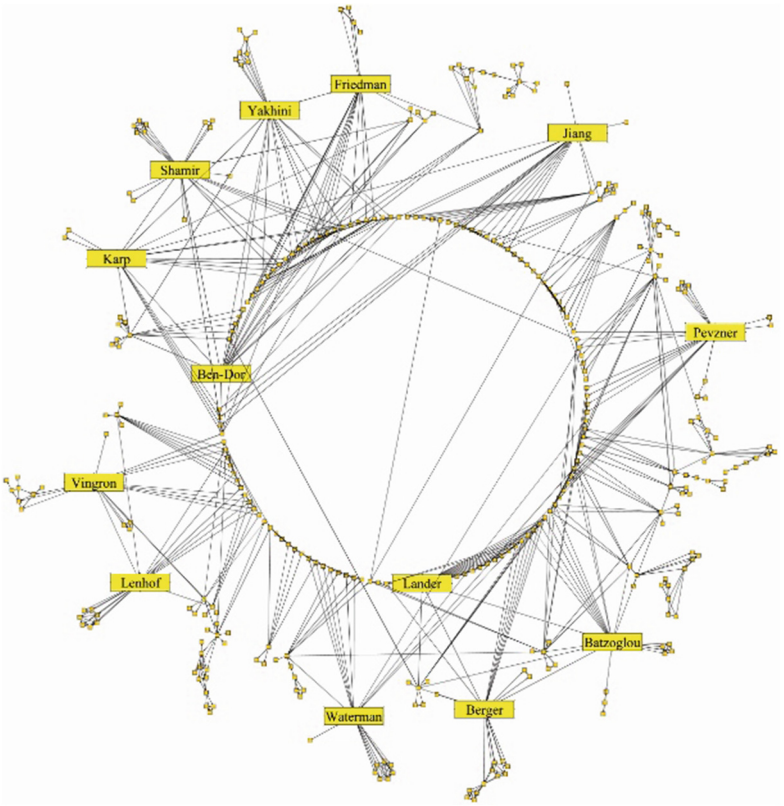


Fig. 1. Graphical view of interactions between RECOMB authors represented as a "protein interaction network" (giant component). Vertices of the graph represent authors while edges connect vertices corresponding to co-authors. Authors whose names are displayed are authors who have at least 16 coauthors.

Table 2. RECOMB's most prolific authors. The table identifies authors who have published at least 4 papers in RECOMB.

Author	Number of Papers	Author	Number of Papers
Richard Karp	12	Serafim Batzoglou	6
Ron Shamir	11	Dan Gusfield	6
Pavel Pevzner	11	Webb Miller	5
Bonnie Berger	10	Fengzhu Sun	5
Amir Ben-Dor	10	Ralf Bundschuh	5
Nir Friedman	9	Jeremy Buhler	5
Eugene Myers	9	Jens Lagergren	5
Zohar Yakhini	9	Roded Sharan	4
Tao Jiang	9	Benno Schwikowski	4
Benny Chor	8	Nancy Amato	4
Michael Waterman	8	Eran Halperin	4
David Sankoff	8	Zheng Zhang	4
Martin Vingron	7	Martin Farach-Colton	4
Ting Chen	7	Sorin Istrail	4
Steven Skiena	7	Vlado Dancík	4
Eric Lander	7	Golan Yona	4
Hans-Peter Lenhof	6	Dannie Durand	4
John Kececioglu	6	Mathieu Blanchette	4
Vineet Bafna	6	Adam Siepel	4
Bruce Donald	6	Tatsuya Akutsu	4
David Haussler	6	Eran Segal	4
Lior Pachter	6	Thomas Lengauer	4

Table 3. RECOMB contributors with more than 10 co-authors. For each author the number of individuals with whom they have coauthored papers is listed.

Author Name	Num of Coauthors	Author Name	Num of Coauthors
Ron Shamir	22	Hans-Peter Lenhof	16
Serafim Batzoglou	20	Vlado Dancik	14
Bonnie Berger	20	Steven Skiena	14
Pavel Pevzner	20	Benny Chor	14
Michael Waterman	19	Lydia Kavraki	13
Zohar Yakhini	19	Bruce Donald	13
Tao Jiang	18	Martin Farach-Colton	12
Richard Karp	18	Sorin Istrail	12
Eric Lander	18	Lior Pachter	12
Nir Friedman	18	Eugene Myers	11
Amir Ben-Dor	17	David Sankoff	11
Martin Vingron	17	Vineet Bafna	11

Berger and Serafim Batzoglou have published 3 papers together and are the only group of three authors which have published more than two papers. The most prolific pair of authors is Amir Ben-Dor and Zohar Yakhini who have published 7 papers together. 21 pairs of authors have published at least 3 papers as shown in Table 4.

Relationships between individual authors can be established in other ways as well. In Figure 2 we analyze the relationships between the most prolific authors (Table 2). By examining the relationships between individuals as advisors in both PhD and post-doctoral positions, the connections between the most prolific authors can be seen as a phylogeny. In addition, the individuals are shown on a timeline indicating the times at which they first began publishing in the field of Computational Biology.

We manually classified each paper into one of 16 categories: Protein structure analysis, Molecular Evolution, Sequence Comparison, Motif Finding, Sequence analysis, Population genetics/SNP/Haplotyping, Physical and Genetic Mapping, Gene Expression, Systems Biology, RNA Analysis, Genome rearrangements, Computational Proteomics, Recognition of Genes, Microarray design, DNA computing and Other. Using these classifications, we can observe which authors have written the most about a single topic and which authors have written about the most topics. Both Bonnie Berger

Table 4. Coauthor Pairs. All pairs of authors who have written 3 or more papers accepted by RECOMB throughout the 10 year history of the conference are listed in the table.

Author Names		**Number of Papers**
Amir Ben-Dor	Zohar Yakhini	7
Bonnie Berger	Eric Lander	4
Zheng Zhang	Webb Miller	4
Serafim Batzoglou	Bonnie Berger	3
Serafim Batzoglou	Eric Lander	3
Amir Ben-Dor	Benny Chor	3
Amir Ben-Dor	Richard Karp	3
Amir Ben-Dor	Benno Schwikowski	3
Benny Chor	Tamir Tuller	3
Tao Jiang	Richard Karp	3
Richard Karp	Ron Shamir	3
David Haussler	Adam Siepel	3
Eric Lander	Jill Mesirov	3
Fengzhu Sun	Ting Chen	3
Ralf Zimmer	Thomas Lengauer	3
Bruce Donald	Christopher Langmead	3
Bruce Donald	Ryan Lilien	3
Nir Friedman	Yoseph Barash	3
Michael Hallett	Jens Lagergren	3
Guang Song	Nancy Amato	3
Eran Segal	Daphne Koller	3

and Benny Chor have contributed the most papers (6) on a single topic, Protein Structure Analysis and Molecular Evolution respectively. Table 5 shows the top contributors in a single area.

On the opposite end of the spectrum are the authors who contributed papers on different topics (Table 6).

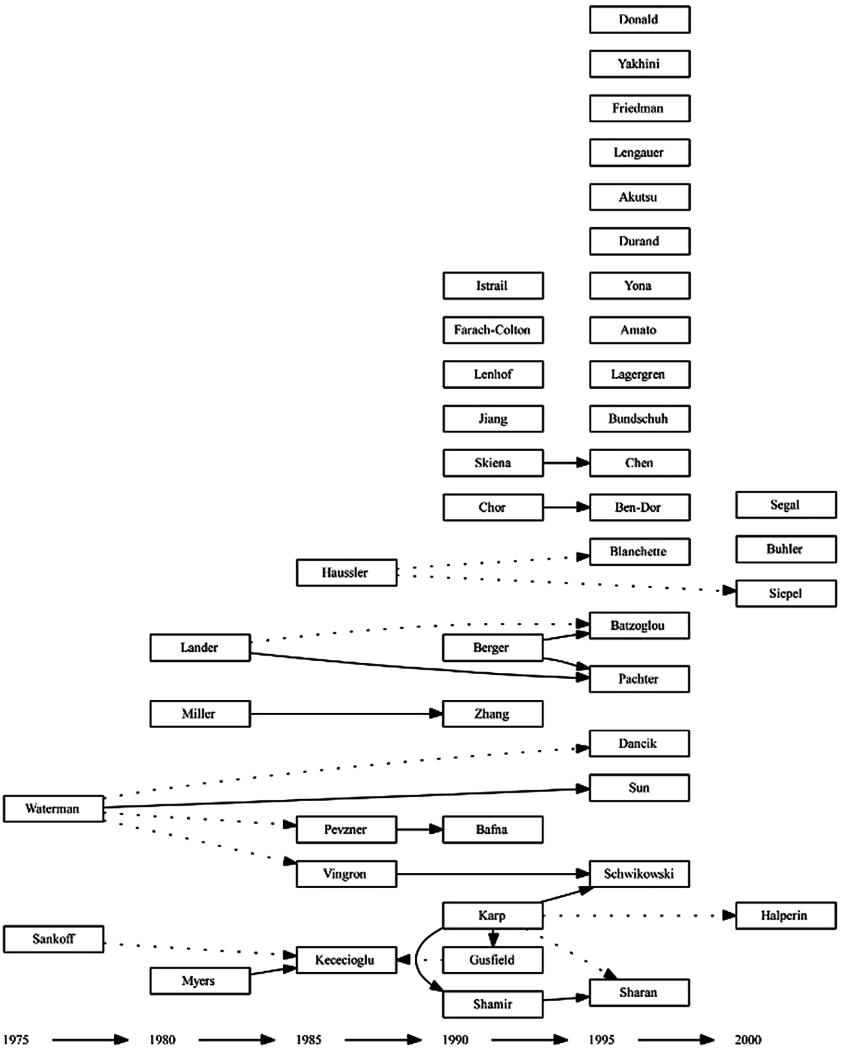


Fig. 2. Phylogeny of Authors. In this figure authors are organized across a timeline representing their earliest publications in the field of Computational Biology. Solid lines indicate PhD advisors, while dotted lines represent postdoctoral advisors. While we attempted to accurately link the timeline and RECOMB authors/genealogy, the figure represents only approximate time estimates and approximate topology of the RECOMB tree.

Table 5. Most consistent authors. For each author in the table, a subject is indicated for which he or she has written at least 3 papers. The number of papers in the 10 years of RECOMB by the author on the given subject is indicated.

Author Name	Author Name	Num of papers
Benny Chor	Molecular Evolution	6
Bonnie Berger	Protein structure analysis	6
David Sankoff	Genome rearrangements	5
Ralf Bundschuh	Sequence Comparison	5
Bruce Donald	Protein structure analysis	5
Nir Friedman	Gene Expression	5
Jens Lagergren	Molecular Evolution	5
Amir Ben-Dor	Gene Expression	4
Richard Karp	Physical and Genetic Mapping	4
Webb Miller	Sequence Comparison	4
David Haussler	Molecular Evolution	4
Zohar Yakhini	Gene Expression	4
Lior Pachter	Recognition of Genes	4
Dannie Durand	Molecular Evolution	4
Eugene Myers	Sequence Comparison	3
John Kececioglu	Sequence Comparison	3
Tao Jiang	Physical and Genetic Mapping	3
Ron Shamir	Sequence analysis	3
Michael Waterman	Physical and Genetic Mapping	3
Hans-Peter Lenhof	Protein structure analysis	3
Zheng Zhang	Sequence Comparison	3
Dan Gusfield	Population genetics/SNP/Haplotyping	3
Tandy Warnow	Molecular Evolution	3
Douglas Brutlag	Protein structure analysis	3
Jon Kleinberg	Protein structure analysis	3
Franco Preparata	Sequence analysis	3
Chris Bailey-Kellogg	Protein structure analysis	3
Michael Hallett	Molecular Evolution	3
Jonathan King	Protein structure analysis	3
Jeremy Buhler	Sequence Comparison	3
Kaizhong Zhang	RNA Analysis	3
Nancy Amato	Protein structure analysis	3
Eran Halperin	Population genetics/SNP/Haplotyping	3
Ryan Lilien	Protein structure analysis	3
Tamir Tuller	Molecular Evolution	3

Table 6. Most Diverse Authors. These are authors spanning the largest number of subjects. Authors are given who have papers in RECOMB in more than 4 subjects.

# of Subjects	Author Name	# of Subjects	Author Name
8	Pavel A. Pevzner	5	Fengzhu Sun
7	Steven S. Skiena	5	Ting Chen
7	Richard M. Karp	4	Tatsuya Akutsu
7	Ron Shamir	4	Bonnie Berger
6	Amir Ben-Dor	4	Hans-Peter Lenhof
6	Tao Jiang	4	Benno Schwikowski
6	Martin Vingron	4	Dan Gusfield
6	Eric S. Lander	4	Thomas Lengauer
6	Zohar Yakhini	4	Vineet Bafna
5	Serafim Batzoglou	4	Nir Friedman
5	Eugene W. Myers	4	Eran Segal
5	Michael S. Waterman	4	Roded Sharan

For each author we create a topic profile which is a 16 dimensional vector containing the number of papers of each topic that an individual has published in RECOMB normalized by dividing by the total number of papers published. Intuitively, an author's topic profile represents the areas of research in which the author works on. Not surprisingly, co-authors tend to work on the same topics. The average pairwise Euclidean distance between any two authors topic profile is 1.19 while the average distance between co-authors is only 0.61. Similarly, papers written by the same author tend to be on the same topic. The chances that any two papers are on the same topic are 0.09 while the chance that two papers that share one author is on the same topic is 0.21.

Trends in RECOMB Authors over Time

The number of authors contributing to the conferences has fluctuated with the largest number in 2006 at 134. 1998 represents the year in which the fewest number of authors submitted multiple papers, that is, most authors had a single paper that was accepted to the conference (Table 7).

2006 had the lowest proportion of single-authored papers with only one of the 40 accepted papers showing a single author (Table 8 and Figure 3).

It appears that over the years there is a trend in an increase in the number of authors per paper with a slight decrease in papers per author. This indicates that while there are more authors on any one single paper, authors are less likely to have multiple papers in any given year.

There are multiple ways to gauge the participation of individuals in the conference. One such measure might be to determine the span of years over which individuals have papers appearing in the proceedings. This was measured by determining the years of the

Table 7. "Authors per paper" and "papers per author" statistics

Year	Papers	Authors	Averages	
			Author per Paper	Paper per Author
1997	42	101	2.8	1.2
1998	38	96	2.6	1.0
1999	35	106	3.3	1.0
2000	36	122	3.8	1.1
2001	35	92	2.8	1.1
2002	35	87	2.7	1.1
2003	35	88	2.8	1.1
2004	38	111	3.1	1.1
2005	39	121	3.4	1.1
2006	40	134	3.4	1.1

Table 8. Author Numbers in Papers. The table shows the percent of papers in each that had the given number of authors determined by counting the number of papers with the indicated number of authors and dividing it by the total number of papers in RECOMB in that year.

Year	Percent of papers with given number of authors										
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
1997	19.0	42.9	14.3	9.5	9.5	0.0	0.0	2.4	0.0	0.0	2.4
1998	18.4	31.6	28.9	15.8	5.3	0.0	0.0	0.0	0.0	0.0	0.0
1999	8.6	37.1	22.9	8.6	11.4	5.7	5.7	0.0	0.0	0.0	0.0
2000	2.8	30.6	19.4	22.2	13.9	5.6	0.0	0.0	0.0	0.0	5.6
2001	17.1	31.4	28.6	8.6	8.6	2.9	2.9	0.0	0.0	0.0	0.0
2002	14.3	34.3	25.7	17.1	8.6	0.0	0.0	0.0	0.0	0.0	0.0
2003	8.6	42.9	25.7	5.7	17.1	0.0	0.0	0.0	0.0	0.0	0.0
2004	7.9	39.5	21.1	13.2	10.5	2.6	2.6	2.6	0.0	0.0	0.0
2005	2.6	28.2	28.2	25.6	7.7	2.6	2.6	2.6	0.0	0.0	0.0
2006	2.5	30.0	32.5	12.5	12.5	5.0	2.5	0.0	2.5	0.0	0.0

first followed by the most recent papers of individual authors, and determining the span of years over which they had participated. Using such a measure, ten authors have papers published over a span of all ten years of the conference listed in the table. These authors are Benny Chor, Bonnie Berger, Sampath Kannan, John Kececioglu, Martin Vingron, David Haussler, Pavel Pevzner, Serafim Batzoglou, Dan Gusfield and Tao Jiang.

However, such a measure may not be completely representative of a researcher's participation in the conference. Over the 10 years of RECOMB, no author has contributed to every year of the conference (Table 9).

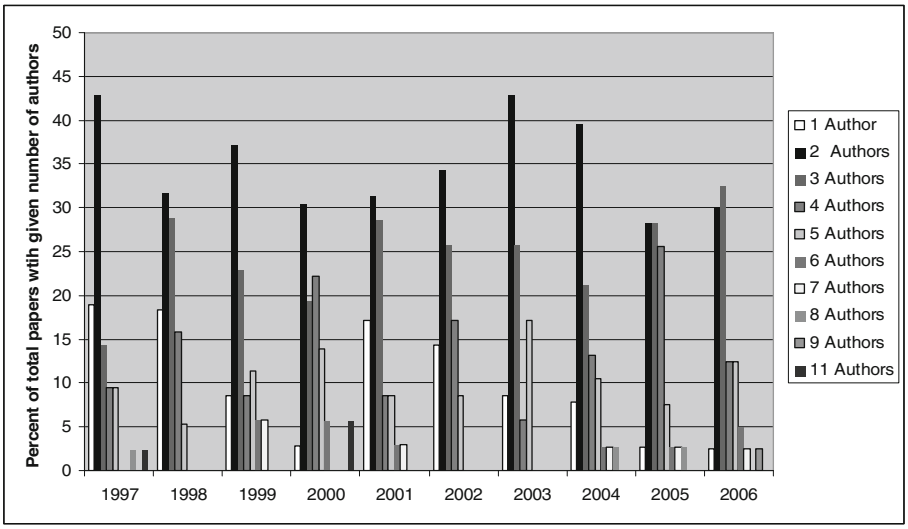


Fig. 3. Distribution of papers with given number of authors over the 10 years of RECOMB

Table 9. Authors with RECOMB papers in most number of years

Author Name	Num of Years	Author Name	Num of Years
Bonnie Berger	9	Tao Jiang	5
Pavel Pevzner	9	Benno Schwikowski	4
Ron Shamir	8	Nancy Amato	4
Amir Ben-Dor	8	Eran Halperin	4
Richard Karp	8	Vineet Bafna	4
Benny Chor	7	Sorin Istrail	4
Zohar Yakhini	7	Webb Miller	4
David Sankoff	7	Vlado Dancík	4
Eugene Myers	6	David Haussler	4
Bruce Donald	6	Mathieu Blanchette	4
Lior Pachter	6	Fengzhu Sun	4
Hans-Peter Lenhof	5	Adam Siepel	4
John Kececioglu	5	Serafim Batzoglou	4
Nir Friedman	5	Dan Gusfield	4
Martin Vingron	5	Jens Lagergren	4
Ting Chen	5	Steven Skiena	4
Ralf Bundschuh	5	Eric Lander	4
Jeremy Buhler	5	Thomas Lengauer	4
Michael Waterman	5		

Trends in RECOMB Paper Topics

As bioinformatics has grown and changed over the 10 years since RECOMB's inception, so have the subjects which comprise the papers accepted at each conference (Table 10). Some subjects, such as Protein Structure Analysis has remained a stronghold in the papers throughout the 10 years of RECOMB. Not only is it the most represented subject over the course of time, at 72 total papers in this field, with a steady portion of the total papers in each year in this field, it entails nearly 30 percent of the accepted papers in 2006.

Table 10. Distribution of topics of RECOMB papers. "Other" category includes more specific subjects such as drug design, DNA denaturization, etc.

Subject	Total	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006
Protein structure analysis	72	16.7	18.4	25.7	27.8	17.1	17.1	11.4	15.8	12.8	30.0
Molecular Evolution	52	4.8	15.8	14.3	13.9	11.4	8.6	11.4	13.2	20.5	25.0
Sequence Comparison	40	28.6	21.1	2.9	8.3	5.7	8.6	2.9	7.9	7.7	10.0
Motif Finding	32	0.0	15.8	5.7	8.3	14.3	8.6	11.4	15.8	7.7	0.0
Sequence analysis	22	0.0	0.0	5.7	5.6	22.9	11.4	8.6	5.3	0.0	2.5
Population genetics/ SNP/ Haplotyping	21	2.4	2.6	0.0	0.0	0.0	11.4	20.0	7.9	7.7	5.0
Physical and Genetic Mapping	20	23.8	7.9	8.6	5.6	0.0	0.0	2.9	0.0	2.6	0.0
Gene Expression	20	0.0	0.0	8.6	11.1	8.6	17.1	5.7	2.6	2.6	0.0
Systems Biology	20	0.0	0.0	5.7	2.8	2.9	2.9	11.4	5.3	12.8	10.0
RNA Analysis	18	0.0	2.6	2.9	2.8	2.9	5.7	2.9	10.5	7.7	10.0
Genome rearrangements	15	9.5	5.3	2.9	2.8	0.0	2.9	5.7	2.6	5.1	2.5
Computational Proteomics	14	0.0	0.0	2.9	2.8	8.6	0.0	2.9	5.3	10.3	5.0
Recognition of Genes	10	7.1	0.0	2.9	2.8	5.7	0.0	0.0	5.3	2.6	0.0
Other	10	0.0	10.5	11.4	2.8	0.0	0.0	0.0	2.6	0.0	0.0
Microarray design	5	2.4	0.0	0.0	2.8	0.0	5.7	2.9	0.0	0.0	0.0
DNA computing	2	4.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

While protein structure remained a consistent part of the RECOMB content, other subjects have fluctuated, disappeared, or gained strength over time. Sequence comparison, which composed well over 25 percent of all papers in the first year of RECOMB, fell to 10 percent of the total content of the 2006 conference. Similarly, Physical and Genetic Mapping which exceeded protein structure analysis in 1997 has completely disappeared in 2006. RNA analysis and Systems Biology have also been growing in popularity since the first papers were accepted in the subjects in 1998 and 1999 respectively.

Computational Proteomics and Population Genetics each represented five percent of the total number of accepted papers. While neither was very abundant in the first

four years of the conference, they seem to be gaining momentum over time. Genome rearrangement has maintained a consistent presence throughout the 10 years of RECOMB. Most notably, however, is the area of molecular evolution which has evolved from a small presence of 4.8 percent of all accepted papers in 1997 to 25 percent of the total accepted papers in 2006.

RECOMB has grown more competitive over time, with an increase in submissions to over 200 in the last three years (Table 11). The number of submissions in 2006 has nearly doubled over the first year of the conference.

Table 11. Paper Acceptance Rates. The table gives the paper acceptance rates based on the number of papers submitted and accepted over the 10 years of RECOMB.

Year	Number Submitted	Number Accepted	Rate
1997	117	43	37%
1998	123	38	31%
1999	147	35	24%
2000	110	36	33%
2001	128	35	27%
2002	118	35	30%
2003	175	35	20%
2004	215	38	18%
2005	217	38	18%
2006	215	40	19%

Table 12. Proportion of USA/Non-USA RECOMB papers

Year	USA	Non-USA
1997	67%	33%
1998	66%	34%
1999	66%	34%
2000	54%	46%
2001	69%	31%
2002	86%	14%
2003	71%	29%
2004	74%	26%
2005	54%	46%
2006	65%	35%

Origins of RECOMB Papers

The first authors of the papers have spanned the globe, representing 25 countries. While US first authors regularly contributed over 60 percent of the papers accepted to the conference, in 2000 and 2005, held in Tokyo and Boston respectively, the split neared 50 percent (Table 12). Most strikingly, over 85 percent of the papers the 2002 conference held in Washington, DC had first authors from US institutions.

Israel, Germany and Canada had first authors contributing papers to nearly every conference (Figure 4). Israel became the second most represented country during 5 years, including 2003 when the conference was held in Germany where 80 percent of non-US authors were from Israel. Canada, Germany and Italy represented the runner-up position during 2 years each. Italy contributed the largest proportion of first authored papers during 2002 when 40% of non-USA first authors were from Italian institutions, which is the second largest percentage in any year.

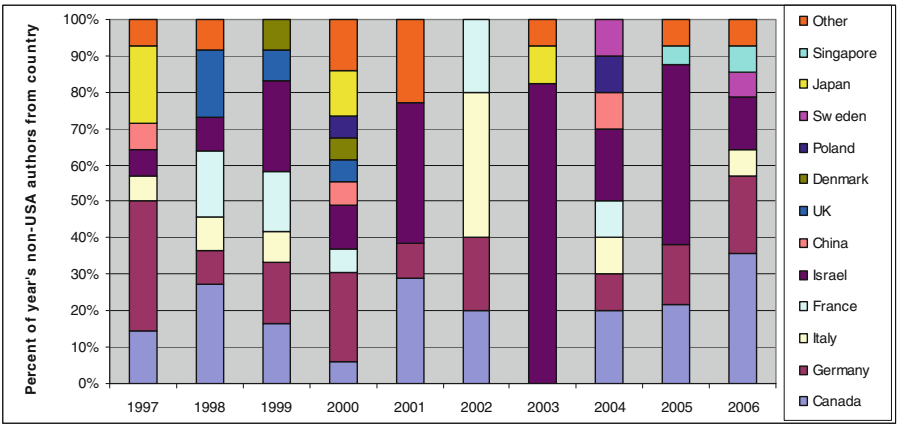


Fig. 4. Distribution of countries of origin of non-US first authors[1]

Throughout RECOMB's history, over 90% of the first authors were involved in the public sector with the exception of a brief interruption in 2001 when just over 11% of first authors were from Industry. In 2002, the conference was hosted by Celera, during which nearly 9% of first authors were involved in the private sector, the second largest amount during the conference's history. However, the contributions from industry have steadily declined since 2002.

RECOMB's Most Cited Papers

Several of the papers published in RECOMB have had a significant influence on research in Computational Biology and have been widely cited. Table 13 contains a list

[1] Category Other includes Chile, Belgium, Australia, Spain, Netherlands, Finland, Switzerland, New Zealand, Austria, and Taiwan.

of the most cited RECOMB papers as of January 2006 according to Google Scholar. A difficulty in obtaining this list is that many of the RECOMB papers are later published in journals and the citations are split between the original RECOMB version and the journal version which may lead to some inaccuracies in calculating the number of citations.

Table 13. RECOMB's most cited papers. The number of citations given in the final column is based on the journal in which they were published, and are accurate as of January 1, 2006 when the citations were last confirmed.

Paper Title	RECOMB Year	Journal	# Citations
Nir Friedman, Michal Linial, Iftach Nachman, Dana Pe'er. "Using Bayesian networks to analyze expression data"	2000	J Comp Biol 2000:7	506
Manolis Kamvysselis, Nick Patterson, Bruce Birren, Bonnie Berger, Eric S. Lander. "Whole-genome comparative annotation and regulatory motif discovery in multiple yeast species"	2003	Nature 2003: 423	385
Amir Ben-Dor, Zohar Yakhini. "Clustering gene expression patterns"	1999	J Comp Biol 1999:6	355
Harmen J. Bussemaker, Hao Li, Eric D. Siggia. "Regulatory element detection using correlation with expression (abstract only)"	2001	Nat Genet 2001:27	265
Amir Ben-Dor, Laurakay Bruhn, Nir Friedman, Iftach Nachman, Michèl Schummer, Zohar Yakhini. "Tissue classification with gene expression profiles"	2000	J Comp Biol 2000:7	245
Serafim Batzoglou, Lior Pachter, Jill P. Mesirov, Bonnie Berger, Eric S. Lander. "Human and mouse gene structure: comparative analysis and application to exon prediction"	2000	Genome Res 2000:10	190
Isidore Rigoutsos, Aris Floratos. "Motif discovery without alignment or enumeration"	1998	Bioinformatics 2000:14	150
Jeremy Buhler, Martin Tompa. "Finding motifs using random projections"	2001	J Comp Biol 2002:9	138
Martin G. Reese, Frank H. Eeckman, David Kulp, David Haussler. "Improved splice site detection in Genie"	1997	J Comp Biol 1997:4	131

(*Continued*)

Table 13. (*Continued*)

Paper Title	RECOMB Year	Journal	# Citations
Haim Kaplan, Ron Shamir, Robert E. Tarjan. "Faster and simpler algorithm for sorting signed permutations by reversals"	1997	SIAM J Comput 1999:29	127
Alberto Caprara. "Sorting by reversals is difficult"	1997	RECOMB 1997	111
Vlado Dancík, Theresa A. Addona, Karl R. Clauser, James E. Vath, Pavel A. Pevzner. "De Novo Peptide Sequencing via Tandem Mass Spectrometry"	1999	J Comp Biol 1999:6	109
Pierluigi Crescenzi, Deborah Goldman, Christos Papadimitriou, Antonio Piccolboni, Mihalis Yannakakis. "On the complexity of protein folding"	1998	J Comp Biol 1998:5	104
Bonnie Berger, Tom Leighton. "Protein folding in the hydrophobic-hydrophilic (HP) is NP-complete"	1998	J Comp Biol 1998:5	89
Donna K. Slonim, Pablo Tamayo, Jill P. Mesirov, Todd R. Golub, Eric S. Lander. "Class prediction and discovery using gene expression data"	2000	RECOMB 2000	87
Mathieu Blanchette. "Algorithms for phylogenetic footprinting"	2001	J Comp Biol 2002:9	84
David Sankoff, Mathieu Blanchette. "Multiple genome rearrangements"	1998	J Comp Biol 1998:5	84
Donna K. Slonim, Leonid Kruglyak, Lincoln Stein, Eric S. Lander. "Building human genome maps with radiation hybrids"	1997	J Comp Biol 1997:4	81

RECOMB Keynote Speaker Series

The conference has been honored to have many excellent speakers throughout the 10 years of the conference. Every year between 7 and 9 distinguished individuals were invited to deliver lectures at the conference in a variety of fields (Table 14).

RECOMB includes a distinguished lecture series which consists of the Stanislaw Ulam Memorial Computational Biology lecture, the Distinguished Biology lecture, and New Technologies lectures delivered by a different set of individuals every year (Table 15) with the exception of 1999 and 2005. In 199 There was no Biology lecture, while in 2005 no distinguished lectures were delivered on new technologies. 2004 included an additional address in which Richard Karp delivered the lecture awarded the Fred Howes Distinguished Service Award.

Table 14. Invited speakers over the 10 years of RECOMB

Year	Speaker names
1997	David Botstein, Sam Karlin, Martin Karplus, Eric Lander, Robert Lipshutz, Jonathan King, Rich Roberts, Temple Smith, Terry Speed
1998	Ruben Abagyan, Charles Cantor, David Cox, Ron Davis, Klaus Guberna tor, Joshua Lederberg, Michael Levitt, David Schwartz, John Yates
1999	Peer Bork, Cyrus Chothia, Gene Myers, John Moult, Pitor Slonimsky, Ed Southern, Peter Willett, John Wooley
2000	Eric Davidson, Walter Gilbert, Takashi Gojobori, Leroy Hood, Minoru Kanehisa, Hans Lehrach, Yvonne Martin, Yusuke Nakamura, Svante Paabo
2001	Mark Adams, Roger Brent, George Church, Franz Lang, Klaus Lindpaintner, Yvonne Martin, Mark Ptashne, Philip Sharp, Matthias Wilm
2002	Ruben Abagyan, Ali Brivanlou, Evan Eichler, Harold Garner, David Ho, Gerry Rubin, Craig Venter, Marc Vidal
2003	Edward Trifonov, Christiane Nüsslein-Volhard, Árpád Furka, Andrew Clark, David Haussler, Arthur Lesk, Dieter Oesterhelt, Terry Speed, Kari Stefansson
2004	Carlos Bustamante, Russell Doolittle, Andrew Fire, Richard Karp, William McGinnis, Deborah Nickerson, Martin Nowak, Christine Orengo, Elizabeth Winzeler
2005	David Altshuler, Wolfgang Baumeister, James Collins, Charles DeLisi, Jonathan King, Eric Lander, Michael Levine, Susan Lindquist
2006	Anne-Claude Gavin, David Haussler, Ajay Royyuru, David Sankoff, Michael Waterman, Carl Zimmer, Roman Zubarev

Table 15. Distinguished lecture series in Computational Biology, Biology, and New Technologies

Year	Stanislaw Ulam Memorial Computational Biology Lecture	Distinguished Biology Lecture	Distinguished New Technologies Lecture
1997	Eric Lander	Rich Roberts	Robert Lipshutz
1998	Joshua Lederberg	Ron Davis	David Cox
1999	Pitor Slonimsky		Ed Southern
2000	Minoru Kanehisa	Walter Gilbert	Leroy Hood
2001	George Church	Philip Sharp	Mark Adams
2002	Craig Venter	David Ho	Harold Garner
2003	Edward Trifonov	Christiane Nüsslein-Volhard	Árpád Furka
2004	Russell Doolittle	Andrew Fire	Carlos Bustamante
2005	Charles DeLisi	Jonathan King	
2006	Michael Waterman	Anne-Claude Gavin	Roman Zubarev

The RECOMB Organizers

Since its inception in 1997, many scientists have participated in the conference in many fashions. While the committees have enjoyed the membership of over 170 different individuals between 1997 and 2006, many have participated over multiple years. The Steering Committee had consistent presence of 5 scientists between 1997 and 2005, including Michael Waterman, Pavel Pevzner, Ron Shamir, Sorin Istrail and Thomas Lengauer. The steering committee included 6 members throughout the first 8 years of the conference, with Richard Karp rounding out the group through 2003, and passing the position on to Terry Speed in 2004. In 2005 Michal Linial joined the Steering Committee to increase its size to 7.

Table 16. RECOMB Committee Membership. Each year shows the number of members in each committee.

Year	Number of Members		
	Steering	Organizing	Program
1997	6	5	23
1998	6	4	21
1999	6	6	29
2000	6	8	27
2001	6	9	23
2002	6	11	28
2003	6	5	31
2004	6	9	42
2005	7	17	43
2006	7	9	38

Table 17. RECOMB Program Committee Membership

Name	Years
Michael Waterman	10
Pavel Pevzner	10
Ron Shamir	10
Thomas Lengauer	10
Sorin Istrail	10
Martin Vingron	9
Richard Karp	9
Terry Speed	7
David Sankoff	6
Satoru Miyano	6
Gene Myers	5
Tandy Warnow	5
Dan Gusfield	5
Gordon Crippen	5
Sridhar Hannenhalli	5

The organizing committee has had a far more variable composition. Between 1997 and 2006, a total of 81 individuals have comprised the committee. The program committee has grown in size throughout the years of the conference (Table 16). While the size of the organizing and program committees do not correlate perfectly, the trend toward an increasing number of members per year has been exhibited in both. Numerous individuals have served on program committees in multiple years (Table 17).

RECOMB Funding

RECOMB has received support from a variety of sources. The US Department of Energy, US National Science Foundation and the SLOAN Foundation have been 3 major sponsors over the 10 years. Many other sponsors have significantly contributed to the conference, including IBM, International Society for Computational Biology (ISCB), SmithKline Beecham, Apple, Applied Biosystems, Celera, Compaq, Compugen, CRC Press, Glaxo-SmithKline, Hewlett-Packard, The MIT Press and the Broad Institute, Accelerys. Affymetrix, Agilent Technologies, Aventis, Berlin Center for Genome Based Bioinfornatics-BCB, Biogen, Boston University's Center for Advanced Genomic Technology, Centre de recherche en calcul applique (CERCA), CNRS, Conseil Regional Rhone-Alpes, Eurogentec-Seraing, Geneart GmbH, Genome Therapeutics, IMGT, INRA, LION Biosceince, LIPHA, Mairie de Lyon, Mathworks, Millennium Pharmaceuticals, Max Planck Institute for Molecular Genetics, Microsoft Research, NetApp, Novartis, Paracel, Partek Incorporated, Pfizer, Rosetta Biosoftware, Schering AG, Sun Microsystems, Technologiestiftung Berlin, The European Commission, High-level Scientific Conferences, The German Federal Ministry for Education and Research, The San Diego Supercomputer Center, The University of California-San Diego, Timelogic, Wyeth, Universitat degli Studi di Padova, Italy, DEI and AICA.

Conclusion

The approach of the 10th RECOMB conference held in Venice Italy provides us an opportunity to reflect on RECOMB's history. The landscape of computational biology has changed drastically since the first RECOMB Conference was held in Santa Fe, New Mexico. Today's conference contains papers covering research topics that did not exist 10 years ago. Over this period, many individuals have made significant research contributions through published papers. Many of the original founders of the RECOMB conference are still active, and many new faces are becoming active in the community each year.

Contents

Short Abstracts

RECOMB Retrospective

The Second Decade of the International Conference on Research in Computational Molecular Biology (RECOMB)

Farhad Hormozdiari[1(✉)], Fereydoun Hormozdiari[2], Carl Kingsford[3], Paul Medvedev[4], and Fabio Vandin[5]

[1] University of California at Los Angeles, Los Angeles, CA, USA
fhormoz@cs.ucla.edu
[2] University of California at Davis, Davis, CA, USA
[3] Carnegie Mellon University, Pittsburgh, PA, USA
[4] Pennsylvania State University, University Park, PA, USA
[5] University of Padova, Padova, Italy

Abstract. The year 2016 marks the 20th anniversary of the RECOMB (REsearch in Computational Molecular Biology) conference. On this occasion, we collect some facts and statistics about the conference's papers, authors, speakers, reviewers, and organizers. These data provide a succinct summary of the RECOMB conference over the last decade and can serve as a starting point for introspection and discussion about the future of the conference.

1 Introduction

Twenty years has passed since inception of the RECOMB (REsearch in Computational Molecular Biology) conference and during these years RECOMB has established itself as the flagship conference for computational biology. The twenty year anniversary of RECOMB has given us a chance to review the accomplishments of the community and reflect on strengths and weaknesses of the conference. Since a review was written in 2006 summarizing the first 10 years [1], this review focuses on the most recent 10 years of the conference. In the past ten years RECOMB has been held all over the world covering three continents and seven different countries as shown in Table 1. A total of 355 papers have been accepted and presented at the RECOMB conference covering diverse topics in computational biology. The highest number of papers accepted was for year 2011 with 43 accepted papers while the lowest number of papers accepted was for year 2012 with only 31 papers accepted.

These papers have advanced the field of computational biology, and introduced new computational methods. Some of these methods have been widely utilized for the analysis of high throughput genomic data. Some of the methods have contributed to high profile consortia projects including the ENCyclopedia of DNA Elements (ENCODE) project, The Cancer Genome Atlas (TCGA), and the 1000 Genomes project.

M. Singh (Ed.): RECOMB 2016, LNBI 9649, pp. 3–16, 2016.
DOI: 10.1007/978-3-319-31957-5_1

Table 1. RECOMB conference location and organizers in the past ten years

Year	Location	Dates	Program Chair	Conference Chair	# papers
2007	San Francisco, USA	April 21 – April 25	Terry Speed	Sandrine Dudoit	37
2008	Singapore	March 30 – April 2	Martin Vingron	Limsoon Wong	34
2009	Tucson, USA	May 18 – May 21	Serafim Batzglou	John Kececioglu	36
2010	Lisbon, Portugal	August 12 – August 15	Bonnie Berger	Arlindo Oliveira	36
2011	Vancouver, Canada	March 28 – March 28	Vineet Bafna	S. Cenk Sahinalp	43
2012	Barcelona, Spain	April 21 – April 21	Benny Chor	Rodric Guigo	31
2013	Beijing, China	April 7	Fengzhu Sun	Xuegong Zhang	32
2014	Pittsburgh, USA	April 2 – April 5	Roded Sharan	Panayiotis Benos	35
				Russell Schwartz	
2015	Warsaw, Poland	April 12 – April 15	Teresa Przytycka	Jerzy Tiuryn	36
				Bartek Wilczyński	
2016	Los Angeles, USA	April 17 – April 21	Mona Singh	Eleazar Eskin	35

2 The Authors and Presenters: The People that Have Made RECOMB a Success

2.1 The Authors

During these past ten years RECOMB has benefited from the contribution of many junior researchers in addition to well-established scientists. The most prolific authors — those with the largest number of RECOMB papers in the last 10 years — are shown in the Table 2. There were only four authors who published more than 1 paper on average over these 10 RECOMBs: Drs. Eleazar Eskin, Pavel Pevzner, Bonnie Berger and Benjamin J. Raphael from UCLA, UCSD, MIT and Brown University respectively. Some of the most prolific authors in the first ten years of RECOMB conference (from 1997 to 2006) have continued to publish many influential papers in RECOMB. However, there are many junior researchers that have been added to the list of prolific authors. This is a great testimony to the inclusiveness of the RECOMB conference.[1]

A co-authorship graph (Fig. 1) gives a sense of the collaborative nature of the RECOMB community. In this graph, nodes represent authors and edges are drawn between authors who have co-authored at least four RECOMB paper together in the last 10 years. The size of the node corresponds to the number of papers published. The width of the edges are proportional to the number of papers the two researchers co-authored, and the color of the node indicates the degree of the author: redder indicates a larger number of co-authors.

[1] Several authors have used variants of their names over the years. For all the analyses described in this paper, an effort was made to manually resolve slight name variations. However, it is possible that some errors remain in the analysis.

Table 2. Authors with the most papers in RECOMB over the last 10 years.

# papers	Author	# papers	Author
15	Eleazar Eskin	5	Richard M. Karp
13	Pavel A. Pevzner	5	Pei Zhou
12	Bonnie Berger	5	Nuno Bandeira
11	Benjamin J. Raphael	5	Nebojsa Jojic
9	Vineet Bafna	5	Bernard M. E. Moret
9	Eran Halperin	5	Alexander J. Hartemink
9	Bruce Randall Donald	4	Yu Lin
8	Fabio Vandin	4	William Stafford Noble
8	Cenk Sahinalp	4	Vladimir Jojic
7	Serafim Batzoglou	4	Sebastian Bcker
7	Roded Sharan	4	Ron Shamir
7	Jinbo Xu	4	Paul Medvedev
7	Chris Bailey-Kellogg	4	Noah Zaitlen
6	Sebastian Will	4	Ming Li
6	Rolf Backofen	4	Mathias Mhl
6	Niko Beerenwinkel	4	Louxin Zhang
6	Jianyang Zeng	4	Leen Stougie
6	Jian Peng	4	Jérôme Waldispühl
6	Eric P. Xing	4	Jian Ma
6	Carl Kingsford	4	Glenn Tesler
5	Ziv Bar-Joseph	4	Eli Upfal
5	Yun S. Song	4	Deniz Yörükoglu
5	Wing-Kin Sung	4	Christopher James Langmead
5	Teresa M. Przytycka	4	Buhm Han
5	Sivan Bercovici		

It is not surprising that the depicted graph consists of one big component and few smaller components. The big component includes a large portion of the RECOMB authors which are centered around few hubs. There are clear clusters inside this big component and it seems each of these clusters are mostly based on collaborations inside each institute and are centered around prolific authors. For instance you can see a clear clustering of authors from Carnegie Mellon University as an extended segment in this component. A similar pattern is observed for researchers from UCSD. In addition, the graphs shows frequent collaborations between researchers from UCLA and Tel Aviv University.

The pairs of authors who have the largest number of shared RECOMB publications are shown in Table 3. This table illustrates the productive collaboration between the members of the trios {Fabio Vandin, Benjamin J. Raphael,

Eli Upfal} working on systems biology/networks in cancer and {Rolf Backofen, Sebastian Will, Mathias Möhl}, with publications on RNA structure.

Table 3. Most frequent co-authorship pairs.

Author	Author	# papers
Benjamin J. Raphael	Fabio Vandin	6
Rolf Backofen	Sebastian Will	5
Bruce Randall Donald	Pei Zhou	5
Mathias Möhl	Sebastian Will	4
Mathias Möhl	Rolf Backofen	4
Buhm Han	Eleazar Eskin	4
Eli Upfal	Fabio Vandin	4
Benjamin J. Raphael	Eli Upfal	4

2.2 Keynote Speakers

The RECOMB conference has benefited from a line of high-profile keynote speakers. These include Nobel laureates Dr. Elizabeth H. Blackburn, Dr. Ada E. Yonath and Dr. Michael Levitt, in addition to Curt Stern Award recipients Dr. Patrick O. Brown, Dr. Leonid Kruglyak, and Dr. Evan E. Eichler. A list of keynote speakers is given in Table 4.

3 The Papers: It's All About the Science

Word clouds of the titles and abstracts (see Figs. 2 and 3) of the last 10 years of RECOMB papers help to illustrate the main topics of the conference. The expected words are revealed as frequent in these word clouds: "Sequence", "Protein", "Model", "Data", "Method", "Algorithm", "Gene", and so on — neatly summarizing the focus of the conference on method and algorithm development, recently particularly in genome and protein analysis.

We tracked how the top-twenty biologically meaningful words that appeared in the abstracts changed over time. Figure 4 shows the number of abstracts that contain these terms for each year of the last 10. The top five most common words used were "genome", "protein", "gene", "structure" and "complex", again showing the consistency of the focus of the conference at least at this broad level. We did not observe any major trends in the change in the relative frequency of the terms over the late 10 years.

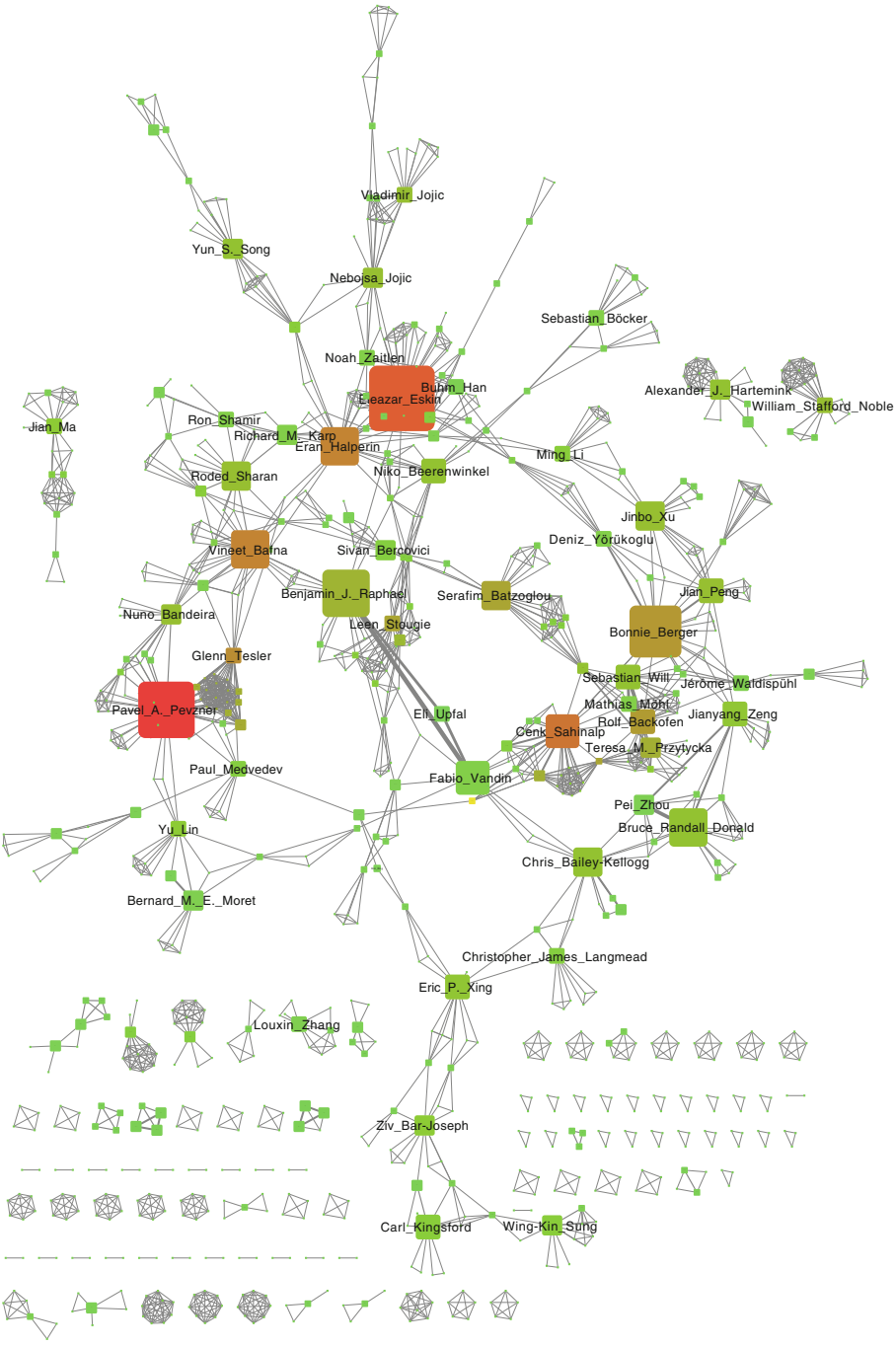


Fig. 1. Co-authorship graph.

Table 4. Keynote speakers

Year	Speakers
2007	Elizabeth H. Blackburn, Jennifer Marshall Graves, Yishi Jin, Jay D. Keasling, Harry F. Noller, Patrick O. Brown, Abby Dernburg, Aviv Regev
2008	Edison Liu, Sang Yup Lee, Andrei Lupas, Howard Cedar, Vivian Cheung, Temple F. Smith, Suzanne Cory
2009	Carlos D. Bustamante, Rade Drmanac, Mark Gerstein, Eran Halperin, Michael Hammer, Joanna Mountain, Stephen Quake, Mostafa Ronaghi, Pardis Sabeti, Michael Snyder
2010	Cecilia Arraiano, David Bartel, Isaac Kohane, Phil Kapranov, Norbert Perrimon Mona Singh
2011	Evan Eichler, Daphne Koller, Marco Marra, Karen Nelson, Elaine Mardis, Joseph Nadeau
2012	Richard Durbin, Eileen Furlong, Thomas Gingeras, Alfonso Valencia, Ada E. Yonath
2013	Scott Fraser, Takashi Gojobori, Deborah Nickerson, Nadia A. Rosenthal, Chung-I Wu, Sunny Xie
2014	Ian T. Baldwin, Atul Butte, James J. Collins, Trey Ideker, Tom Mitchell, Sarah A. Tishkoff
2015	M. Madan Babu, Bonnie Berger, M. Magda Konarska, Michael Levitt, Bas van Steensel, Wacaw Szybalski
2016	Karen Adelman, Phil Bradley, Peter S. Kim, Rob Knight, Leonid Kruglyak, Teresa Przytycka

Fig. 2. Word clouds showing the frequency of words in titles of accepted RECOMB papers since 2007.

3.1 Citations

Using Google Scholar, we compiled the number of citations of all the papers published in RECOMB in the past ten years. If a paper was published in both

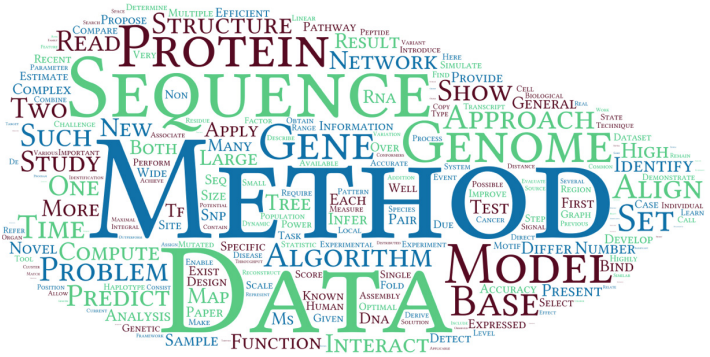


Fig. 3. Word clouds showing the frequency of words in abstracts of accepted RECOMB papers since 2007.

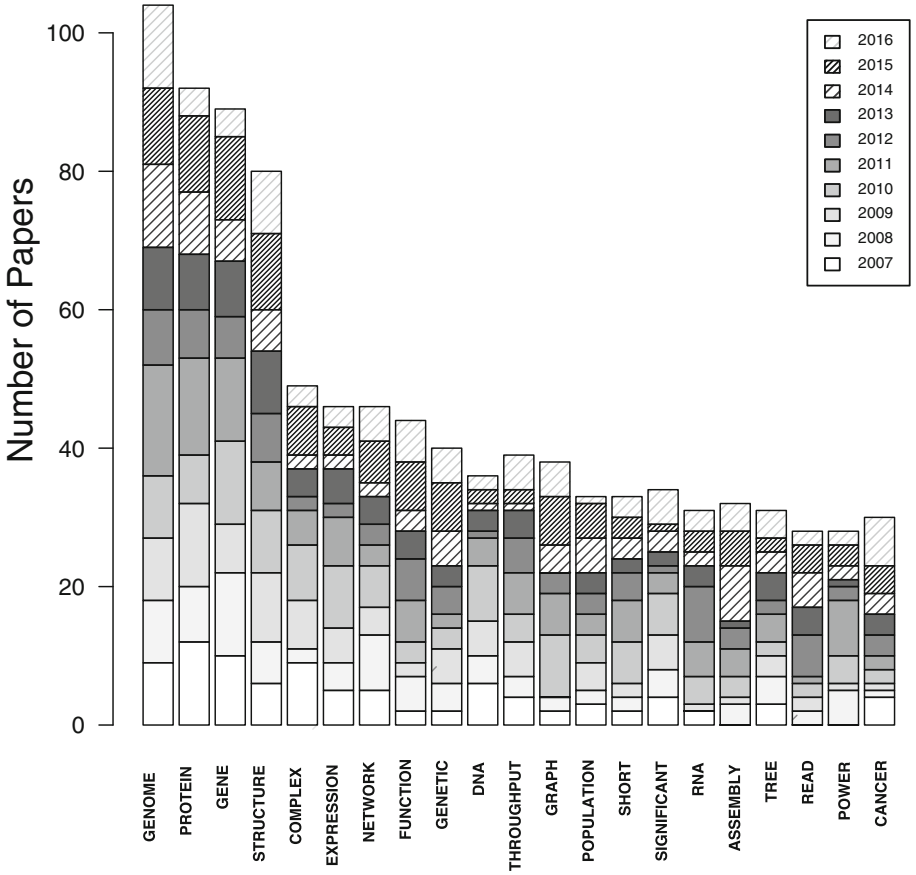


Fig. 4. Frequency of terms over the last 10 years.

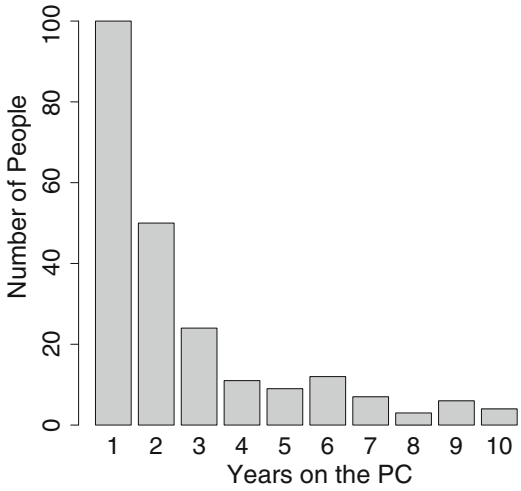


Fig. 5. Histogram of the number of years each PC member has served.

RECOMB and in a journal (e.g., JCB, Genome Research, etc.) we included the citations count for both maximum and the sum in the table.[2]

The papers with over seventy citations are shown in Table 5. Overall, 21 papers have a citation count over 70, indicating that RECOMB has published a number of influential and widely-read papers.

In addition, we calculated the average number of citations per year for each paper and the list of papers with more than 10 citation per year is shown in Table 6. It is interesting to note that the papers with high citation span many topics in computational biology. However, there is a clear bias for genomics/assembly papers and systems biology/networks papers to be among the top cited papers. We assigned the papers with over high citations into five groups of *Genomics, Systems Biology, RNA/Protein structure, Phylogeny/Evolution* and *Expression/Transcription*. Interestingly out of the 21 papers 16 (76 %) of them fall into two category of *Genomics* or *Systems biology*. This might be an indication that the number of researchers working in these fields is generally larger than the number of researchers working for instance in evolution or structure (RNA and protein).

4 The Organizers and Reviewers: People that Make It All Happen

The organization of the conference is a significant undertaking each year. It relies on the volunteered time of many individuals, most notably those on the program committee who provide careful reviews to inform the selection of the paper.

[2] If a paper's title changed significantly between the RECOMB version and the journal version, it is quite possible that additional citations were missed.

Table 5. Papers with at least 70 citations.

Paper authors and title	RECOMB year	Journal	Num. citations (max)	(sum) Num. citations
Rohit Singh, Jinbo Xu, Bonnie Berger. Pairwise global alignment of protein interaction networks by matching neighborhood topology	2007	PNAS	299	492
Nicholas D. Pattengale, Masoud Alipour, Olaf R. P. Bininda-Emonds, Bernard M. E. Moret, Alexandros Stamatakis. How many Bootstrap replicates are necessary?	2009	J Comput Biol	211	371
Fereydoun Hormozdiari, Can Alkan, Evan E. Eichler, Sleyman Cenk Sahinalp. Combinatorial algorithms for structural variation detection in high throughput sequenced genomes	2009	Genome Res	224	224
Fabio Vandin, Eli Upfal, Benjamin J. Raphael. De novo discovery of mutated driver pathways in cancer	2011	Genome Res	132	132
Jason Flannick, Antal F. Novak, Chuong B. Do, Balaji S. Srinivasan, Serafim Batzoglou. Automatic parameter learning for multiple network alignment	2008	J Comput Biol	77	130
Fabio Vandin, Eli Upfal, Benjamin J. Raphael. Algorithms for detecting significantly mutated pathways in cancer	2010	J Comput Biol	115	128
Joshua A. Grochow, Manolis Kellis. Network motif discovery using subgraph enumeration and symmetry-breaking	2007		121	121
Maxim Kalaev, Vineet Bafna, Roded Sharan. Fast and accurate alignment of multiple protein networks	2008	J Comput Biol	61	118
Banu Dost, Tomer Shlomi, Nitin Gupta, Eytan Ruppin, Vineet Bafna, Roded Sharan. QNet: a tool for querying protein interaction networks	2007	J Comput Biol	58	115
Song Gao, Niranjan Nagarajan, Wing-Kin Sung. Opera: reconstructing optimal genomic scaffolds with high-throughput paired-end sequences	2011	J Comput Biol	105	105
Osvaldo Zagordi, Lukas Geyrhofer, Volker Roth, Niko Beerenwinkel. Deep sequencing of a genetically heterogeneous sample: local haplotype reconstruction and read error correction	2009	J Comput Biol	85	95
Sourav Chatterji, Ichitaro Yamazaki, Zhaojun Bai, Jonathan A. Eisen. CompostBin: a DNA composition-based algorithm for binning environmental shotgun reads	2008		92	92
Yu Peng, Henry C.M. Leung, SM Yiu, Francis Chin. IDBA - a practical iterative de Bruijn graph de novo assembler	2010		92	92
Wei Li, Jianxing Feng, Tao Jiang. IsoLasso: a LASSO regression approach to RNA-Seq based transcriptome assembly	2011	J Comput Biol	87	87
Jonathan Laserson, Vladimir Jojic, Daphne Koller. Genovo: de novo assembly for metagenomics	2010	J Comput Biol	70	83
Tali Raveh-Sadka, Michal Levo, Eran Segal. Incorporating nucleosomes into thermodynamic models of transcription regulation	2009	Genome Res	82	82
Christos Kozanitis, Chris Saunders, Semyon Kruglyak, Vineet Bafna, George Varghese. Compressing genomic sequence fragments using SlimGene	2010	J Comput Biol	58	80
Chen Yanover, Ora Schueler-Furman, Yair Weiss. Minimizing and learning energy functions for side-chain prediction	2007	J Comput Biol	48	80
Yu-Wei Wu, Yuzhen Ye. A novel abundance-based algorithm for binning metagenomic sequences using l-tuples	2010	J Comput Biol	53	78
Sharon Bruckner, Falk Hffner, Richard M. Karp, Ron Shamir, Roded Sharan. Topology-free querying of protein interaction networks	2009	J Comput Biol	78	105
Eilon Sharon, Eran Segal. A feature-based approach to modeling protein-DNA interactions	2007	PLoS Comput Biol	73	73

Table 6. Papers with average number of citations per year ≥10.

Paper authors and title	RECOMB year	Journal	Avg. citations (max)	Avg. citations (sum)
Rohit Singh, Jinbo Xu, Bonnie Berger. Pairwise global alignment of protein interaction networks by matching neighborhood topology	2007	PNAS	33.2	54.6
Nicholas D. Pattengale, Masoud Alipour, Olaf R. P. Bininda-Emonds, Bernard M. E. Moret, Alexandros Stamatakis. How many Bootstrap replicates are necessary?	2009	J Comput Biol	30.1	53.0
Fereydoun Hormozdiari, Can Alkan, Evan E. Eichler, Sleyman Cenk Sahinalp. Combinatorial algorithms for structural variation detection in high throughput sequenced genomes	2009	Genome Res	32.0	32.0
Fabio Vandin, Eli Upfal, Benjamin J. Raphael. De novo discovery of mutated driver pathways in cancer	2011	Genome Res	26.4	26.4
Fabio Vandin, Eli Upfal, Benjamin J. Raphael. Algorithms for detecting significantly mutated pathways in cancer	2010	J Comput Biol	19.2	21.3
Song Gao, Niranjan Nagarajan, Wing-Kin Sung. Opera: reconstructing optimal genomic scaffolds with high-throughput paired-end sequences	2011	J Comput Biol	21	21
Layla Oesper, Ahmad Mahmoody, Benjamin J. Raphael. Inferring intra-tumor heterogeneity from high-throughput DNA sequencing data	2013	Genome Biol	19	20.6
Maxim Kalaev, Vineet Bafna, Roded Sharan. Fast and accurate alignment of multiple protein networks	2008	J Comput Biol	7.6	19.6
Wei Li, Jianxing Feng, Tao Jiang. IsoLasso: a LASSO regression approach to RNA-Seq based transcriptome assembly	2011	J Comput Biol	17.4	17.4
Jason Flannick, Antal F. Novak, Chuong B. Do, Balaji S. Srinivasan, Serafim Batzoglou. Automatic parameter learning for multiple network alignment	2008	J Comput Biol	9.6	16.2
Melissa Gymrek, David Golan, Saharon Rosset, Yaniv Erlich. lobSTR: a short tandem repeat profiler for personal genomes	2012	Genome Res	16.3	16.3
Yu Peng, Henry C.M. Leung, SM Yiu, Francis Chin. IDBA - a practical iterative de Bruijn Graph de novo assembler	2010		15.3	15.3
Jonathan Laserson, Vladimir Jojic, Daphne Koller. Genovo: de novo assembly for metagenomics	2010	J Comput Biol	11.6	13.8
Osvaldo Zagordi, Lukas Geyrhofer, Volker Roth, Niko Beerenwinkel. Deep sequencing of a genetically heterogeneous sample: local haplotype reconstruction and read error correction	2009	J Comput Biol	12.1	13.5
Joshua A. Grochow, Manolis Kellis. Network motif discovery using subgraph enumeration and symmetry-breaking	2007		13.4	13.4
Christos Kozanitis, Chris Saunders, Semyon Kruglyak, Vineet Bafna, George Varghese. Compressing genomic sequence fragments using SlimGene	2010	J Comput Biol	9.6	13.3
Leonid Chindelevitch, Daniel Ziemek, Ahmed Enayetallah, Ranjit Randhawa, Ben Sidders, Christoph Brockel, Enoch S. Huang. Causal reasoning on biological networks: interpreting transcriptional changes	2012	Bioinformatics	12	13
Yu-Wei Wu, Yuzhen Ye. A novel abundance-based algorithm for binning metagenomic sequences using l-tuples	2010	J Comput Biol	8.3	13
Banu Dost, Tomer Shlomi, Nitin Gupta, Eytan Ruppin, Vineet Bafna, Roded Sharan. QNet: a tool for querying protein interaction networks	2007	J Comput Biol	6.4	12.7
Tali Raveh-Sadka, Michal Levo, Eran Segal. Incorporating nucleosomes into thermodynamic models of transcription regulation	2009	Genome Res	11.7	11.7
Sourav Chatterji, Ichitaro Yamazaki, Zhaojun Bai, Jonathan A. Eisen. CompostBin: a DNA composition-based algorithm for binning environmental shotgun reads	2008		11.5	11.5
Sharon Bruckner, Falk Hffner, Richard M. Karp, Ron Shamir, Roded Sharan. Topology-free querying of protein interaction networks	2009	J Comput Biol	10.6	10.6
Osvaldo Zagordi, Armin Tpfer, Sandhya Prabhakaran, Volker Roth, Eran Halperin, Niko Beerenwinkel. Probabilistic inference of viral quasispecies subject to recombination	2012		6.5	10.5
Salim A. Chowdhury, Rod K. Nibbe, Mark R. Chance, Mehmet Koyuturk. Subnetwork state functions define dysregulated subnetworks in cancer	2010	J Comput Biol	9	10.3
Y. William Yu, Deniz Yorukoglu, Jian Peng, Bonnie Berger. Quality score compression improves downstream genotyping accuracy	2014	Nature Biotech	6	10

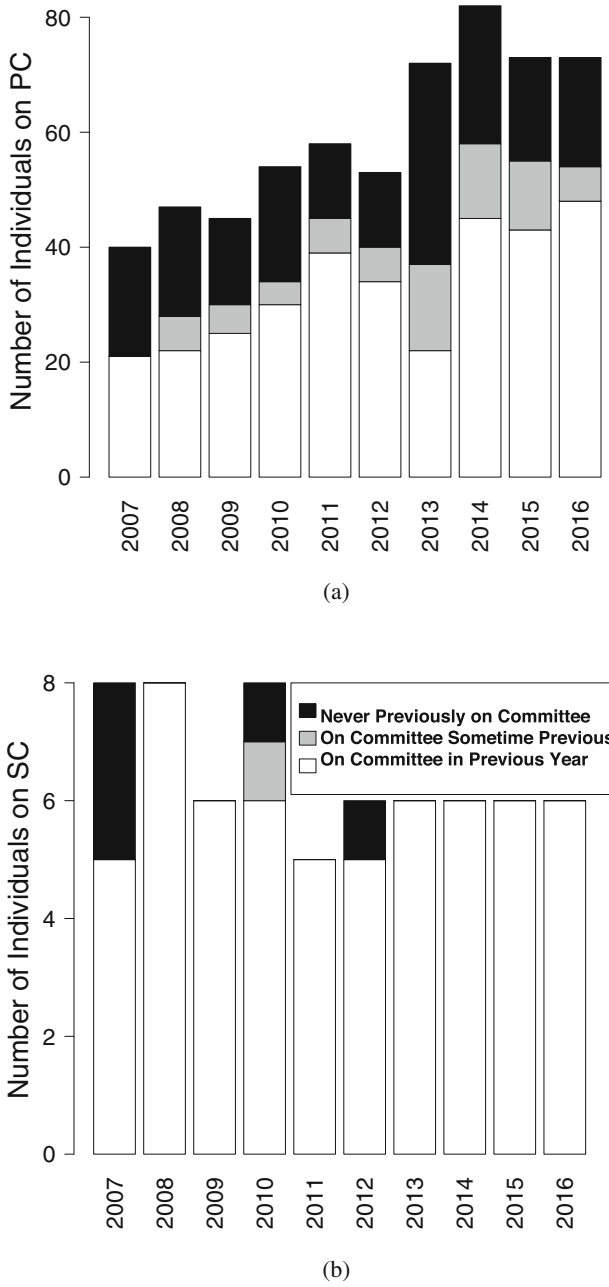


Fig. 6. Categorizing the PC (Program Committee) and SC (Steering Committee) to "Never Previously on Committee", "On Committee in Previous Year", and "On Committee Sometime Previously". Panel (a) and (b) illustrate this categorization for PC and SC, respectively.

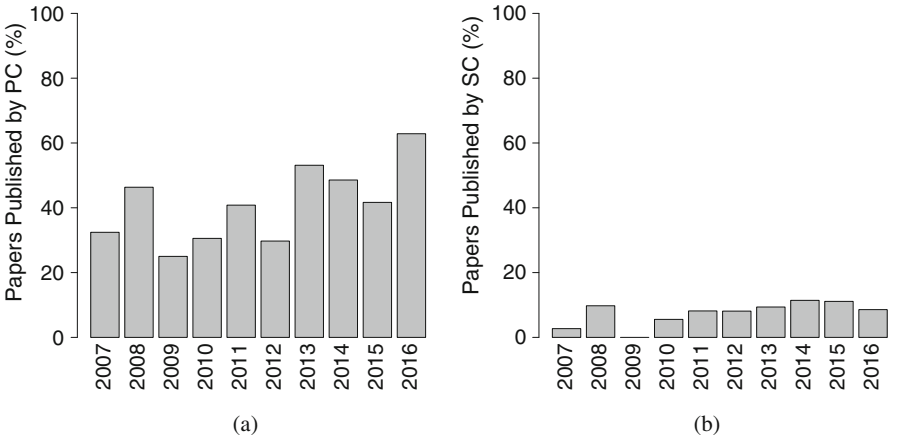


Fig. 7. Panel (a) and Panel (b) illustrate a histogram of the percentage of papers published each year from PC and SC members, respectively. We observe that over the years, anywhere from 25 % to 63 % of papers that are published in RECOMB are contributed by the PC members. We observe that 2 % to 11 % of papers that are published in RECOMB are contributed by the SC members.

Figure 5 gives a histogram of the number of people serving for various numbers of years. Many people have served on the RECOMB program committee for only one year of the last 10. A few people have served nearly every year since 2007. They are: Marie-France Sagot, Russell Schwartz, Thomas Lengauer (each served 8 years); Bonnie Berger, Knut Reinert, Michal Linial, Satoru Miyano, Sorin Istrail, William Noble (each serving 9 years); Jens Lagergren, Martin Vingron, Tatsuya Akutsu, Mona Singh (each serving all 10 years).

The PC has ranged in size between 40 people (2007) and 82 people (2014). Figure 6(a) shows the consistency of the PC over time. We classified each PC member into one of three categories: the PC member has never previously been on the PC of RECOMB, the PC member was on the PC the previous year and the PC member was not on the PC the previous year but had been on the PC sometime previously. As shown in the figure, there is a consistent core of PC members throughout the years. Figure 6(b) provides the same analysis for the SC.

On average over the last 5 years, 30 % of the PC members were newcomers each year, meaning that they had not served between 2007 and that year (see Fig. 6). On average each year 45 % of the program committee had not served in the year previous. These numbers indicate that while there is a strong core of consistent program committee members, the program committee is fairly dynamic and changing year-to-year.

Figure 7(a) illustrates a histogram for the fraction of the papers published each year from the PC members. We observe that over the years, anywhere from 25 % to 63 % of papers that are published in RECOMB are contributed by

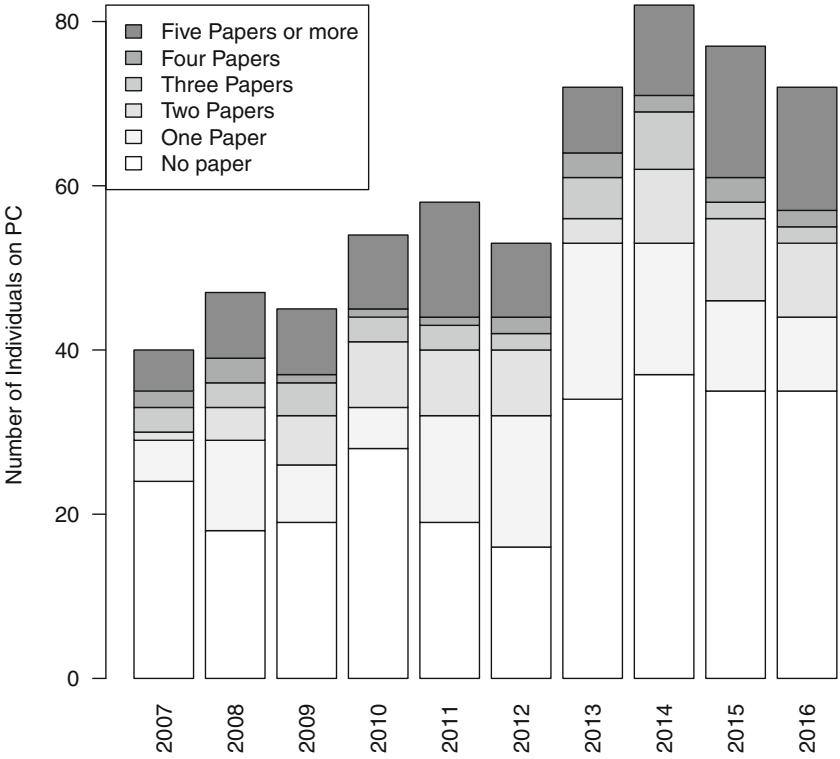


Fig. 8. Categorization of the PC members in the last 10 years (2007–2016) based on the number of papers published.

the PC members. We conduct the same experiment for the steering committee (SC). We observe that 2 % to 11 % of papers that are published in RECOMB are contributed by the SC members (see Fig. 7(b)).

We categorize the PC members for each year in the last 10 years based on the number of papers published. We categorize them into 6 categories which are as follow: PC members who have no paper published in the last 10 years, PC members who have published one paper in the last 10 year, PC members who have published two papers in the last 10 year, PC members who have published three paper in the last 10 year, PC members who have published four paper in the last 10 year, and PC members who have published five and more paper in the last 10 year. Figure 8 illustrates this categorization of the PC members.

Abstract. Molecular simulation techniques are widely applicable to other leading anomaly-detection methods, and yield a higher proportion of predictions with standard methods and other resources, to even further improve accuracy. Third, transcript quantification is often not known. To combat this problem, which can actually be measured from a recently proposed approach and demonstrate that this limits the usefulness of our quasispecies assembly method and provide efficient algorithms for STR profiling. We validated our algorithm is a spatial arrangement of physicochemical features in a biological sample. Due to size and scope. In studying the strength and specificity of the molecules of special interest as a vaccine, it is never possible to know/measure the precise binding positions of the most intriguing and challenging topic in structure-based computational protein design algorithms to evaluate the probability of observing a high discrimination ability. Unfortunately, however, BayesCall is too computationally expensive to date (a divide-and-conquer approach), showing that combining pairs of reads that map to multiple loci as a population of recipients. We refer to those facets of the local alignment problem– identifying many known features of the input ligands, which may have divergent sequences but similar folds. To implement this, we give a probabilistic model-based approach to identify the transcript isoforms from which one solves a theoretically hard but practically tractable optimization problem on each chromosome). Modern bulk genome sequencing mixes the signals of tumour clones frequently differ with respect to the problem is known to be isolated. However, biological processes are highly functionally coherent.

Fig. 9. Abstract generated from a 2nd-order Markov model trained on RECOMB abstracts from 2007–2015.

5 Conclusion

RECOMB remains a central and important part of the computational biology community. It has produced a number of highly cited and influential papers, and still serves as the premier venue for algorithmic work directed at biological problems. A great many people have worked to make this happen, only some of which it was possible to mention by name in this short article.

Although we can not predict which direction RECOMB will go in the next 10 years, we can use the current data to make some predictions. Figure 9 gives a simulated RECOMB abstract generated randomly from a 2nd-order Markov model trained on the RECOMB abstracts of between 2007–2015. While clearly the goal of completely automating biological discovery has a long way to go, RECOMB over the last 10 years has moved the community significantly further along toward that goal, and hopefully that progress will continue with RECOMB at its center for many more years.

Reference

1. Aerni, S.J., Eskin, E.: 10 years of the international conference on research in computational molecular biology (RECOMB). In: Apostolico, A., Guerra, C., Istrail, S., Pevzner, P.A., Waterman, M. (eds.) RECOMB 2006. LNCS (LNBI), vol. 3909, pp. 546–562. Springer, Heidelberg (2006)

Extended Abstracts

A MAD-Bayes Algorithm for State-Space Inference and Clustering with Application to Querying Large Collections of ChIP-Seq Data Sets

Chandler Zuo, Kailei Chen, and Sündüz Keleş(✉)

Department of Statistics, Department of Biostatistics and Medical Informatics, University of Wisconsin-Madison, Madison, WI, USA
keles@stat.wisc.edu

Abstract. Current analytic approaches for querying large collections of chromatin immunoprecipitation followed by sequencing (ChIP-seq) data from multiple cell types rely on individual analysis of each dataset (i.e., peak calling) independently. This approach discards the fact that functional elements are frequently shared among related cell types and leads to overestimation of the extent of divergence between different ChIP-seq samples. Methods geared towards multi-sample investigations have limited applicability in settings that aim to integrate 100s to 1000s of ChIP-seq datasets for query loci (e.g., thousands of genomic loci with a specific binding site). Recently, [1] developed a hierarchical framework for state-space matrix inference and clustering, named MBASIC, to enable joint analysis of user-specified loci across multiple ChIP-seq datasets. Although this versatile framework both estimates the underlying state-space (e.g., bound vs. unbound) and also groups loci with similar patterns together, its Expectation-Maximization based estimation structure hinders its applicability with large numbers of loci and samples. We address this limitation by developing a MAP-based Asymptotic Derivations from Bayes (MAD-Bayes) framework for MBASIC. This results in a K-means-like optimization algorithm which converges rapidly and hence enables exploring multiple initialization schemes and flexibility in tuning. Comparisons with MBASIC indicates that this speed comes at a relatively insignificant loss in estimation accuracy. Although MAD-Bayes MBASIC is specifically designed for the analysis of user-specified loci, it is able to capture overall patterns of histone marks from multiple ChIP-seq datasets similar to those identified by genome-wide segmentation methods such as ChromHMM and Spectacle.

Keywords: Small-variance asymptotics · MAD-Bayes · Unified state-space inference and clustering · ChIP-seq

1 Introduction

Many large consortia (e.g., ENCODE [2], REMC [3]) as well as investigator-initiated projects generated large collections of ChIP-seq data profiling multiple

M. Singh (Ed.): RECOMB 2016, LNBI 9649, pp. 19–36, 2016.
DOI: 10.1007/978-3-319-31957-5_2

proteins and histone modifications across a wide variety of systems. Most current approaches for analyzing data from multiple cell types perform initial analyses such as peak calling in ChIP-seq independently in each cell/tissue/condition type. This approach ignores the fact that functional elements are frequently shared between related cell types, and leads to an over estimation of the extent of functional divergence between the conditions. Although the uniform processing pipelines developed by data-generating consortia and the resulting analysis of consortia data enable easy access to these data, joint analysis approaches that take advantage of the inherent relationships between datasets and cell types are required. Joint inference for ChIP-seq datasets can be formulated as inferring for each locus whether or not it exhibits ChIP-seq signal in a given condition and also grouping loci based on their profile similarity across multiple samples.

It is now widely accepted that joint analysis of these types of data can uncover signals that are otherwise too small to detect from a single experiment [4,5]. Among the available joint analysis methods, jMOSAiCS [6] builds on ChIP-seq peak-caller MOSAiCS [7] and incorporates a multi-layer hidden states model that governs the relationship of enrichment among different samples. [8] utilizes a one-dimensional Markov random field (MRF) model to account for spatial dependencies along the genome while modeling individual components by mixtures of Zero Inflated Poisson or Negative Binomial models. dCaP [9] uses a three-step log-likelihood ratio test to jointly identify binding events in multiple experimental conditions. ChromHMM [10] and Segway [11] are two commonly adopted approaches for segmenting the genome into chromatin states based on histone ChIP-seq and rely on hidden Markov models and Bayesian Networks, respectively. Recently, Spectacle [12] provided a transformative improvement of ChromHMM by utilizing spectral learning for parameter estimation in HMMs. hiHMM [13] uses a Bayesian non-parametric formulation of the HMMs while taking into account species-specific biases.

Overall, available strategies for considering multiple ChIP-seq datasets simultaneously can be broadly classified based on (i) whether or not they can deal with only TFs [14,15], only histone modifications [10–12,16,17], or both [5,6] types of ChIP-seq data; (ii) whether or not they rely on a priori analysis of individual datasets [10,12,14,15,17], (iii) whether or not they focus on differential occupancy and can handle very few numbers of conditions [14,18,19], (iv) whether or not they can scale up to 100s to 1000s of datasets. These approaches, with the potential exception of [12], do not scale up to 100s to 1000s of datasets since they, to a large extent, utilize variants of hidden Markov models and/or implement variants of the Expectation-Maximization (EM) algorithm [20] for parameter estimation. Furthermore, none of these approaches accommodate querying of multiple datasets for *selected* loci. Their analysis results serve to "annotate" user-specified loci without any notion of uncertainty.

We recently introduced MBASIC [1] as a probabilistic method for querying multiple ChIP-seq datasets jointly for user-specified loci. When multiple ChIP-seq datasets (multiple TFs profiled in different cell/tissue types under a variety of conditions) are available, the key inference encompasses both identifying peaks

in individual datasets (*state-space mapping*) as well as identifying groups of loci that cluster across different experiments (*state-space clustering*). At the core of MBASIC are biologically validated and commonly adapted models for measurements from individual experiments (e.g., read data models from [7,21] for state-space mapping) and a mixture model for clustering of the loci with similar state-space mapping. Parameter estimation in this versatile model is based on the EM algorithm and hence does not scale up with large numbers of user-specified loci and ChIP-seq datasets. In this paper, we adopt a small-variance asymptotics framework for MBASIC and derive a K-means-like MAD-Bayes algorithm [22]. This alternative estimation framework for MBASIC targets large-scale datasets and genomic loci. Specifically, we consider a mixture of Log-normal distributions for state-specific observations with a Chinese Restaurant Process (CRP) [23,24] as the clustering prior. Small-variance asymptotics for maximizing the posterior distribution leads to a K-means like objective function with a key penalty term for the number of clusters. Extensive comparisons with MBASIC indicate that this approach can significantly speed up model estimation without significant impact on the estimation performance. Although methods like ChromHMM and Spectacle inherently have a different purpose than MAD-Bayes MBASIC, we compared the three on histone ChIP-seq data from GM12878 cells. This comparison indicated that MAD-Bayes MBASIC can capture the overall patterns that these segmentation methods identify.

2 Method

We begin our exposition with an overall description of the Bayesian MBASIC model (Fig. 1) and then derive the MAD-Bayes algorithm. Some key aspects of our approach are model initialization and tuning parameter selection. Although these aspects arise in all of the above mentioned joint analysis methods, they are typically not well studied because of the computational costs.

2.1 The Bayesian MBASIC Model

We consider I genomic loci of interest, indexed from $i = 1, \cdots, I$, from the reference genome with observations from K different experimental conditions. We use the notion of loci loosely in the sense that these loci could correspond to promoter regions of genes (all or members of specific pathways), locations of genome with a specific transcription factor (TF) binding motif, or peaks from a specific ChIP-seq experiment. The K conditions denote different TFs and cell/tissue types. Then, the key inference concerns analyzing I loci based on these K experiments. To further motivate the circumstances this inference problem arises, we consider an example from GATA-factor biology. In [25], we were interested in an overall analysis of all the E-box-GATA composite elements based on all the ENCODE ChIP-seq data to identify sites similar to the functional E-box-GATA composite element at the +9.5 loci which is causal for MonoMAC disease (a rare genetic disorder associated with myelodysplasia, cytogenetic abnormalities,

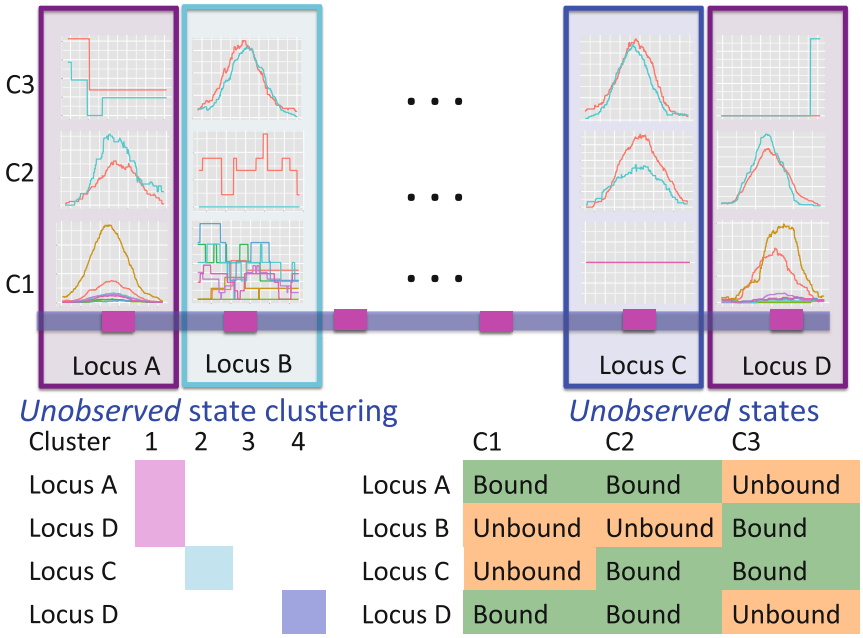


Fig. 1. Overview of the MBASIC modeling framework. Curves within each panel depict different replicates under the experimental conditions C1, C2, and C3. Loci A and D are in the same cluster.

and myeloid leukemias) [26]. The E-box-GATA composite elements are represented by CANNTGN{6-14}AGATAA oligonucleotides, where N denotes any nucleotide and N{6-14} denotes any nucleotide sequence of length 6 to 14 bps and are found abundantly in the genome, e.g., hg19 harbors ∼102 K of them. Joint analysis of these loci over, for example, all the available ENCODE TF ChIP-seq datasets (∼880 based on https://www.encodeproject.org) to identify groups of loci that are similar to the +9.5 element represents one potential application. In the MBASIC framework, the binding states are governed by a clustering structure, which groups genomic loci with similar overall binding states across experiments together. For the E-box-GATA composite elements example, in addition to the binding states for each candidate loci across experiments, MBASIC also reports a clustering of loci based on the binding states. The cluster with the +9.5 loci harbors candidate E-box-GATA elements to follow up [25].

Let n_k denote the number of experimental replicates for the k-th condition. We denote the observation for the i-th locus under condition k for the l-th replicate by Y_{ikl}, for $1 \leq i \leq I$, $1 \leq k \leq K$, and $1 \leq l \leq n_k$. We assume that a latent state is associated with the i-th locus and the k-th condition. θ_{iks} is the indicator for the state to be s, where s takes values in a discrete state-space $\{1, \cdots, S\}$. In a ChIP-seq experiment, we typically have $S = \{1, 2\}$, where $\theta_{ik1} = 1$ or $\theta_{ik2} = 1$ indicates that the i-th locus is unenriched (unbound) or

enriched (bound) under condition k, respectively. Our model consists of two key components. The first component, *state-space mapping*, assumes the following distribution of Y_{ikl} conditional on θ_{ik}:

$$(Y_{ikl}|\theta_{iks}=1) \overset{\text{i.i.d.}}{\sim} f_s(\cdot|\mu_{kls},\sigma_{kls},\gamma_{ikls}),$$

where f_s is a density function with parameters μ_{kls}, σ_{kls}, and γ_{ikls} denotes covariates encoding known information for locus i. Note that γ_{ikls} carries information related to how the counts for unenriched loci arise (when $\theta_{ik}=0$), i.e., data from control Input experiments, GC content, and mappability [21]. In this paper, we take f_s to be Log-normal distribution to represent ChIP-seq read counts after potential normalization for mappability and GC content:

$$(\log(Y_{ikl}+1)|\theta_{iks}=1) \overset{\text{i.i.d.}}{\sim} N(\mu_{kls}\gamma_{ikls},\sigma^2_{kls}), \tag{1}$$

where we utilize conjugate priors $\mu_{kls} \sim N(\xi,\tau^2)$ and $\sigma^2_{kls} \sim Gamma(\omega,\nu)$.

The second part of the Bayesian MBASIC model is *state-space clustering*. We assume that the loci can be clustered into J groups denoted by $C_1,\cdots,C_J$, i.e., $\{1,2,\cdots,I\}=C_1\cup\cdots\cup C_J$. Let $z_{ij}=1$ if the i-th locus belongs to cluster j and 0 otherwise. The states for the loci within the same cluster follow a product multinomial distribution:

$$(\theta_{iks})_{s=1}^{S}|z_{ij}=1 \overset{\text{i.i.d.}}{\sim} \text{Multinomial}\left(1,(w_{jks})_{1\le s\le S}\right), \quad \sum_{s=1}^{S} w_{jks}=1, \tag{2}$$

with non-informative prior $(w_{jks})_{1\le s\le S} \sim \text{Dir}(1,\ 1,\cdots,\ 1)$. We further assume a Chinese Restaurant Process [24] as a prior for the number of clusters J. Let α be a hyper-parameter of the model. The first locus forms C_1 at the start and each locus gets assigned to a cluster recursively. Suppose we have assigned loci $1,\ \cdots,\ i-1$ to J' clusters. The i-th locus is then assigned to $C_{j'},\ j'\le J'$ with probability proportional to the size of $C_{j'}$. It can also form a new cluster $C_{j'+1}$ with probability proportional to α. Then, the prior density for a partition with J clusters is

$$f(z_{ij},i=1,\cdots,I,j=1,\cdots,J)=\alpha^{J-1}\frac{\Gamma(\alpha+1)}{\Gamma(\alpha+I)}\prod_{j=1}^{J}\left(\sum_{i=1}^{I} z_{ij}-1\right)!. \tag{3}$$

With these specifications, we can derive the posterior density of the model for parameter estimation. Although the resulting posterior density leads to a Gibbs sampling algorithm, such a Gibbs sampling scheme requires excessive computational time for mixing (data not shown). Therefore, we derive MAD-Bayes algorithm by utilizing small-variance asymptotics.

2.2 MAD-Bayes Algorithm

We further make the following small-variance assumptions for the MBASIC model:

Assumption 1. *All data sets have equal variance:* $\sigma_{kls}^2 = \sigma^2 \to 0$.

Assumption 2. *For a given cluster and condition, one of the hidden states dominates with* $w_{jks} \in \{1-(S-1)e^{-\lambda_w/\sigma^2}, e^{-\lambda_w/\sigma^2}\}$ *for* $\lambda_w > 0$.

Assumption 3. $\alpha = e^{-\lambda_w \lambda_r / 2\sigma^2} \overset{\sigma^2 \to 0}{\longrightarrow} 0$ *for* $\lambda_w, \lambda_r > 0$.

Proposition 1. *Under 1, 2, 3, and as* $\sigma^2 \to 0$, *the posterior density reduces to*

$$\begin{aligned} & -2\sigma^2 \log \mathbb{P}(\theta, z, \mu, \sigma, w, J|Y) \\ =& \sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{l=1}^{n_k}\sum_{s=1}^{S} \theta_{iks}[\log(y_{ikl}+1) - \mu_{kls}\gamma_{ikls}]^2 \\ +& \lambda_w \sum_{i=1}^{I}\sum_{j=1}^{J} z_{ij} \left[\sum_{k=1}^{K}\sum_{s=1}^{S}(\theta_{iks} - w_{jks})^2\right] + \lambda_w\lambda_r(J-1) + \text{Constant} + o(1). \end{aligned} \tag{4}$$

This proposition implies that the MAP estimate of the MBASIC framework with CRP and Log-normal mixture model is asymptotically equivalent to the solution of the following optimization problem:

$$\begin{aligned} \min_{\mu,z,\theta,w,J} & \sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{l=1}^{n_k}\sum_{s=1}^{S} \theta_{iks}[\log(y_{ikl}+1) - \mu_{kls}\gamma_{ikls}]^2 \\ & + \lambda_w \sum_{i=1}^{I}\sum_{j=1}^{J} z_{ij} \left[\sum_{k=1}^{K}\sum_{s=1}^{S}(\theta_{iks} - w_{jks})^2\right] + \lambda_w\lambda_r(J-1), \end{aligned} \tag{5}$$

where the objective function can be viewed as a weighted loss function that integrates the state inference error from Log-normal density as the first term, the clustering error as the second term, and the cost for creating new clusters as the third term. Here, $\lambda_w > 0$ and $\lambda_r > 0$ are tuning parameters that ensure that the cluster assignments are non-trivial. The equal variance assumption is inherently quite strong for ChIP-seq data; however, it was recently shown to work well as a first approximation in a differential ChIP-seq analysis context [19]. We next derive the MAD-Bayes algorithm to generate a local solution for this minimization problem (Algorithm 1).

We note that each step of this algorithm does not increase the objective function in Eq. (5), and the updates for w_{jks}'s and μ_{kls}'s minimize the objective function for a fixed configuration of θ_{iks}'s and z_{ij}'s. Moreover, there are finite number of combinations for θ_{iks}'s and z_{ij}'s such that no cluster is empty and all clusters are distinct from one another. With such observations, we conclude the convergence of this algorithm.

Proposition 2. *Algorithm 1 converges after a finite number of iterations to a local minimum of the objective function in Eq. (5).*

Algorithm: The MAD-Bayes algorithm for the Bayesian MBASIC model.

repeat

1. Update the cluster labels z_{ij}'s. For each $i = 1, \cdots, I$, compute the distance between locus i and each existing cluster $j = 1, \cdots, J$ as:

$$t_j = \sum_{k=1}^{K} \sum_{s=1}^{S} (\theta_{iks} - w_{jks})^2$$

   and find the minimal $j_0 = \arg\min t_j$. If $t_{j_0} < \lambda_r$, assign $z_{ij_0} = 1$. Otherwise,generate a new cluster $J + 1$ with a single locus i.
2. Assign the states θ_{iks}'s. For $i = 1, \cdots, I$, $k = 1, \cdots, K$, and $s = 1, \cdots, S$, let

$$s_0 \leftarrow \arg\min_s \sum_{l=1}^{n_k} [\log(y_{ikl} + 1) - \mu_{kls}\gamma_{ikls}]^2 + \lambda_w \sum_{j=1}^{J} z_{ij} \left[(1 - w_{jks})^2 + \sum_{s' \neq s} w_{jks'}^2 \right]$$

   and let $\theta_{iks_0} = 1$, $\theta_{iks} = 0$ for $s \neq s_0$.
3. Update the Log-normal mean parameters μ_{kls}'s. For $k = 1, \cdots, K$, $l = 1, \cdots, n_k$, and $s = 1, \cdots, S$,

$$\mu_{kls} \leftarrow \frac{\sum_{i=1}^{I} \theta_{iks} \log(y_{ikl} + 1)\gamma_{ikls}}{\sum_{i=1}^{I} \theta_{iks}\gamma_{ikls}}.$$

4. Update the Multinomial parameters w_{jks}'s. For $j = 1, \cdots, J$, $k = 1, \cdots, K$, and $s = 1, \cdots, S$,

$$w_{jks} \leftarrow \frac{\sum_{i=1}^{I} z_{ij}\theta_{iks}}{\sum_{i=1}^{I} z_{ij}}.$$

until *Convergence*;

Algorithm 1. The MAD-Bayes algorithm for the Bayesian MBASIC model.

2.3 Model Initialization

Similar to the EM algorithm variants for HMMs, the MAD-Bayes algorithm for MBASIC also converges to a local solution and hence can be sensitive to initial starting values. We present a guided two-stage initialization strategy for the states and clusters to attenuate the impact of initialization. We start from initialization of the states by minimizing the state inference error (the first term in Eq. (5)), which has a degenerate form if $\lambda_w = 0$:

$$\min_{\mu,\theta} \sum_{i=1}^{I} \sum_{k=1}^{K} \sum_{l=1}^{n_k} \sum_{s=1}^{S} \theta_{iks} [\log(y_{ikl} + 1) - \mu_{kls}\gamma_{ikls}]^2. \tag{6}$$

Therefore, we use Algorithm 1 by setting $\lambda_w = 0$ to initialize θ_{iks}'s and μ_{kls}'s.

We utilize these initial values of θ_{iks}'s and consider three options for the cluster initialization (i.e., z_{ij}'s and w_{iks}'s): K-means, K-means++, and Adaptive K-means++, where the first two require a pre-determined number of clusters J which we discuss in Sect. 2.4. The K-means option runs hard K-means algorithm on the θ_{iks}'s; while the K-means++ option assigns a cluster label to each unit i with probability inversely proportional to its distance to the current clusters $d_i = \sum_{j=1}^{J} z_{ij} \sum_{k=1}^{K} \sum_{s=1}^{S} (\theta_{iks} - w_{jks})^2$. The adaptive K-means initialization uses a K-means++ style, but increases the number of clusters from $J = 1$, until the value of the function in Eq. (7) does not decrease.

2.4 Selecting the Tuning Parameters

We note that the CRP prior for the number of clusters and the small-variance asymptotics assumptions introduce tuning parameters for the MAD-Bayes algorithm (Algorithm 1). Even for the models with one tuning parameter, [22] acknowledged the difficulty in choosing their appropriate values in practice. Hence, we propose an empirically-motivated method for tuning parameter selection. In practice, we don't expect our small-variance assumption $e^{-\lambda_w/\sigma^2} \to 0$ as $\sigma^2 \to 0$ to hold rigidly for real data; however, we expect $e^{-\lambda_w/\sigma^2}$ to be small since it represents the prior probability of enrichment. To maintain the relative small value of $e^{-\lambda_w/\sigma^2}$, we set λ_w as $2\hat{\sigma}^2$ with $\hat{\sigma}^2$ obtained by optimization of the first term in Eq. (5):

$$\hat{\sigma}^2 = \min_{\mu,\theta} \sum_{i=1}^{I} \sum_{k=1}^{K} \sum_{l=1}^{n_k} \sum_{s=1}^{S} \theta_{iks} [\log(y_{ikl} + 1) - \mu_{kls}\gamma_{ikls}]^2.$$

Our computational experiments (data not shown) indicate that varying λ_w in the order of $\hat{\sigma}^2$ does not impact model estimation. The λ_r parameter mediates between the clustering error and the cost of the number of clusters for fixed λ_w. We choose a set of candidate λ_r values by considering the conjugacy between λ_r and J. Suppose J is a global minimum of the objective function in Eq. (5), then fixing the θ_{iks}'s, λ_w, λ_r, J minimizes

$$\sum_{i=1}^{I} \sum_{j=1}^{J'} z_{ij} \left[\sum_{k=1}^{K} \sum_{s=1}^{S} (\theta_{iks} - w_{jks})^2 \right] + \lambda_r (J - 1). \tag{7}$$

Therefore, we let

$$L(J') = \min_{z,w} \left\{ \sum_{i=1}^{I} \sum_{j=1}^{J'} z_{ij} \left[\sum_{k=1}^{K} \sum_{s=1}^{S} (\theta_{iks} - w_{jks})^2 \right] \right\},$$

with $L(J) - L(J+1) \leq \lambda_r \leq L(J-1) - L(J)$ (Fig. 4). Algorithm 2 applies this idea to choose a list of candidate λ_r values up to the square root of total number of instances.

Algorithm: Choosing m candidate values of the tuning parameter λ_r.

1. Compute the surrogate values of $L(J')$ for $1 \leq J' \leq \lfloor \sqrt{I} \rfloor := J_{\max}$,
2. Let $\lambda'_j = (L(j-1) - L(j+1))/2$ for $2 \leq j \leq J_{\max} - 1$
3. Choose $\frac{1}{m+2}$-th, $\frac{1}{m+2}$-th, $\cdots$, $\frac{m}{m+2}$-th quantile in the $\{\lambda'_r\}$ as candidate values.
4. Given a selected λ_r, choose the initial number of clusters as $J \leftarrow \arg\min_j |\lambda'_j - \lambda_r|$.

Algorithm 2. Algorithm for choosing m candidate λ_r values.

Finally, we use the Silhouette score [27], which has been successfully used for evaluating goodness of fit in clustering, across these values of the tuning parameters.

3 Results

3.1 Computational Experiments

We designed computational experiments to evaluate MAD-Bayes MBASIC in settings where the underlying truth is known. In our experiments, we considered I user-specified loci (e.g., promoters from I genes, binding sites of a transcription factor, or peaks from a ChIP-seq experiment). Given multiple simulated ChIP-seq datasets, there are different "baseline" methods for performing these loci-focused analysis. Therefore, in addition to MBASIC, we considered such alternative approaches that practitioners might adopt.

- **MBASIC**: The EM algorithm on the full MBASIC model, where singleton, i.e., unclusterable loci, are also taken into account.
- **SE-HC**: A two-stage method with first **S**tate **E**stimation on individual datasets (i.e., conventional peak calling), and then combining the results by hierarchical clustering on the posterior probabilities of the states $\tilde{\theta}_{iks} = P(\theta_{iks} = 1|Y)$ from the first stage.
- **SE-MC**: A two-stage method with first **S**tate **E**stimation on individual datasets (i.e., conventional peak calling), and then combining the results by mixture clustering on the binarized results $\theta^*_{iks_0} = 1$, where $s_0 = \arg\max_s P(\theta_{iks} = 1|Y)$ from the first stage.

– **PE-MC**: A two stage method with first **P**arameter **E**stimation on individual datasets to determine the state-specific observations distributions (e.g., distributions of the read counts), and then combining the results by simultaneous state inference and mixture clustering. This is essentially similar to MBASIC, except that state-specific densities are fixed and not updated at every iteration.

The alternatives to MBASIC use two-stage procedures for model estimation, decoupling either the estimation of the state-space variables or the distributional parameters from the mixture modeling of state-space clustering. For example, SE-HC corresponds to overlapping user-loci with the peak sets from the ENCODE project and generating and clustering the binary overlap or peak confidence profiles of the loci. In contrast, PE-MC is analogous to estimating the distributional parameters of state-space for each individual experiment separately and then clustering with these fixed distributions as in [6,28]. These benchmark algorithms are in spirit analogous to procedures in many applied genomic data analyses where the association between observational units are estimated separately from the estimation of individual data set specific parameters [28–30].

For the MAD-Bayes algorithm, we evaluated all the three clustering initializations: Adaptive K means, K means, and Kmeans++. The MAD-Bayes algorithm automatically selects the number of clusters. We used the Silhouette score for SE-HC to accommodate hierarchical clustering and used Bayesian Information Criterion for the other methods. The experiments utilized $I = 4,000$ genomic loci, $J = 10$ clusters, and $K = 20$ experimental conditions. For each condition, the number of replicates, n_k, were drawn from 1 to 3 with probabilities $(0.3, 0.5, 0.2)$. The clustering concentration parameter was simulated from non-informative prior $\alpha \sim \text{Dir}(0.1, \cdots, 0.1)$. The state probabilities, w_{jks}s, were simulated from $\text{Dir}(1, \cdots, 1)$. The Log-normal parameters were set as follows: the mean was simulated from $N(2*s, 0.05^2)$, where s represented the state label; and the standard error was set to 0.5. We considered four scenarios by varying the number of states S between 2 and 4, and the proportion of singleton loci as $\zeta = 0, 0.4$. Here, singletons represented loci with overall ChIP-seq enrichment profile different than the clusters, i.e., unclusterable locus, and introduced noise to the model. Results for each setting were summarized over 10 simulated datasets. We compared the algorithms in terms of run-time, state-space inference (identifying whether or not each locus is bound), and also the clustering structure via the adjusted Rand index [31].

Figure 2(a) displays run-time comparisons of the methods and indicates that all three implementations of the MAD-Bayes algorithm are about 100 times faster than the EM on full MBASIC and the PE-MC algorithm, and about 10 times faster than the two-step SE-HC and SE-MC algorithms. This speed improvement is significant and makes it possible for the MBASIC framework to scale up. For example, MAD-Bayes can process $I = 100,000$ and $K = 2000$ (e.g., 100,000 DNase accessible regions in the genome across all the available

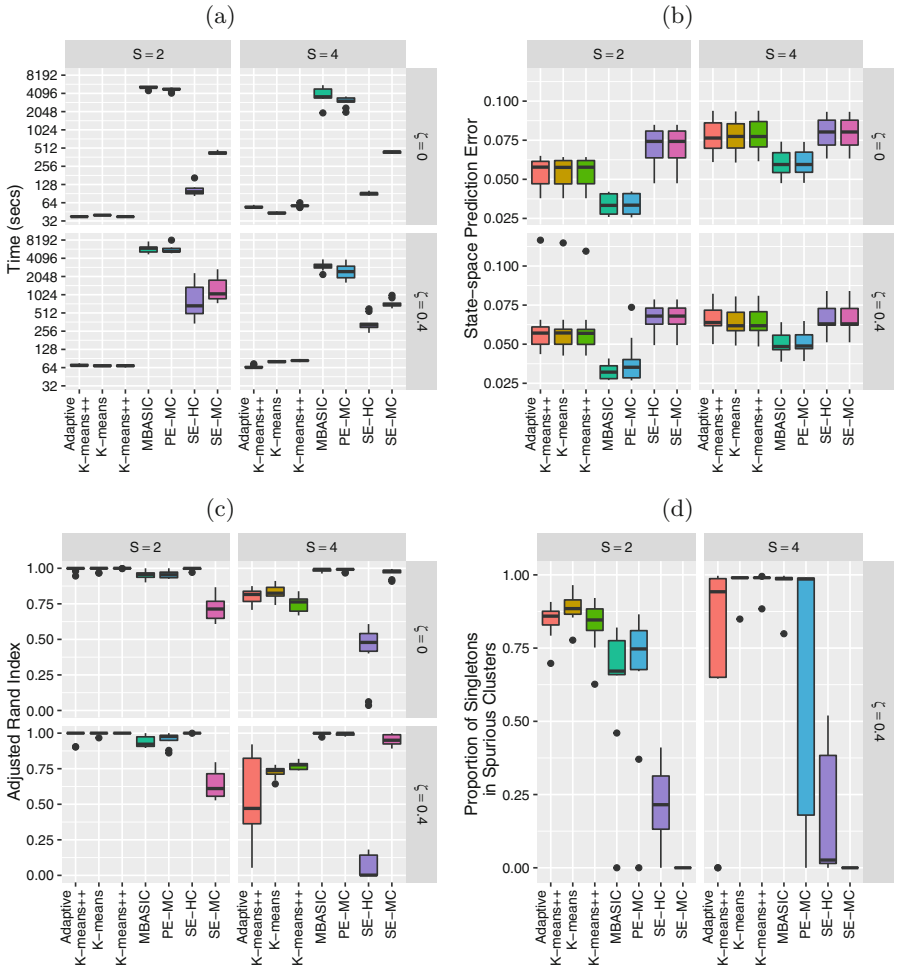


Fig. 2. (a) Run-time comparisons on a 64 bit machine with Intel Xeon 3.0 GHz processor and 64 GB of RAM and 8 cores. (b) State-space prediction error. (c) Clustering accuracy based on the adjusted Rand index. (d) Clustering assignments of the singletons when $\zeta = 0.4$.

ENCODE ChIP-seq data) in about 6 hours while the EM algorithm on full MBASIC requires more than a week.

We also observe that speed up in run time does not come at a significant loss in accuracy. Figure 2(b) compares state-space prediction errors of the algorithms and indicates that while MAD-Bayes MBASIC does not perform as accurately as the EM algorithm on full MBASIC and PE-MC, it performs significantly better than SE-HC and SE-MC algorithms, both of which would be the baseline choices for many practitioners. Existence of singleton genomic loci deteriorate

performance of all the algorithms. When there are no singletons, MAD-Bayes with varying cluster initializations perform the best (Fig. 2(c)). When $\zeta = 0.4$ indicating that 40 % genomic loci do not belong to any cluster, the MAD-Bayes algorithm tends generate extra, i.e., spurious, clusters for such loci (Fig. 2(d)) instead of forcing them into other clusters. As a result, the true clusters are largely preserved and less polluted by singletons (Fig. 5) compared to other methods which do not handle singletons (PE-MC, SE-HC, SE-MC).

3.2 Application to Histone ChIP-Seq Data from GM12878 Cells

The key inference question for the MBASIC framework is identifying the enrichment patterns for a given set of user-specified loci across large sets of ChIP-seq datasets and grouping these loci to elucidate similarities and differences. From this point of view, the MBASIC framework is more loci-focused and not directly comparable with any of the available joint analysis methods that can handle large datasets. However, to get a general sense of how MBASIC would compare with ChromHMM [10] and its computationally efficient version Spectacle [12], we analyzed ChIP-seq data of 8 histone marks (H3k4me1, H3k4me2, H3k4me3, H3k9ac, H3k27ac, H3k27me3, H3k36me3, and H4k20me1 from GM12878 cells) from the ENCODE project. Raw data and peak calls for these marks are available at https://www.encodeproject.org/. We used the 9038 peaks on chr 18 from the ENCODE uniform processing pipeline as the input loci to MAD-Bayes MBASIC and fixed the number of clusters as 20 since Spectacle identified robust number of chromatin states across multiple chromatin modification datasets as 20. As a result, we also set the number of emission states in chromHMM as 20.

We then performed pairwise comparisons of all the three approaches by matching their clusters/states via maximizing the sum of Jaccard index [32]. We reordered the cluster/state labels of MAD-Bayes and Spectacle according to their agreement with ChromHMM. For example, MAD-Bayes cluster "C1" and Spectacle emission state "E1" are both matched to ChromHMM emission state "E1"; however, this does not necessarily indicate that these two are the best matches between MAD-Bayes and Spectacle.

Figure 3(a) displays that the overall agreements between MAD-Bayes vs. Spectacle and MAD-Bayes vs. ChromHMM follow the same trend with the degree of agreement between Spectacle vs. ChromHMM, which we think of as the baseline agreement since they are both HMM based. In particular, for the emission states with agreement between Spectacle vs. ChromHMM, the corresponding MAD-Bayes clusters also have higher agreement with these. When there is large discrepancy between Spectacle vs. ChromHMM, the MAD-Bayes clusters tend to agree with results from one of the methods. For example, MAD-Bayes "C2" agrees better with Spectacle, and MAD-Bayes "C18" overlaps better with ChromHMM. Figure 3(b) and (c) display comparisons of MAD-Bayes MBASIC to ChromHMM and Spectacle, respectively. We observe that some of MAD-Bayes clusters are distributed over multiple clusters of ChromHMM and Spectacle, e.g., MAD-Bayes cluster "C5" overlaps with the "E12", "E13", "E14" of both ChromHMM and Spectacle. This overall agreement indicates that the

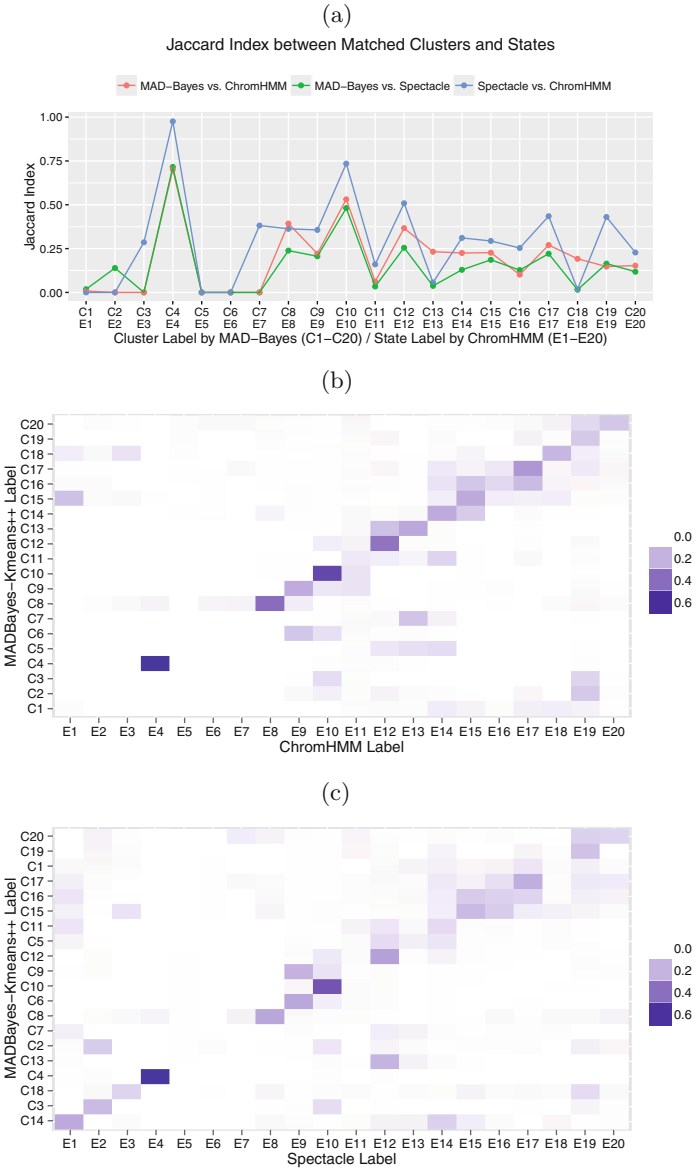


Fig. 3. (a) Comparison of clusters and state labels between MAD-Bayes, Spectacle, and ChromHMM. (b) Jaccard index between MAD-Bayes clusters and ChromHMM states. (c) Jaccard index between MAD-Bayes clusters and Spectacle states. The diagonal blocks indicate agreement between clusters and states; MAD-Bayes clusters and Spectacle states are ordered according to their overlap with the ChromHMM states.

clustering task of MAD-Bayes on the histone marks is reasonable even though it is using selected loci and is not accounting for local dependencies inherent among genomic loci with broad histone marks.

4 Discussion

In this paper, we derived a MAD-Bayes algorithm by developing a Bayesian version of the MBASIC model. Our evaluations indicated that MAD-Bayes MBASIC significantly improves the computational time without sacrificing accuracy. We also observed that even though MAD-Bayes MBASIC does not have a built-in mechanism for singletons (unclusterable loci), it groups singletons as additional clusters and minimizes their effect on other more coherent clusters.

We developed MAD-Bayes MBASIC as a fast method for querying large sets (1000s) of ChIP-seq data with user-specified large sets of loci. This represents the first application of the MAD-Bayes framework in a large scale genome regulation context. From a practical point of view, we showed that this approach is both more efficient and powerful than using individual analysis of each datasets and clustering them with an off-the-shelf method such as hierarchical clustering or finite mixture models. From an algorithmic point of view, we developed an empirical method for selecting tuning parameters. This improves the current state-of-the-art for MAD-Bayes implementations since they lack principled methods for tuning parameter selection. The MBASIC framework offers flexibility in a number of aspects of experimental design, such as different numbers of replicates under individual experimental conditions. This is a relatively important point because many peak callers will operate separately on individual peaks sets or handle two jointly [33] leaving the reconciliation of peaks over multiple replicates to the user. Our current derivation of the MAD-Bayes relied on Log-normal distribution; however, it can be extended to larger class of exponential family distributions via the Bregman divergence [34]. Such extensions are likely to foster its use with other genomic data types such as RNA-seq, DNAse-seq, and Methyl-seq, where both state-space estimation and clustering of similar loci pose significant challenges.

Appendix

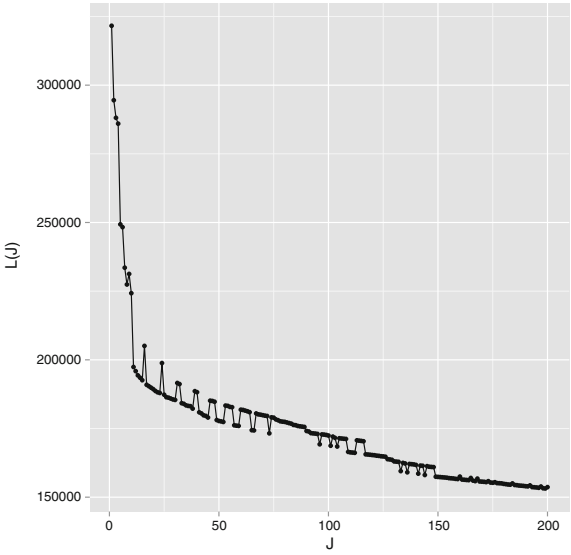


Fig. 4. A graphical interpretation of the conjugacy between λ_r and J. We use the K-means initialization to compute surrogate values for $L(J)$ for a large collection of $J \geq 1$. The λ_r value that can yield J clusters in the global solution must satisfy: $\sup_{J'>J} \frac{L(J)-L(J')}{J-J'} \leq \lambda_r \leq \inf_{J'>J} \frac{L(J')-L(J)}{J'-J}$. When λ_r satisfies this condition, a line with slope $-\lambda_r$ passing through $(J, L(J))$ on the graph should be tangent to the trace of all $L(J)$ values. Although using the surrogate $L(J)$ values can lead to the curve connecting the $L(J)$ values to be con-convex, making the solution for λ_r not hold for some J, we can use a convex approximation to the trace of $L(J)$ so that so that a λ_r exists for each J. A simpler approach is to order the $L(J)$ from largest to smallest and require the following condition for λ_r. $L(J) - L(J+1) \leq \lambda_r \leq L(J-1) - L(J)$. Algorithm 2 essentially applies this idea to select the λ_r values. Each J corresponds to a λ_r of value $[L(J-1) - L(J+1)]/2$ that satisfies the conjugacy inequality. The algorithm essentially tries to identify the range of λ_r that leads up to $\sqrt{I}$ number of clusters.

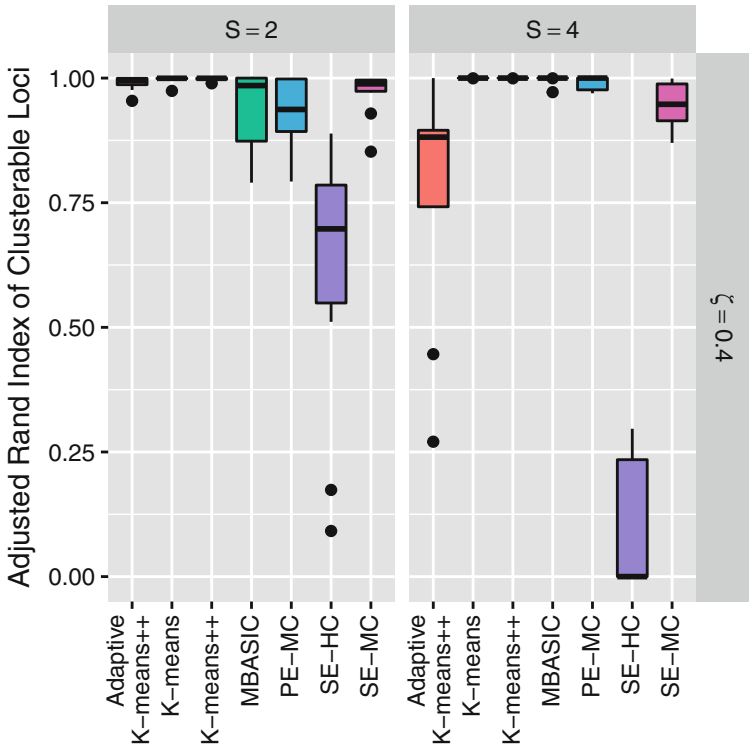


Fig. 5. Comparison of the clustering accuracy with the adjusted Rand index by excluding the singleton loci.

References

1. Zuo, C., Hewitt, K.J., Bresnick, E.H., Keleş, S.: A hierarchical framework for state-space matrix inference and clustering. Ann. Appl. Stat. (Revised)
2. The ENCODE project consortium: an integrated encyclopedia of DNA elements in the human genome. Nature **489**, 57–74 (2012)
3. Roadmap epigenomics consortium: integrative analysis of 111 reference human epigenomes. Nature **518**(7539), 317–330 (2015)
4. Bardet, A.F., He, Q., Zeitlinger, J., Stark, A.: A computational pipeline for comparative ChIP-seq analyses. Nat. Protoc. **7**(1), 45–61 (2012)
5. Bao, Y., Vinciotti, V., Wit, E., AC't Hoen, P.: Accounting for immunoprecipitation efficiencies in the statistical analysis of ChIP-seq data. BMC Bioinform. **14**(1), 169 (2013)
6. Zeng, X., Sanalkumar, R., Bresnick, E.H., Li, H., Chang, Q., Keleş, S.: jMOSAiCS: joint analysis of multiple ChIP-seq datasets. Genome Biol. **14**, R38 (2013). Highly accessed. An R package for joint analysis of multiple ChIP-seq datasets. Available in Bioconductor http://bioconductor.org/packages/2.12/bioc/html/jmosaics.html

7. Kuan, P.F., Chung, D., Pan, G., Thomson, J., Stewart, R., Keleş, S.: A statistical framework for the analysis of ChIP-Seq data. J. Am. Stat. Assoc. **106**, 891–903 (2011). Software available on Galaxy http://toolshed.g2.bx.psu.edu/ and also on Bioconductor http://bioconductor.org/packages/2.8/bioc/html/mosaics.html
8. Bao, Y., Vinciotti, V., Wit, E., 't Hoen, P.: Joint modeling of ChIP-seq data via a Markov random field model. Biostatistics **15**(2), 296–310 (2014)
9. Chen, K.B., Hardison, R., Zhang, Y.: dCaP: detecting differential binding events in multiple conditions and proteins. BMC Genomics **15**(9), 1–14 (2014)
10. Ernst, J., Kellis, M.: Discovery and characterization of chromatin states for systematic annotation of the human genome. Nat. Biotechnol. **28**(8), 817–825 (2010)
11. Hoffman, M.M., Buske, O.J., Wang, J., Weng, Z., Bilmes, J.A., Noble, W.S.: Unsupervised pattern discovery in human chromatin structure through genomic segmentation. Nat. Methods **9**, 473–476 (2012)
12. Song, J., Chen, K.C.: Spectacle: fast chromatin state annotation using spectral learning. Genome Biol. **16**(1), 33 (2015)
13. Sohn, K.A., Ho, J.W.K., Djordjevic, D., Jeong, H.H., Park, P.J., Kim, J.H.: hiHMM: Bayesian non-parametric joint inference of chromatin state maps. Bioinformatics, btv117 (2015)
14. Liang, K., Keleş, S.: Detecting differential binding of transcription factors with ChIP-seq. Bioinformatics **28**(1), 121–122 (2012). Available in Bioconductor (http://www.bioconductor.org/packages/2.12/bioc/html/DBChIP.html)
15. Mahony, S., Edwards, M.D., Mazzoni, E.O., Sherwood, R.I., Kakumanu, A., Morrison, C.A., Wichterle, H., Gifford, D.K.: An integrated model of multiple-condition ChIP-Seq data reveals predeterminants of Cdx2 binding. PLoS Comput. Biol. **10**(3), e1003501 (2014)
16. Song, Q., Smith, A.D.: Identifying dispersed epigenomic domains from ChIP-Seq data. Bioinformatics **27**, 870–1 (2011)
17. Ferguson, J.P., Cho, J.H., Zhao, H.: A new approach for the joint analysis of multiple ChIP-seq libraries with application to histone modification. Stat. Appl. Genet. Mol. Biol. **11**(3), Article 1 (2012)
18. Taslim, C., Huang, T., Lin, S.: DIME: R-package for identifying differential ChIP-seq based on an ensemble of mixture models. Bioinformatics **27**(11), 1569–70 (2011)
19. Ji, H., Li, X., Wang, Q.F., Ning, Y.: Differential principal component analysis of ChIP-seq. Proc. Nat. Acad. Sci. U.S.A. **110**(17), 6789–6794 (2013)
20. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Stat. Soc. B Met. **39**, 1–38 (1977)
21. Zuo, C., Keleş, S.: A statistical framework for power calculations in ChIP-seq experiments. Bioinformatics **30**(6), 853–860 (2014)
22. Broderick, T., Kulis, B., Jordan, M.: MAD-Bayes: MAP-based asymptotic derivations from Bayes. In: Proceedings of the 30th International Conference on Machine Learning (2013)
23. Blackwell, D., MacQueen, J.B.: Ferguson distributions via Polya urn schemes. Ann. Stat. **1**(2), 353–355 (1973)
24. Aldous, D.J.: Exchangeability and related topics. In: Hennequin, P.L. (ed.) École d'Été de Probabilités de Saint-Flour XIII, vol. 1117, pp. 1–198. Springer, Heidelberg (1983)
25. Hewitt, K.J., Kim, D.H., Devadas, P., Prathibha, R., Zuo, C., Sanalkumar, R., Johnson, K.D., Kang, Y.A., Kim, J.S., Dewey, C.N., Keleş, S., Bresnick, E.: Hematopoietic signaling mechanism revealed from a stem/progenitor cell cistrome. Mol. Cell **59**(1), 62–74 (2015)

26. Johnson, K.D., Hsu, A., Ryu, M.J., Boyer, M.E., Keleş, S., Zhang, J., Lee, Y., Holland, S.M., Bresnick, E.H.: Cis-element mutation in a GATA-2-dependent immunodeficiency syndrome governs hematopoiesis and vascular integrity. J. Clin. Inv. **10**(122), 3692–3704 (2012)
27. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. **20**, 53–65 (1987)
28. Wei, Y., Li, X., Wang, Q.F., Ji, H.: iASeq: integrative analysis of allele-specificity of protein-DNA interactions in multiple ChIP-seq datasets. BMC Genomics **13**, 681 (2012)
29. Gerstein, M.B., Kundaje, A., Hariharan, M., Landt, S.G., Yan, K.K., Cheng, C., Mu, X.J., Khurana, E., Rozowsky, J., Alexander, R., Min, R., Alves, P., Abyzov, A., Addleman, N., Bhardwaj, N., Boyle, A.P., Cayting, P., Charos, A., Chen, D.Z., Cheng, Y., Clarke, D., Eastman, C., Euskirchen, G., Frietze, S., Fu, Y., Gertz, J., Grubert, F., Harmanci, A., Jain, P., Kasowski, M., Lacroute, P., Leng, J., Lian, J., Monahan, H., O'Geen, H., Ouyang, Z., Partridge, E.C., Patacsil, D., Pauli, F., Raha, D., Ramirez, L., Reddy, T.E., Reed, B., Shi, M., Slifer, T., Wang, J., Wu, L., Yang, X., Yip, K.Y., Zilberman-Schapira, G., Batzoglou, S., Sidow, A., Farnham, P.J., Myers, R.M., Weissman, S.M., Snyder, M.: Architecture of the human regulatory network derived from ENCODE data. Nature **489**(7414), 91–100 (2012)
30. Wei, Y., Tenzen, T., Ji, H.: Joint analysis of differential gene expression in multiple studies using correlation motifs. Biostatistics **16**(1), 31–46 (2015)
31. Rand, W.M.: Objective criteria for the evaluation of clustering methods. J. Am. Stat. Assoc. **66**(336), 846–850 (1971)
32. Tan, P.N., Steinbach, M., Kumar, V.: Cluster analysis: basic concepts and algorithms. In: Introduction to Data Mining, chap. 8 (2005)
33. Landt, S.G., Marinov, G.K., Kundaje, A., Kheradpour, P., Pauli, F., Batzoglou, S., Bernstein, B.E., Bickel, P., Brown, J.B., Cayting, P., et al.: ChIP-seq guidelines and practices of the encode and modencode consortia. Genome Res. **22**(9), 1813–1831 (2012)
34. Banerjee, A.: Clustering with Bregman divergences. J. Mach. Learn. Res. **6**, 1705–1749 (2005)

Accurate Recovery of Ribosome Positions Reveals Slow Translation of Wobble-Pairing Codons in Yeast

Hao Wang[1], Joel McManus[1,2], and Carl Kingsford[1(✉)]

[1] Computational Biology Department, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA
carlk@cs.cmu.edu

[2] Department of Biological Sciences, Carnegie Mellon University, Pittsburgh, PA, USA

Abstract. Ribosome profiling quantitatively captures ribosome locations during translation. The resulting profiles of ribosome locations are widely used to study translational speed. However, an accurate estimation of the ribosome location depends on identifying the A-site from ribosome profiling reads, a problem that was previously unsolved. Here, we propose a novel method to estimate the ribosome A-site positions from high-coverage ribosome profiling reads. Our model allows more reads to be used, accurately explains the 3-nt periodicity of ribosome profiling reads from various lengths, and recovers consistent ribosome positions across different lengths. Our recovered ribosome positions are correctly highly skewed towards a single frame within a codon. They retain subcodon resolution and enable detection of off-frame translational events, such as frameshifts. Our method improves the correlation with other estimates of codon decoding time. Further, the refined profiles show that yeast wobble-pairing codons are translated slower than their synonymous Watson-Crick-pairing codons. These results provide evidence that protein synthetic rate can be tuned by codon usage bias.

Keywords: Ribosome profiling · A-site recovery · Translation rate

1 Introduction

Ribosome profiling is an important sequencing technique that enables various genome-wide translational studies, including on translational response to stress [16,20,37], protein synthesis rate [24], alternative translation initiation [13,23], translation evolution [3,26], cell development [4,34], and the role of specific translation regulation factors [17,18,38]. The experiment extracts mRNA fragments protected by bound ribosomes (also called ribosome footprints) from RNase I digestion [20]. The technique is analogous to taking snapshots of ribosome locations during translation. Therefore the ribosome footprint counts at codon locations should be related to the elongation time [19,21]. The vector

M. Singh (Ed.): RECOMB 2016, LNBI 9649, pp. 37–52, 2016.
DOI: 10.1007/978-3-319-31957-5_3

of footprint counts at codon locations of a mRNA is called a ribosome profile, and each individual count is called a ribosome pileup. To date, ribosome profiles are generally used to qualitatively visualize ribosome pauses (e.g., [18,21]), translation initiation, and translation termination (e.g., [1,10]). Yet attempts to quantify translation speed, even from the same experiment, often result in controversial conclusions on the determinants of translation rate [2].

One of the challenges in translation speed quantification is accurate measurement of ribosome decoding locations. Currently, there is no method to extract the precise ribosome decoding locations when the snapshots are taken [2,25]. The ribosome P-site or A-site is usually considered the active decoding site [2,14, 21,22,25,27,30,35,38]. This is because the A-site is where the aminoacyl-tRNA enters the ribosome, and the P-site is the position of peptide bond formation. Only the location of either the P-site or the A-site needs to be estimated from the experiment data, and the other one can be inferred.

In past analyses, the A-site location estimation is usually based on simple heuristics. One widely used strategy is that the A-site is simply placed at 15 bases away from the 5' end of the footprint read [20,27,32,35]. This is shown to be accurate for the typical ribosome footprint size (about 28 nt for yeast) [20]. However, the read length from ribosome profiling experiments can span a wide range [18,22,25,38], with as little as 40 % being 28-nt reads [26]. The A-site position for 28-nt reads might not be suitable for other read lengths.

Since a read length not equal to the typical footprint size is mainly caused by incomplete RNase digestion during the experimental procedure [19], an alternative strategy is to use a constant A-site offset for a given read length [7,21,22]. This assumes that the digested portion is always the same for all reads with the same length. Because ribosomes move in units of codons (3 nt), such a strategy implies a 3-nt periodic ribosome position pileups. For every 3-nt period, these pileups should also be highly concentrated on a single base (a reading frame). However, such a highly skewed frame distribution is not always observed in read pileups for all read lengths (See one example in Sect. S1, Fig. S1). Thus, a large fraction of ribosome footprints have incomplete or over-digestion (length $\neq 28$), and the simple offset heuristic is insufficient to explain the observed complex frame distribution pattern caused by various nuclease digestion possibilities.

In short, ribosome profiling is a powerful technique to study genome-wide translation mechanisms, but ribosome profiling data are inherently noisy due to complicated experiment pipelines. Specifically, imperfect RNase digestions distort true ribosome profiles and might bury biologically meaningful insights. Such complicated non-universal digestions vary between replicates and laboratories and cannot be well captured by existing simple heuristics of A-site assignments. We introduce a new model and computational method to recover the A-site positions from ribosome profiling data. Our method does not make the incorrect assumption that all reads with the same size are digested to the same extent. Instead, we systematically remove the distortion caused by imperfect digestions and retrieve true ribosome positions. Our procedure results in better A-site position estimation, which enables comparisons of ribosome profiling data

from different replicates, conditions, and labs, and will hopefully lead to a better understanding of translation speed and regulation.

2 Contributions

Observing that read pileups for each read length have a unique start for the 3-nt periodicity, we assume that there is a predominant digestion pattern for each read length. However, individual reads can be over-digested or under-digested to a certain amount centered around this major digestion pattern. Such an imperfect digestion causes the ribosome A-site to be a variable distance away from the read start. We also assume that there is an unknown underlying true A-site profile consistent across all read lengths. We define this true A-site profile as the ribosome position signal. Such a signal at a particular location is blurred to its surrounding neighborhood due to imperfect RNase digestions. We therefore model the observed read pileups as a blurring of the unknown ground truth positions. We then recover the ground truth positions by combining read pileups from different lengths and allowing the reads to be re-allocated with a non-universal A-site offset (deblur).

Compared to previous work, our procedure does not assume any specific prior distribution of RNase digestion patterns, nor do we assume the imperfect digestion is limited to a 3-nt window [7,39]. Rather, we learn the probabilities of the digestion for each read length from the observed data, enabling a more flexible model to explain the ribosome read pileups. Also, unlike heuristics that discard the off-frame reads [35] or take the sum of reads in all three frames [14,30], we do not assume all ribosome reads are from a single reading frame, nor do we need to distinguish reads from different frames. Instead, we re-distribute reads to their nearby loci, naturally causing the ribosome pileups to be concentrated towards a single frame within a codon. Our approach therefore preserves the sub-codon resolution in the estimated A-site positions. We show that on a synthetic frameshift test set, our method retains the frame preferences and strengthens the frame skewness in the estimated A-site profiles.

We showcase our method by estimating codon decoding time (CDT) [7] in yeast ribosome profiling data [1]. Although abundant tRNAs are expected to speed up codon decoding, the naïve global offset heuristic only recovers a weak negative correlation between the tRNA abundance estimates and the CDT. This correlation improves after using our deblurred profiles. Also, for codons decoded by the same tRNA, our estimated CDT shows that the less stable wobble pairing codons generally translate more slowly than their synonymous codons with Watson-Crick pairing. We find that the difference in decoding time between Watson-Crick-paired codons and wobble-paired codons is generally larger than the difference between two wobble-paired codons. Such phenomena was previously only observed in metazoans [35]. This observation is consistent with the expectation that wobble pairing is likely to be delayed by the higher probability of tRNA rejection [36]. Our result therefore provides evidence for the first time in yeast to support such a mechanism. Together, our analysis gives further

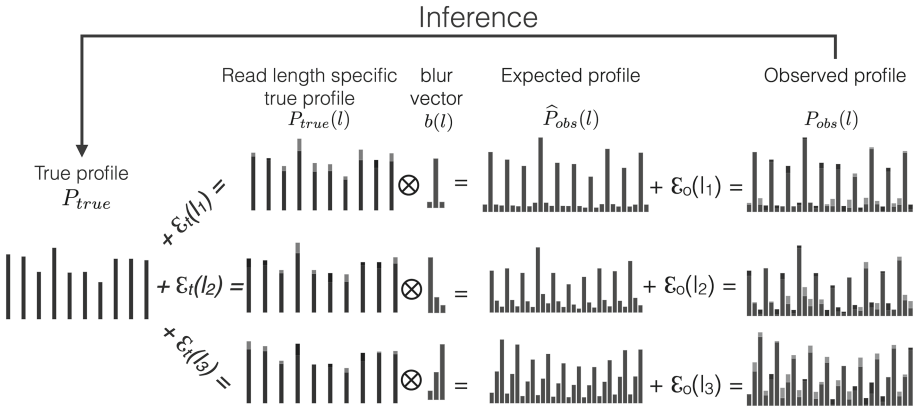


Fig. 1. Model of the observed ribosome profiling read pileups. The observed read pileups $P_{obs}(l)$ for read length l are modeled as a convolution effect between a blur vector $b(l)$ and a clear ribosome position signal $P_{true}(l)$. The blur vector diffuses a signal to its nearby locations. The clear signal is somewhat consistent across all read lengths, and can be captured by a consensus clear signal P_{true}. An additive slack variable ε_t is used to match $P_{true}(l)$ with P_{true}, and an additive error ε_o is used to match the modeled pileups with the observed pileups. Our goal is to extract the consensus clear ribosome positions P_{true} from the observed ribosome pileups for all read lengths ($P_{obs}(l)$).

evidence that frequent codons translate faster than rare codons, and that both tRNA abundance and wobble pairing play roles in elongation speed.

3 Methods

3.1 Algorithm Overview

For a given transcript and each read length l, let $P_{obs}(l)$ be the observed ribosome distribution from ribosome profiling reads. We model $P_{obs}(l)$ as the result of a blurring effect on an unknown, lengh-specific clear ribosome position signal $P_{true}(l)$. We assume such a position signal is consistent across all read lengths, and is deviated from an unknown consensus position signal P_{true} (Fig. 1). We aim to recover the clear position signal from the observed blurred version of the read positions across all read lengths. The length-specific clear signals $P_{true}(l)$ should be consistent with each other, and our modeled positions $\widehat{P}_{obs}(l)$ should agree well with the observed read positions $P_{obs}(l)$. We formulate this task as a total least square optimization problem, where the difference between P_{true} and $P_{true}(l)$ and the difference between $P_{obs}(l)$ and $\widehat{P}_{obs}(l)$ are simultaneously minimized. We develop an EM-like procedure to optimize the objective and to extract the hidden clear position signal P_{true} concurrently. One example of our deblur result is shown in Fig. S2.

3.2 Modeling Observed Profiles as Blurred Ribosome Position Signals

We model the observed ribosome read distribution $P_{obs}(l)$ for read length l as a convolution between an unknown clear ribosome position distribution $P_{true}(l)$ and an unknown blur probability vector $b(l)$: $\widehat{P}_{obs}(l) = b(l) * P_{true}(l)$, where $*$ is the convolution operator. The blur vector diffuses the position signal to its neighbor areas. This means, for location i on a transcript, the estimated observed ribosome abundance is a linear combination of the nearby true signals:

$$\widehat{P}_{obs}(l)[i] = \sum_{j=-w}^{w} b(l)[j] \times P_{true}(l)[i-j],$$

where w is the width of the blurring effect. The notation $x[i]$ indicates the ith element of vector x.

We require $P_{true}(l)$ to be as consistent as possible across all read lengths. Specifically:

$$P_{true}(l)[i] = P_{true}[i - k_l] - \varepsilon_t(l)[i - k_l],$$

where P_{true} is the consensus position signal consistent across all read lengths, $\varepsilon_t(l)$ is the deviation of $P_{true}(l)$ from P_{true} due to length-specific digestion preferences, and k_l is a shift to align profiles with different lengths.

Profiles with different lengths can be aligned by observing that the start of the 3-nt periodicity is read length specific. We observe from the meta-profiles that the 3-nt peridicity for reads with length l starts at $-l + 16$ (Fig. S1). Therefore the amount of shift between profile of length l_1 and profile of length l_2 is $-l_1 + 16 - (-l_2 + 16) = l_2 - l_1$. In our model, to align profiles with different lengths, $P_{true}(28)$ is used as the anchor, therefore $P_{true}(l)$ can be aligned to $P_{true}(28)$ by shifting $k_l = l - 28$ to the right. We denote by $P_{true}^{k_l}$ and $\varepsilon_t^{k_l}(l)$ the shifted version of the original vectors.

The starts of the 3-nt periodicity also indicate the locations of the majority of the ribosome read 5' boundaries when ribosomes start translating. They thus give the most probable A-site offsets for different read lengths. Although these offsets themselves cannot entirely capture the various distances between the A-site and the read boundaries, they serve as a good starting point for explaining the major digestion pattern of a given read length.

Putting everything together, the observed read locations $P_{obs}(l)$ of length l are assumed to be generated from the hidden P_{true} signal as:

$$P_{obs}(l) = \overbrace{\underbrace{\left(P_{true}^{k_l} - \varepsilon_t^{k_l}(l)\right)}_{P_{true}(l)} * b(l)}^{\widehat{P}_{obs}(l)} + \varepsilon_o(l),$$

where $\varepsilon_o(l)$ is the deviation of the modeled profile $\widehat{P}_{obs}(l)$ from the observed profile $P_{obs}(l)$. In short, the hidden consensus P_{true} is shifted with an additive

difference $\varepsilon_t(l)$, convolved with a blur vector $b(l)$ to get the modeled profile $\widehat{P}_{obs}(l)$, and the difference between the observed profile $P_{obs}(l)$ and the modeled profile is then measured with an additive error $\varepsilon_o(l)$. The parameters k_l, $\varepsilon_t(l)$, $\varepsilon_o(l)$, $b(l)$ must be optimized to find the hidden P_{true}. We explained above the rationale of choosing k_l, and we describe how other parameters are optimized in the following sections.

3.3 Deblurring Ribosome Profiles — a Least Square Optimization

Our goal is to use the blurred observed profiles $P_{obs}(l)$ to deconvolve the clear ribosome position signal P_{true} of a transcript. Such clear signals should be consistent across all read lengths, and should be a good estimate of the observed ribosome distribution after applying the blurring effect. The consensus clear position signal P_{true} and the deviation between the consensus and the length-specific ribosome signal ($\varepsilon_t(l)$) are adjusted to minimize two terms: the difference between the observed profile and the modeled profile and the difference between the consensus and the length-specific ribosome signal. Specifically:

$$\min_{\boldsymbol{P_{true}},\boldsymbol{\varepsilon_t(l)}} \sum_l \alpha(l) \left[\|P_{obs}(l) - \widehat{P}_{obs}(l)\|_2^2 + \|P_{true}^{k_l} - P_{true}(l)\|_2^2 \right], \tag{1}$$

where $\alpha(l)$ is the total read count with length l for the tested transcript. Intuitively, if some read length is more abundant, the true position signal recovered from that read length should be weighted more.

Using $P_{true}(l) = P_{true}^{k_l} - \varepsilon_t^{k_l}(l)$ and $\widehat{P}_{obs}(l) = b(l) * \left(P_{true}^{k_l} - \varepsilon_t^{k_l}(l)\right)$, we rewrite (1) to be:

$$\min_{\boldsymbol{P_{true}},\boldsymbol{\varepsilon_t(l)}} \sum_l \alpha(l) \left[\|P_{obs}(l) - b(l) * \left(\boldsymbol{P_{true}^{k_l}} - \boldsymbol{\varepsilon_t^{k_l}(l)}\right)\|_2^2 + \|\boldsymbol{\varepsilon_t^{k_l}(l)}\|_2^2 \right]. \tag{2}$$

If the blur vectors $b(l)$ are known, we can use an EM-like framework to find the least square solution:

M-step: We fix P_{true} and adjust $\varepsilon_t(l)$ to optimize the total least square problem in (2), where ε_t for each l can be optimized separately:

$$\min_{\boldsymbol{\varepsilon_t(l)}} \|b(l) * \boldsymbol{\varepsilon_t^{k_l}(l)} - \left(b(l) * P_{true}^{k_l} - P_{obs}(l)\right)\|_2^2 + \|\boldsymbol{\varepsilon_t^{k_l}(l)}\|_2^2, \tag{3}$$

where bold indicates the variables we are optimizing. The optimal $\varepsilon_t(l)$ is found via a least square solver with Ridge regression [12] (damp = 1).

E-step: We fix $\widehat{P}_{obs}(l)$ and $P_{true}(l)$ as estimated from the M-step, and adjust the consensus P_{true} to minimize the objective in (1). The expected P_{true} is therefore the weighted average of all $P_{true}(l)$. After the M-step, the new estimation of $P_{true}(l)$ is $P_{true}^{k_l} - \varepsilon_t^{k_l}(l)$, so the weighted average of $P_{true}(l)$ is:

$$P'_{true} = \sum_l \frac{\alpha(l)(P_{true}^{k_l} - \varepsilon_t^{k_l}(l))}{\sum_{l'} \alpha(l')} = P_{true} - \sum_l \frac{\alpha(l)\varepsilon_t^{k_l}(l)}{\sum_{l'} \alpha(l')}. \tag{4}$$

We set all negative entries of P'_{true} to be zero, and renormalize P'_{true} so that it sums to 1. This constrains P_{true} to remain valid and in practice appears to have a minor effect on the shape of P_{true}.

We repeat the EM-like procedure until the change of the objective in (2) compared to the objective value from the previous step is smaller than 0.01.

We initially set P_{true} to be the in-frame values of the observed read pileups with length 28:

$$P_{init}[i] = \begin{cases} P_{obs}(28)[i] & \text{if } i \text{ is a multiple of 3,} \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

$P_{obs}(28)$ is used as the initial consensus because 28 is the typical ribosome footprint size for yeast. For such size, the real physical ribosome footprint boundaries should be most likely to overlap with the read ends. This is because an imperfect digestion for reads with length 28 has to be caused by a simultaneous over-digestion from one end and an under-digestion from the other end, which is likely to be relatively rare. Therefore, the observed read pileups with read length 28 should be the most clear and the closest to the ground truth position signal. Indeed, these profiles show the strongest frame skewness and the most visible 3-nt periodicity (Fig. S1).

3.4 Estimating Blur Vectors from Meta-profiles

The deblur process depends on a known set of blur vectors ($b(l)$) — a crucial element to model the imperfect digestion in ribosome reads. These vectors describe the probability of relocating ribosomes to transfer the clear ribosome position signal to the observed read pileups. Since these pileups are read 5' end pileups (see below), essentially the blur vectors adjust a true footprint boundary to the observed read boundary. They therefore indicate the probability of the amount of under/over digestion from the 5' end, and capture the read-length-specific digestion patterns.

These blur vectors can be estimated directly from the ribosome reads via meta-profiles. Meta-profiles are widely used to reveal the positional patterns of ribosome profiles [6,15,17,20–22,33]. They do so by summing read pileups from all transcripts for each position. The blur vectors can be estimated from these meta-profiles because convolution satisfies the distributive property.

To generate the meta-profiles, we group reads by lengths, and accumulate the positions of the 5' ends relative to the start codon for all transcripts. We then include the first 350 locations away from the start codon in the meta-profiles. We only use transcripts with length >350 to reduce convolution boundary effect. Also, to avoid the outlier points biasing the shape of the blur vector, we exclude locations in the meta-profiles with the top 1.65 % highest read counts. This threshold is chosen by assuming the top 5 % of in-frame reads (1/3 of total reads) are outliers.

To estimate the blur vectors, we use an EM-like procedure similar to the earlier deblur optimization. In this procedure, the observed transcript profiles

are replace by the meta-profiles, and the blur vectors are adjustable variables. The procedure is exactly the same as described in the previous section, except that the blur vector is first estimated prior to the M-step:

$$\min_{\boldsymbol{b(l)}} \|M_{true}^{k_l} * \boldsymbol{b(l)} - M_{obs}(l)\|_2^2,$$

where the "M" variables are the meta-profiles, and we replace P_{true} by M_{true} and $P_{obs}(l)$ by $M_{obs}(l)$ in (3) – (5). The blur vector size, which limits the diffusion range of the position signal, is set to 31. This way the signal can be diffused, either to the left or to the right, as far as approximately half the size of a ribosome. A non-negative least square solver (`scipy.optimize.nnls`) is used to find the best $b(l)$. All blur vectors with different read lengths are optimized separately.

3.5 Estimating the A-Site Profile

We merge the length-specific true profiles to get an overall ribosome position signal for a given transcript — the A-site profile. It is the weighted sum of all the length-specific true profiles, shifted to the right by 15:

$$C_{true}[i] = \sum_l \alpha(l) P_{true}(l)[i + k_l - 15].$$

The shift is needed since the true profiles are estimated from the reads 5' ends, and $P_{true}(28)$ is the anchor to align profiles with different length. We shift by 15 since it is the major A-site offset of reads with length 28, and it is the A-site offset under perfect digestion.

3.6 Codon Decoding Time Estimation

To investigate the influence of tRNA abundance and wobble pairing on translation speed, we estimate codon decoding time using the procedure in [7]. The in-frame (frame-0) deblurred read counts are used as the input ribosome count for each codon position. Such counts are normalized by the average ribosome count for each transcript, as is done in [22,38]. Following [7], these normalized counts are grouped by codon types to form codon count distributions, with the exclusion of the first and last 20 codon positions of each transcript and positions with ribosome counts less than 1. Each codon distribution is fit with a log normal distribution. The skewness of the log normal distribution is used as an estimate of the codon decoding time, as it has been shown to be informative for estimating the elongation speed from ribosome profiling data among various species [7].

3.7 Read Alignment and Data Preprocessing

We test the deblur method on ribosome profiling data from *Saccharomyces cerevisiae*, where ambiguous mapping is not ubiquitous. We use ribosome reads from

a yeast study, where the data is of high quality and of high sequencing depth (GSM1335348) [1].

Reads were first aligned to the yeast noncoding RNA reference, which includes rRNA, tRNA, snoRNA, etc., to remove noncoding contaminants. The remaining reads are then mapped to the yeast transcriptome. The yeast noncoding RNA reference and the transcriptome reference are downloaded from the Saccharomyces Genome Database [11]. Alignments are performed with STAR [9] with parameters `--clip3pAdapterSeq CTGTAGGCACCATCAAT --outFilterMismatchNmax 1`, which automatically 'softclips' the unaligned adapter sequences and any unaligned bases at the 5' end of the reads, allowing at most 1 mismatch. Only uniquely mapped reads, about 83 % of the non-contaminated reads, are used to generate the observed profiles $P_{obs}(l)$.

The observed profile of a given length is included for a transcript in the deblur process if more than 50 % of the in-frame loci have non-zero ribosome counts. Here, we define 'in-frame' as the frame with the highest total read count. Only transcripts with at least two observed profiles from different read lengths are tested for deblur.

Such filtering results in 1966 transcripts with high ribosome coverage. This transcript set size agrees with the size of highly expressed transcript set: 2108 transcripts share an estimated expression level >100 transcript per million (TPM) (expression are estimated using Salmon [29] with the RNA-seq data from the same experiment (GSM1335347) [1]).

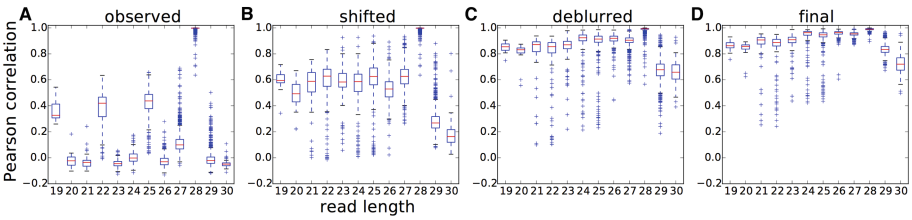


Fig. 2. Effect of shifting and deblurring on profile consistencies across different read lengths. $P_{obs}(l)$ is the observed profile of length l, P_{init} is the in-frame values of $P_{obs}(28)$, which is also the initial guess of the true profile, k_l is the shift applied to a profile, $P_{true}(l)$ is the length-specific deblurred profile, and P_{true} is the consensus of $P_{true}(l)$s. Bar plots of the Pearson correlation are between (A) $P_{obs}(l)$ and P_{init}, (B) $P^{k_l}_{obs}(l)$ and P_{init}, (C) $P^{k_l}_{true}(l)$ and P_{init}, and (D) $P^{k_l}_{true}(l)$ and P_{true}. The improvement of the Pearson correlation between the read-length-specific profiles and the true profiles is the combinational effect of the right amount of shifts and the success of deblurring.

4 Results

4.1 Consistent Read-length-specific Profiles

To test how shifting and deblurring affect the consistencies among profiles with different read lengths, we compare read-length-specific profiles with the in-frame

values of the observed profiles with length 28 (P_{init}, Eq. (5)). We choose P_{init} for comparison because it is the original data in which we have the most confidence. We use the Pearson correlation coefficient as a measurement of the consistency between the read-length-specific profile and P_{init}.

Two factors jointly improve the consistencies of ribosome profiles among different lengths: the deblur procecess, and allowing a length-specific shift and deviation from the consensus. Initially, none of the raw observed profiles ($P_{obs}(l)$) correlate well with the in-frame values of $P_{obs}(28)$ (Fig. 2A). However, the correlations are improved if the observed profiles are properly shifted and aligned to $P_{obs}(28)$ (Fig. 2B). The correlations can be further increased by applying the deblur process to recover the length-specific clear profiles ($P_{true}(l)$) (Fig. 2C). Lastly, compared to the initial guess of the consensus (P_{init}), at the end of the deblur process, the final consensus estimation (P_{true}) correlate better with the length-specific clear profiles (Fig. 2D). Overall, the correlation between $P_{true}(l)$ and P_{true} for most lengths is close to 1. Since P_{true} is the centroid of $P_{true}(l)$, the good correlation between the two indicates that the deblurred profiles are consistent across different read lengths.

4.2 Improved Frame Skewness

Our deblur process improves the frame skewness of the recovered A-site profiles, even if it does not explicitly optimize or force frame skewness. The in-frame position is the reading frame where reads are preferentially distributed within a codon. It is usually frame 0 for the A-sites of ribosome footprints with the absence of frameshifts. It is desirable for the recovered ribosome A-site profiles to be highly skewed towards frame 0. This is because ribosomes move in units of codons, so ribosome profiles should have 3-nt periodicity. Also, such profiles should be mainly concentrated on frame 0, since frameshifts are rare. Indeed, after deblur, the in-frame skewness does improve from an average of 71 % to 92 % (Mann-Whitney U test $p < 3 \times 10^{-308}$; Fig. 3). This indicates that the deblur process produces ribosome profiles with less noise. It also enables more reads to be used in downstream analysis. For instance, if only the in-frame reads are used to represent the codon-level ribosome counts, the deblur process will allow on average 20 % more reads to be used.

4.3 Deblur Process Produces Sub-codon Resolution Profiles

The deblur procedure does not assume that the recovered A-site profiles are all from a fixed frame, thus it keeps the sub-codon resolution of the A-site profiles, and allows detection of potential programmed frameshifts. To test whether the deblur process can recover profiles with frameshifts, we synthetically generate frameshifts as follows: We first choose a random frame-0 location as the frameshift point in a transcript, we then shift all reads with a start location after such point to the right. This is to simulate an insertion in the transcript

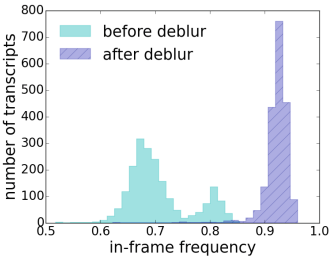


Fig. 3. Histograms of in-frame (frame 0) portion of reads before and after deblur. The recovered A-site profiles have a higher in-frame skewness compared to the original profiles.

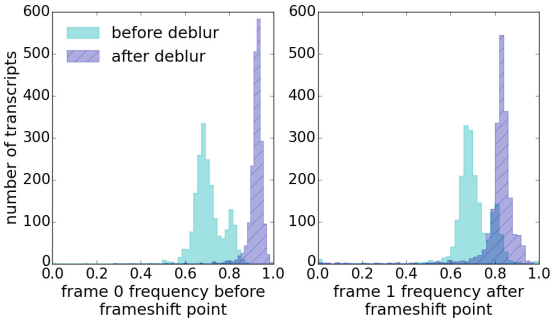


Fig. 4. Histograms of the frame-0 portion of the ribosome profiles before the frameshift point and the frame-1 portion of the ribosome profiles after the frameshift point. The deblur process strengthens the frame skewness while keeping the estimated ribosome positions to be in the correct frame.

to induce a frameshift. The recovered A-site profiles should have a high skewness towards frame 0 before the frameshift point, and a high skewness towards frame 1 after the frameshift point (see example in Fig. S6).

The deblur process successfully maintained and improved the skewness of frame 0 before the frameshift point and the skewness of frame 1 after the frameshift point (Fig. 4). Therefore, combining profiles with lengths other than 28 during the deblur process results in a recovery of a clear frameshifted A-site profile, regardless of the incorrect initial guess. To sum up, frameshift detection is an important task, but current frameshift detection method [27] suffers from high false positive rates. Our deblur process recovers ribosome profiles with a clear frame preference, which will promote the development of a better frameshift detection.

4.4 Wobble Pairing Codons Translate Slower Than Watson-Crick Pairing

The tRNA abundance was expected to be negatively correlated with the codon decoding time (CDT) [7,8,14,22], and such correlation is strengthened using our deblurred profiles. After deblur, 85 % of the codon distributions have a smaller variance, indicating the deblur process successfully removes noise from the observed read pileups. From these distributions, the estimated CDT is the skewness of a lognormal fit [7] (details in Methods). Such estimated CDT is compared with the tRNA Adaptation Index (tAI) [31] — proxy for the tRNA concentration. The deblur process strengthens the Spearman correlation between the tAI and the estimated CDT from -0.21 ($p = 0.1$) to -0.46 ($p = 1 \times 10^{-4}$). This provides stronger evidence that tRNA abundance play roles in elongation

speed. Similarly, the raw frequency of codon usage also negatively correlates with the estimated CDT (Spearman correlation -0.5, $p = 3.7 \times 10^{-5}$), indicating frequent codons are translated faster than rare codons.

Wobble pairing could also affect the elongation speed. Since there are usually fewer tRNA types than codon types, some of the codons that encode the same amino acid must be decoded by the same tRNA. Wobble pairing allows a tRNA to recognize more than one codon. Within these synonymous codons, the determinant of the codon decoding speed is the efficiency of the tRNA recognizing the corresponding codon. According to the Wobble hypothesis [5], the last two bases of the tRNA anticodon form Watson-Crick base pairs and bond strongly to the first two bases of the codon. However, the anticodon's first base can form a wobble pair: The base G can either Watson-Crick pair with C, or wobble pair with U; the base I (inosine, edited from A) can also both wobble pair with C and U, but I:U pairing has a less favorable geometry [35]; the base U can Watson-Crick pair with A, and wobble pair with G. It has been hypothesized that wobble paired codons tend to be translated slower than their synonymous Watson-Crick paired codons, since wobble pairs are more likely to be rejected before peptidyl transfer, causing the tRNA selection cycle to be repeated [36].

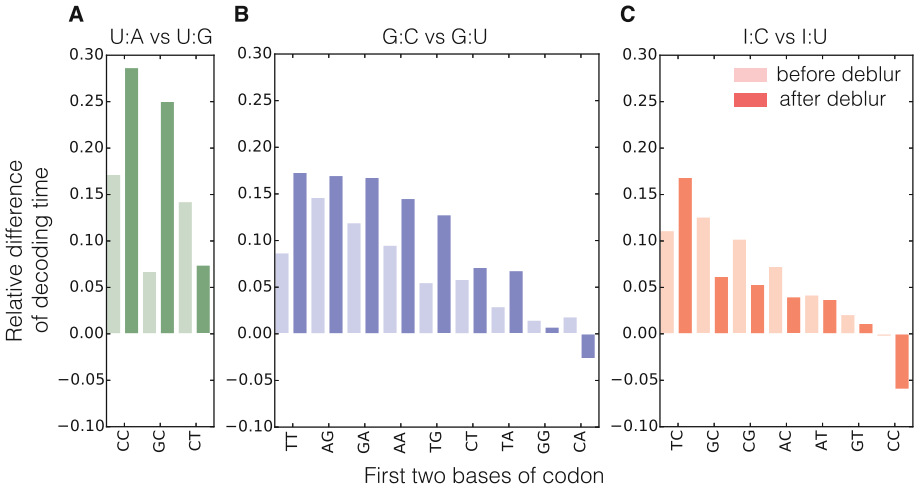


Fig. 5. Relative differences of the CDT between pairs of codons that are decoded by the same tRNA. Lighter colors are time differences estimated from original ribosome profiles, and darker colors are time differences estimated from deblurred profiles. The anticodon : codon pairings are: (A) U:A (Watson-Crick) vs U:G (wobble), (B) G:C (Watson-Crick) vs G:U (wobble), (C) I:C (stronger wobble) vs I:U (weaker wobble). The average relative time difference estimated from the deblurred profiles is: 0.2 between U:A and U:G, 0.1 between G:U and G:C, and 0.04 between I:U and I:C; the average relative time difference estimated from the original profiles is 0.12 between U:A and U:G, 0.07 between G:U and G:C, and 0.07 between I:U and I:C.

We investigated how wobble pairing influences CDT in yeast. We focus on pairs of codons that are translated by the same tRNA, so that the influence of tRNA concentration on the elongation speed is controlled. In this case, the codon pair shares the first two bases, and differs in the third base. We compared the estimated decoding time between the codon pairs, and find that the wobble pairing codons indeed are estimated to often translate slower than the Watson-Crick pairing codons (Fig. 5).

We expect the decoding time difference between two wobble paired codons to be smaller than the difference between a wobble pair codon and a Watson-Crick pair codon, if the wobble-paired tRNA is truly more likely to leave the ribosome without successful peptidyl transfer [36]. For the three codon pairs being compared, we would therefore expect the time difference between I:C and I:U to be smaller than the time difference between G:C and G:U, and between U:A and U:G. To control for the absolute level of the translation time, we use the relative decoding time difference between a synonymous codon pair. It is defined as: $\Delta t = (t_{\text{wobble}} - t_{\text{Watson-Crick}})/t_{\text{Watson-Crick}}$, where t_x is the estimated decoding time for a codon with either wobble pairing or Watson-Crick pairing.

Using the profiles from the deblur process, the decoding time difference is inline with the above expectation. The decoding time difference between a wobble paired codon and a Watson-Crick paired codon is indeed visibly larger than the decoding time difference between two wobble pair codons (Fig. 5). Although such a trend was first seen in metazoans [35], it was not observed for most wobble paired codons in yeast [2,14,28]. It is also less obvious when CDTs are estimated from the original ribosome profiles (Fig. 5). This indicates that the uncorrected ribosome profiles obscure true ribosome A-site positions. Together, the CDT estimated from the deblurred profiles strengthen the conclusion that wobble pairing slows translation. These results also suggest that wobble pairing can be used as a mechanism to regulate elongation speed.

5 Discussion

Estimating ribosome A-site positions from ribosome profiling data is a challenging necessary step in quantifying codon-specific translation speed and ribosome pausing. There are controversial conclusions about whether the tRNA level plays an important role in CDT. Different analysis pipelines performed on different experiments show that the estimated CDT sometimes strongly correlates with the codon usage [14], sometimes weakly correlates with tAI [22], and sometimes does not correlate with the codon optimality [2] or tRNA level [30]. Different estimates of CDT alone produce different correlations between the estimated decoding time and the tRNA level among different species [7,8]. The fact that there is evidence both for and against the correlation between tRNA levels and CDT indicates that a better codon decoding time analysis pipeline is needed.

We here by no means try to touch all aspects of the elongation time estimation, nor do we try to emphasize or diminish the impact of tRNA level on the translation dynamics. We focus on recovering the A-site positions from the

ribosome profiling data, the first step of any quantitative analysis on codon rate or pausing strength. We show via several lines of intrinsic and extrinsic evidence that our deblur method provides better estimates of A-site profiles, leading new insights on translation dynamics. Source code for the deblur method and the analysis can be found at: http://www.cs.cmu.edu/~ckingsf/software/riboasitedeblur/. An appendix including Sect. S1 and Figs. S1, S2, S6 and additional information is available at http://www.cs.cmu.edu/~ckingsf/software/riboasitedeblur/deblur_appendix.pdf.

Acknowledgments. We thank Geet Duggal, Darya Filipova, Heewook Lee, Brad Solomon, Jing Xiang, Hongyi Xing, David Pellow, Pieter Spealman and Chengxi Ye for useful discussions. This research is funded in part by the Gordon and Betty Moore Foundation's Data-Driven Discovery Initiative through Grant GBMF4554 to Carl Kingsford, by the US National Science Foundation (CCF-1256087, CCF-1319998), and by the US National Institutes of Health (R21HG006913, R01HG007104). C.K. received support as an Alfred P. Sloan Research Fellow.

References

1. Albert, F.W., Muzzey, D., Weissman, J.S., Kruglyak, L.: Genetic influences on translation in yeast. PLoS Genet. **10**(10), e1004692 (2014)
2. Artieri, C.G., Fraser, H.B.: Accounting for biases in riboprofiling data indicates a major role for proline in stalling translation. Genome Res. **24**(12), 2011–2021 (2014)
3. Artieri, C.G., Fraser, H.B.: Evolution at two levels of gene expression in yeast. Genome Res. **24**(3), 411–421 (2014)
4. Brar, G.A., Yassour, M., Friedman, N., Regev, A., Ingolia, N.T., Weissman, J.S.: High-resolution view of the yeast meiotic program revealed by ribosome profiling. Science **335**(6068), 552–557 (2012)
5. Crick, F.H.: Codon-anticodon pairing: the wobble hypothesis. J. Mol. Biol. **19**(2), 548–555 (1966)
6. Dana, A., Tuller, T.: Determinants of translation elongation speed and ribosomal profiling biases in mouse embryonic stem cells. PLoS Comput. Biol. **8**(11), e1002755 (2012)
7. Dana, A., Tuller, T.: Properties and determinants of codon decoding time distributions. BMC Genomics **15**(6), S13 (2014)
8. Dana, A., Tuller, T.: The effect of tRNA levels on decoding times of mRNA codons. Nucleic Acids Res. **42**(14), 9171–9181 (2014)
9. Dobin, A., Davis, C.A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., Batut, P., Chaisson, M., Gingeras, T.R.: STAR: ultrafast universal RNA-seq aligner. Bioinformatics **29**(1), 15–21 (2013)
10. Dunn, J.G., Foo, C.K., Belletier, N.G., Gavis, E.R., Weissman, J.S.: Ribosome profiling reveals pervasive and regulated stop codon readthrough in Drosophila melanogaster. Elife **2**, e01179 (2013)
11. Engel, S.R., Cherry, J.M.: The new modern era of yeast genomics: community sequencing and the resulting annotation of multiple *Saccharomyces cerevisiae* strains at the Saccharomyces Genome Database. Database (Oxford) 2013, bat012 (2013)

12. Fong, D.C., Saunders, M.: LSMR: an iterative algorithm for sparse least-squares problems. SIAM J. Sci. Comput. **33**(5), 2950–2971 (2011)
13. Gao, X., Wan, J., Liu, B., Ma, M., Shen, B., Qian, S.B.: Quantitative profiling of initiating ribosomes in vivo. Nat. Methods **12**(2), 147–153 (2015)
14. Gardin, J., Yeasmin, R., Yurovsky, A., Cai, Y., Skiena, S., Futcher, B.: Measurement of average decoding rates of the 61 sense codons in vivo. Elife **3**, e03735 (2014)
15. Gerashchenko, M.V., Gladyshev, V.N.: Translation inhibitors cause abnormalities in ribosome profiling experiments. Nucleic Acids Res. **42**(17), e134 (2014)
16. Gerashchenko, M.V., Lobanov, A.V., Gladyshev, V.N.: Genome-wide ribosome profiling reveals complex translational regulation in response to oxidative stress. Proc. Natl. Acad. Sci. U.S.A. **109**(43), 17394–17399 (2012)
17. Guo, H., Ingolia, N.T., Weissman, J.S., Bartel, D.P.: Mammalian microRNAs predominantly act to decrease target mRNA levels. Nature **466**(7308), 835–840 (2010)
18. Guydosh, N.R., Green, R.: Dom34 rescues ribosomes in 3' untranslated regions. Cell **156**(5), 950–962 (2014)
19. Ingolia, N.T.: Ribosome profiling: new views of translation, from single codons to genome scale. Nat. Rev. Genet. **15**(3), 205–213 (2014)
20. Ingolia, N.T., Ghaemmaghami, S., Newman, J.R., Weissman, J.S.: Genome-wide analysis in vivo of translation with nucleotide resolution using ribosome profiling. Science **324**(5924), 218–223 (2009)
21. Ingolia, N.T., Lareau, L.F., Weissman, J.S.: Ribosome profiling of mouse embryonic stem cells reveals the complexity and dynamics of mammalian proteomes. Cell **147**(4), 789–802 (2011)
22. Lareau, L.F., Hite, D.H., Hogan, G.J., Brown, P.O.: Distinct stages of the translation elongation cycle revealed by sequencing ribosome-protected mRNA fragments. Elife **3**, e01257 (2014)
23. Lee, S., Liu, B., Lee, S., Huang, S.X., Shen, B., Qian, S.B.: Global mapping of translation initiation sites in mammalian cells at single-nucleotide resolution. Proc. Natl. Acad. Sci. U.S.A. **109**(37), E2424–2432 (2012)
24. Li, G.W., Burkhardt, D., Gross, C., Weissman, J.S.: Quantifying absolute protein synthesis rates reveals principles underlying allocation of cellular resources. Cell **157**(3), 624–635 (2014)
25. Martens, A.T., Taylor, J., Hilser, V.J.: Ribosome A and P sites revealed by length analysis of ribosome profiling data. Nucleic Acids Res. **43**(7), 3680–3687 (2015)
26. McManus, C.J., May, G.E., Spealman, P., Shteyman, A.: Ribosome profiling reveals post-transcriptional buffering of divergent gene expression in yeast. Genome Res. **24**(3), 422–430 (2014)
27. Michel, A.M., Choudhury, K.R., Firth, A.E., Ingolia, N.T., Atkins, J.F., Baranov, P.V.: Observation of dually decoded regions of the human genome using ribosome profiling data. Genome Res. **22**(11), 2219–2229 (2012)
28. O'Connor, P., Andreev, D., Baranov, P.: Surveying the relative impact of mRNA features on local ribosome profiling read density in 28 datasets. bioRxiv, 018762 (2015)
29. Patro, R., Duggal, G., Kingsford, C.: Salmon: accurate, versatile and ultrafast quantification from RNA-seq data using lightweight-alignment. bioRxiv, 021592 (2015)
30. Pop, C., Rouskin, S., Ingolia, N.T., Han, L., Phizicky, E.M., Weissman, J.S., Koller, D.: Causal signals between codon bias, mRNA structure, and the efficiency of translation and elongation. Mol. Syst. Biol. **10**, 770 (2014)

31. dos Reis, M., Savva, R., Wernisch, L.: Solving the riddle of codon usage preferences: a test for translational selection. Nucleic Acids Res. **32**(17), 5036–5044 (2004)
32. Sabi, R., Tuller, T.: A comparative genomics study on the effect of individual amino acids on ribosome stalling. BMC Genomics **16**(10), S5 (2015)
33. Shah, P., Ding, Y., Niemczyk, M., Kudla, G., Plotkin, J.B.: Rate-limiting steps in yeast protein translation. Cell **153**(7), 1589–1601 (2013)
34. Stadler, M., Artiles, K., Pak, J., Fire, A.: Contributions of mRNA abundance, ribosome loading, and post- or peri-translational effects to temporal repression of *C. elegans* heterochronic miRNA targets. Genome Res. **22**(12), 2418–2426 (2012)
35. Stadler, M., Fire, A.: Wobble base-pairing slows in vivo translation elongation in metazoans. RNA **17**(12), 2063–2073 (2011)
36. Tarrant, D., von der Haar, T.: Synonymous codons, ribosome speed, and eukaryotic gene expression regulation. Cell. Mol. Life Sci. **71**(21), 4195–4206 (2014)
37. Vaidyanathan, P.P., Zinshteyn, B., Thompson, M.K., Gilbert, W.V.: Protein kinase A regulates gene-specific translational adaptation in differentiating yeast. RNA **20**(6), 912–922 (2014)
38. Woolstenhulme, C.J., Guydosh, N.R., Green, R., Buskirk, A.R.: High-precision analysis of translational pausing by ribosome profiling in bacteria lacking EFP. Cell Rep. **11**(1), 13–21 (2015)
39. Zupanic, A., Meplan, C., Grellscheid, S.N., Mathers, J.C., Kirkwood, T.B., Hesketh, J.E., Shanley, D.P.: Detecting translational regulation by change point analysis of ribosome profiling data sets. RNA **20**(10), 1507–1518 (2014)

Multitask Matrix Completion for Learning Protein Interactions Across Diseases

Meghana Kshirsagar[1(✉)], Jaime G. Carbonell[2], Judith Klein-Seetharaman[3], and Keerthiram Murugesan[2]

[1] IBM T. J. Watson Research, Yorktown Heights, NY 10598, USA
mkshirs@us.ibm.com
[2] Language Technologies Institute, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213, USA
{jgc,kmuruges}@cs.cmu.edu
[3] Metabolic and Vascular Health, Warwick Medical School, University of Warwick, Coventry, UK
judithks@cs.cmu.edu

Abstract. Disease causing pathogens such as viruses, introduce their proteins into the host cells where they interact with the host's proteins enabling the virus to replicate inside the host. These interactions between pathogen and host proteins are key to understanding infectious diseases. Often multiple diseases involve phylogenetically related or biologically similar pathogens. Here we present a multitask learning method to jointly model interactions between human proteins and three different, but related viruses: *Hepatitis C*, *Ebola* virus and *Influenza A*. Our multitask matrix completion based model uses a shared low-rank structure in addition to a task-specific sparse structure to incorporate the various interactions. We obtain upto a 39 % improvement in predictive performance over prior state-of-the-art models. We show how our model's parameters can be interpreted to reveal both general and specific interaction-relevant characteristics of the viruses. Our code, data and supplement is available at: http://www.cs.cmu.edu/~mkshirsa/bsl_mtl.

Keywords: Host-pathogen protein interactions · Multitask learning · Matrix completion

1 Introduction

Infectious diseases such as H1N1 influenza, the recent Ebola outbreak and bacterial infections, such as the recurrent *Salmonella* and *E. coli* outbreaks are a major health concern worldwide, causing millions of illnesses and many deaths each year. Key to the infection process are host-pathogen interactions at the molecular level, where pathogen proteins physically bind to human proteins to manipulate important biological processes in the host cell, to evade the host's immune response and to multiply within the host. Very little is known about these protein-protein interactions (PPIs) between pathogen and host proteins for

M. Singh (Ed.): RECOMB 2016, LNBI 9649, pp. 53–64, 2016.
DOI: 10.1007/978-3-319-31957-5_4

any individual disease. However, such PPI data is widely available across several diseases, and the central question in this paper is: *Can we model host-pathogen PPIs better by leveraging data across multiple diseases?* This is of particular interest for lesser known or recently evolved diseases where the data is particularly scarce. Furthermore, it allows us to learn models that generalize better across diseases by modeling global phenomena related to infection.

An elegant way to formulate the interaction prediction problem is via a graph completion based framework, where we have several bipartite graphs over multiple hosts and pathogens as illustrated in supplementary Fig. S1. Nodes in the graphs represent host and pathogen proteins, with edges between them representing interactions (host protein $\xrightarrow{interacts}$ pathogen protein). Given some observed edges (interactions obtained from laboratory based experiments), we wish to predict the other edges in the graphs. Such bipartite graphs arise in a plethora of problems including: recommendation systems (user $\xrightarrow{prefers}$ movie), citation networks (author $\xrightarrow{cites}$ paper), disease-gene networks (gene $\xrightarrow{influences}$ disease) etc. In our problem, each bipartite graph $\mathcal{G}$ can be represented using a matrix M, where the rows correspond to pathogen proteins and columns correspond to host proteins. The matrix entry M_{ij} encodes the edge between pathogen protein i and host protein j from the graph, with $M_{ij} = 1$ for the observed interactions. Thus, the graph completion problem can be mathematically modeled as a matrix completion problem [2].

Most of the prior work on host-pathogen PPI prediction has modeled each bipartite graph separately, and hence cannot exploit the similarities in the edges across the various graphs. Here we present a *multitask* matrix completion method that *jointly models* several bipartite graphs by sharing information across them. From the multitask perspective, a *task* is the graph between one host and one pathogen (can also be seen as interactions relevant to one disease). We focus on the setting where we have a single host species (human) and several related viruses, where we hope to gain from the fact that similar viruses will have similar strategies to infect and hijack biological processes in the human body. Our model is motivated by the following biological intuition governing protein interactions across diseases.

1. An interaction depends on the structural properties of the proteins, which are conserved across similar viruses as they have evolved from common ancestors. We use a component to capture these latent similarities, that is *shared* across tasks.
2. In addition to the shared properties discussed above, each pathogen has also evolved specialized mechanisms to target host proteins. These are unique to the pathogen and can be expressed using a *task-specific* parameter.

This leads us to the following model that incorporates the above ideas. The interactions matrix M_t of task t can be written as: $M_t = \mu_t * (\text{shared component}) + (1 - \mu_t) * (\text{specific component})$, with hyperparameter μ_t allowing each task to customize its' amount of shared and specific components.

To incorporate the above ideas, we assume that the interactions matrix M is generated from two components. The first component has low-rank latent factors

over the human and virus proteins, with these latent factors jointly learned over all tasks. The second component involves a task specific parameter, on which we additionally impose a sparsity constraint as we do not want this parameter to overfit the data. Section 3 discusses our model in detail. We trade-off the relative importance of the two components using task-specific hyperparameters. We can thus learn what is conserved and what is different across pathogens, rather than having to specify it manually.

The applications that we consider involve extremely sparse graphs with a large number of nodes and very few observed edges. There will be nodes i.e., proteins that are not involved in any known interactions – the model should be able to predict links between such prior 'unseen' node pairs (this is called the *cold start problem* in the recommendation systems community). For instance, the host-pathogen PPI network of human-Ebola virus (column-3, Table 1) has ≈ 90 observed edges (equivalent to 0.06 % of the possible edges) which involve only 2 distinct virus proteins. Any biologist studying virus-human interactions will be more interested in the virus proteins which have yet unknown interactions. The main contributions of this work are:

1. We extend the basic matrix decomposition framework from [1] to the multitask setting by incorporating the structure between the tasks and providing a mechanism for the tasks to share information.
2. We leverage node features which allows us to predict on unseen nodes.
3. We apply the model to an important problem – prediction of interactions in disease-relevant host-pathogen protein networks, for multiple related diseases and demonstrate significant gains in performance over prior state-of-the-art multitask models.
4. We use unlabeled data to initialize the parameters of our model, which gives us a modest boost in prediction performance.

1.1 Background: Host-Pathogen Protein Interactions

The experimental discovery of host-pathogen protein interactions involves biochemical and biophysical methods such as co-immunoprecipitation (co-IP), yeast two-hybrid (Y2H) assays, co-crystallization. The most reliable experimental methods are often very time-consuming and expensive, making it hard to investigate the prohibitively large set of possible host-pathogen interactions – e.g., the bacterium *Bacillus anthracis* with about 2321 proteins when coupled with the 100,000 human proteins gives ≈ 232 million protein pairs to validate. Computational techniques complement laboratory-based methods by predicting highly probable PPIs. Supervised machine learning based methods use the few known interactions as training data and formulate the interaction prediction problem in a classification setting.

1.2 Prior Work

Most of the prior work in PPI prediction has focused on building models separately for individual organisms [13,16] or on building a model specific to a

disease in the case of host-pathogen PPI prediction [5,9,17]. There has been limited work on combining PPI datasets to learn joint models. [14] proposed a semi-supervised multi-task framework to predict PPIs from partially labeled reference sets. [10] develop a task regularization based framework that incorporates the similarity in biological pathways targeted by various diseases. [21] uses a collective matrix factorization based approach in a multi-task learning setting for within species PPI prediction. The methods used in all prior work on PPI prediction do not explicitly model the features of the proteins and cannot be applied on proteins which have no known interactions available. Our work addresses both these issues.

2 Bilinear Low-Rank Matrix Decomposition

In this section, we present the matrix decomposition model that we extend for the multitask scenario. In the context of our problem, at a high level, this model states that – protein interactions can be expressed as dot products of features in a lower dimensional subspace.

Let $\mathcal{G}_t$ be a bipartite graph connecting nodes of type υ with nodes of type ς. Let there be m_t nodes of type υ and n_t nodes of type ς. We denote by $M \in \mathbb{R}^{m_t \times n_t}$, the matrix representing the edges in $\mathcal{G}_t$. Let the set of observed edges be Ω. Let $\mathcal{X}$ and $\mathcal{Y}$ be the feature spaces for the node types υ and ς respectively. For the sake of notational convenience we assume that the two feature spaces have the same dimension d_t[1]. Let $\mathbf{x}_i \in \mathcal{X}$ denote the feature vector for a node i of type υ and $\mathbf{y}_j \in \mathcal{Y}$ be the feature vector for node j of type ς. The goal of the general matrix completion problem is to learn a function $f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ that also explains the observed entries in the matrix M. We assume that the function f is bilinear on $\mathcal{X} \times \mathcal{Y}$. This bilinear form was first introduced by [1] and takes the following form:

$$f(\mathbf{x}_i, \mathbf{y}_j) = \mathbf{x}_i^\intercal H \mathbf{y}_j = \mathbf{x}_i^\intercal U V^\intercal \mathbf{y}_j \tag{1}$$

The factor $H \in \mathbb{R}^{d_t \times d_t}$ maps the two feature spaces $\mathcal{X}$ and $\mathcal{Y}$. This model assumes that H has a low-rank factorization given by $H = UV^\intercal$, where $U \in \mathbb{R}^{d_t \times k}$ and $V \in \mathbb{R}^{d_t \times k}$. The factors U and V project the two feature spaces to a common lower-dimensional subspace of dimension k. While the dimensionality of the feature spaces $\mathcal{X}$ and $\mathcal{Y}$ may be very large, the latent lower dimensional subspace is sufficient to capture all the information pertinent to interactions. To predict whether two new nodes (i.e., nodes with no observed edges) with features $\mathbf{p}_i$ and $\mathbf{q}_j$ interact, we simply need to compute the product: $\mathbf{p}_i U V^\intercal \mathbf{q}_j$. This enables the model to avoid the cold start problem that many prior models suffer from. The objective function to learn the parameters of this model has two main terms: (1) a data-fitting term, which imposes a penalty for deviating from the observed entries in Ω and (2) a low-rank enforcing term on the matrix H.

[1] The dimensions being different does not influence the method or the optimization in any way.

The first term can be any loss function such as squared error, logistic-loss, hinge loss. We tried both squared error and logistic-loss and found their behaviour to be similar. The squared error function has the advantage of being amenable to adaptive step-size based optimization which results in a much faster convergence. The low-rank constraint on H (mentioned in (2) above) is NP-hard to solve and it is standard practice to replace it with either the trace-norm or the nuclear norm. Minimizing the trace norm (i.e., sum of singular values) of $H = UV^\intercal$, is equivalent to minimizing $\|U\|_F^2 + \|V\|_F^2$. This choice makes the overall function easier to optimize:

$$\mathcal{L}(U, V) = \sum_{(i,j)\in\Omega} c_{ij}\ \ell(M_{ij}, \mathbf{x}_i^\intercal U V^\intercal \mathbf{y}_j) + \lambda(\|U\|_F^2 + \|V\|_F^2) \tag{2}$$

where $\ell(a, b) = (a - b)^2$

The constant c_{ij} is the weight/cost associated with the edge (i, j) which allows us to penalize the error on individual instances independently. The parameter λ controls the trade-off between the loss term and the regularizer.

3 The Bilinear Sparse Low-Rank Multitask Model (BSL-MTL)

In the previous section, we described the bilinear low-rank model for matrix completion. Note that in order to capture linear functions over the features, we introduce a constant feature for every protein (i.e., $[\mathbf{x}_i 1]$). We now discuss the multitask extensions that we propose. Let $\{\mathcal{G}_t\}$ where $t = 1 \ldots T$ be the set of T bipartite graphs and the corresponding matrices be $\{M_t\}$. Each matrix M_t has rows corresponding to node type υ_t and columns corresponding to the node type ς_t. The feature vectors for individual nodes of the two types be represented by $\mathbf{x}_{ti}$ and $\mathbf{y}_{tj}$ respectively. Let Ω_t be the set of observed links (and non-links) in the graph $\mathcal{G}_t$. Our goal is to learn individual link prediction functions f_t for each graph. In order to exploit the relatedness of the T bipartite graphs, we make some assumptions on how they share information. We assume that each matrix M_t has a low-rank decomposition that is shared across all graphs and a sparse component that is specific to the task t. That is,

$$f_t(\mathbf{x}_{ti}, \mathbf{y}_{tj}) = \mathbf{x}_{ti}^\intercal H_t \mathbf{y}_{tj}, \text{ where } H_t = \mu_t U V^\intercal + (1 - \mu_t) S_t \tag{3}$$

As before, the shared factors U and V are both $\mathbb{R}^{d_t \times k}$ (where the common dimensionality d_t of the two node types is assumed for convenience). The matrix $S_t \in \mathbb{R}^{d_t \times d_t}$ is a sparse matrix. The objective function for the multitask model is given by:

$$\mathcal{L}(U, V, \{S_t\}) = \frac{1}{N} \sum_{t=1}^{T} \sum_{(i,j)\in\Omega_t} c_{ij}^t\ \ell\big(M_{t_{ij}}, \mathbf{x}_{ti}^\intercal H_t \mathbf{y}_{tj}\big)^2 + \lambda(\|U\|_F^2 + \|V\|_F^2) + \sum_{t=1}^{T} \sigma_t \|S_t\|_1 \tag{4}$$

Here $N = \sum_t |\Omega_t|$, is the total number of training examples (links and non-links included) from all tasks. To enforce the sparsity of S_t we apply an ℓ_1 norm. In our experiments, we tried both ℓ_1 and ℓ_2 norms and found that the ℓ_1 norm works better.

Optimization: The function $\mathcal{L}(U, V, \{S_t\})$ is non-convex. However, it is convex in every one of the parameters (i.e., when the other parameters are fixed) and a block coordinate descent method called alternating least squares (ALS) is commonly used to optimize such functions. To speed up convergence we use an adaptive step size.

Convergence: The ALS algorithm is guaranteed to converge only to a local minimum. There is work showing convergence guarantees to global optima for related simpler problems, however the assumptions on the matrix and the parameter structure are not very practical and it is difficult to verify whether they hold for our setting.

Initialization of U and V: We tried random initialization (where we randomly set the values to lie in the range [0 1]), and also the following strategies that initialize: $U^0 \leftarrow$ top-k left singular vectors, and $V^0 \leftarrow$ top-k right singular vectors from the SVD of $\sum_{(i,j)\in\Gamma} \mathbf{x}_i \mathbf{y}_j^\intercal$. We set Γ to (a) training examples from all tasks, or (b) a random sample of 10000 unlabeled data from all tasks. We found that using the unlabeled data for initialization gives us a better performance.

3.1 Handling the 'Curse of Missing Negatives'

For the MC algorithm to work in practice the matrix entries M_{ij} should represent interaction scores (range [0 1]) or take binary values (1s for positives and 0s for negatives). Our experiments with PPI probabilities (obtained using the MINT-scoring algorithm) gave bad models. The binary matrix setting requires some observed 0s. However non-interactions are not available as they cannot be verified experimentally for various reasons. Please refer to the supplementary Sect. S1 for details.

4 Experimental Setup

4.1 Datasets and Features

We use three human-virus PPI datasets from the PHISTO [18] database (version from 2014), the characteristics of which are summarized in Table 1. The *Influenza A* task includes various strains of flu: H1N1, H3N2, H5N1, H7N3. Similarly, the *Hepatitis* task includes various strains of the virus.[2] All three are single-strand RNA viruses, with *Hepatitis* being a positive-strand ssRNA

[2] Since we use data from several strains for each task, the PPI data contains some interactions that are interologs. Please see the supplementary Sect. S4 for details.

Table 1. Tasks and their sizes. Each column corresponds to one bipartite graph between human proteins and the pathogen indicated in the column header. All pathogens are single stranded RNA viruses. The interactions and the protein count both includes data across various strains of each pathogen

Task →	*Influenza A*	*Hepatitis C*	*Ebola*
Number of HP PPIs (positives)	848	981	90
# of distinct virus proteins in PPIs	54	151	2
# of distinct human proteins in PPIs	362	385	88
Total # of virus proteins across strains	542	163	150
Number of negatives	84800	98100	9000
Density of observed graph‡ (as %)	.15	.60	.06

HP PPI: host-pathogen protein-protein interactions
‡: considering all proteins from the two tasks involved
Note: considering the total number of human proteins to be $\approx 100,000$

whereas *Influenza* and *Ebola* are negative-strand viruses. The density of the known interactions is quite small when considering the entire proteome (i.e., all known proteins) of the host and pathogen species (last row in Table 1).

Features: Since the sequence of a protein determines its structure and consequently its function, it may be possible to predict PPIs using the amino acid sequence of a protein pair. [15] introduced the "conjoint triad model" for predicting PPIs using only amino acid sequences. They partitioned the twenty amino acids into seven classes based on their electrostatic and water affinities.[3] A protein's amino acid sequence is first transformed to a class-sequence (by replacing each amino acid by its class). For k=3, they count the number of times each distinct trimer (set of three consecutive amino acids) occurred in the sequence. Since there are 343 (7^3) possible trimers (with an alphabet of size 7), the feature vector containing the trimer frequency counts will have 343 elements. To account for protein size, they normalized the counts by linearly transforming them to lie between 0 and 1. Thus the value of each feature in the feature vector is the normalized count for each of the possible amino acid three-mers. We use di-, tri- and four-mers thus leading to a total of 2793 features ($7^2 + 7^3 + 7^4$). Such features have been successfully applied in prior work [5,10].

4.2 Competing Methods

We compare BSL-MTL to various single-task and multitask methods, which includes conventional multitask methods and other recent low-rank and sparse models, and prior work on HP PPI prediction. Wherever appropriate, we concatenated the features of the two node types into a single feature vector. Let $W \in \mathbb{R}^{T \times d_t}$ be the matrix with the task-specific weight vectors $\mathbf{w}_t$. For a uniform comparison we used least squared loss in all the methods. The MALSAR

[3] For details of these classes, please refer to the supplementary or the original paper.

package was used for some of the baselines. Refer to supplementary Sect. S2 for parameter tuning.

Single task (STL): Ridge regression with ℓ_2 regularization.
MMTL: The mean regularized multitask learning model from [6].
Low rank model (TraceNorm): A low-rank structure is enforced on W by minimizing the nuclear norm $\|W\|_*$.
Sparse + low-rank (SpLowRank) [3]: W is assumed to have the decomposition: $W = P + Q$, where P is sparse and Q has a low-rank structure.
IMC [8,12]: This is the link-prediction model from Sect. 2, where data from all tasks is combined without incorporating any task relationships (comparable to the 'union' setting from [20]). U and V are shared by all tasks. We use the same initialization for this method as we do for our model. A comparison to this model tells us how much we gain from the task-specific sparsity component S_t.
MTPL [10]: A biologically inspired regularizer is used to capture task similarity.
BSL-MTL: This work, Bilinear sparse low-rank multitask learning.

Table 2. Area Under the Precision-Recall curve for each task in the two settings. X% training indicates the fraction of the labeled data used for training and tuning the model with the rest (100-X)% used as test data. We report the average AUC-PR over 10 random train-test splits (stratified splits that maintain the class-skew of 1:100). The standard deviation is also shown. The performance of the best baseline and the overall best method (BSL-MTL) is highlighted in bold. The first row is the only single-task method and all others are multitask models.

	10 % training			30 % training		
	Ebola	*Hep-C*	*Influenza*	*Ebola*	*Hep-C*	*Influenza*
STL (Ridge Reg.)	0.189±.09	0.702±.08	0.286±.02	0.220±.03	0.802±.03	0.428±.03
MMTL [6]	0.113±.04	0.767±.03	0.321±.02	0.129±.02	0.802±.04	0.430±.03
Trace-norm	0.199±.11	0.767±.03	0.318±.02	0.207±.02	0.808±.02	0.409±.03
SpLowRank [3]	0.144±.07	0.767±.02	0.318±.02	0.153±.02	**0.814**±.01	0.414±.03
MTPL [10]	**0.217**±.08	0.695±.02	0.345±.02	**0.260**±.05	0.713±.01	**0.496**±.03
IMC [12]	0.087±.04	**0.779**±.02	**0.362**±.01	0.122±.02	0.801±.01	0.410±.03
BSL-MTL	**0.233**±.10	**0.807**±.02	**0.486**±.02	**0.361**±.03	**0.842**±.01	**0.560**±.02

4.3 Evaluation Setup

We first compare all the methods in two settings, where a small proportion of the available labeled data is randomly sampled and used to train a model which is then evaluated on the remaining data. For the first setting we randomly split the labeled data from each task into 10 % training and 90 % test, such that the class-skew of 1:100 is maintained in both splits (i.e., stratified splits).

The second setting uses a 30 % training, 70 % test split. We use identical splits for all algorithms. In each setting we generate ten random splits and average the performance over the ten runs. Next, we do a standard 10-fold cross validation (CV) experiment (8 folds to train, 1 fold as held-out, 1 fold as test data). In this setting, each algorithm has access to a much larger training set but a significantly smaller test set. The two prior settings (10 % and 30 %) portray a more realistic multitask scenario where we have access to little training data from each task.

We report the area under the precision recall curve (AUC-PR) along with the standard deviation. AUC-PR has been shown to give a more informative picture of an algorithm's performance than ROC curves in high class imbalance datasets [4] such as ours.

5 Results

Table 2 has the AUC-PR for all methods. Note that the AUC-PR of a random classifier model is ≈ 0.01. The first row (STL) is the single-task baseline and all others are multitask models. In general, we notice that multitask learning benefits all tasks. The first three columns show the results in the 10 % setting. Our model (last row) has significant gains for *Influenza* (1.3 times better than the next best) and modest improvements for the other tasks. The variance in the performance is high for the Ebola task (column 1) owing to the small number of positives in the training splits (8 positives). The most benefits for our model are seen in the 30 % setting for all tasks, with improvements of 39 %, 3 % and 12 % on the *Ebola*, *Hepatitis* and *Influenza* tasks, respectively. *Ebola*, the data-poorest task, benefits the most. 10 fold CV results are in the supplementary Sect. S3.

5.1 Biological Significance of the Model

The model parameters U, V and S are a source of rich information which can be used to further understand host-pathogen interactions. Note that our features are derived from the amino acid sequences of the proteins which provide opportunities to interpret the parameters.

Clustering Proteins Based on Interaction Propensities. We analyze the proteins by projecting them using the model parameters U and V into a lower dimensional subspace (i.e., computing $XU^\intercal$ and $YV^\intercal$ to get projections of the virus and human proteins respectively). The principal component analysis (PCA) of this lower dimensional representation is compared with PCA in the original feature space (protein sequence features) in Fig. 1. Firstly, the projected data has a much better separation than the original data. Secondly, Fig. 1 (right) tells us that Hepatitis-C and Influenza have many proteins with similar binding tendencies, and that these behave differently than most Ebola virus proteins. This observation is not obvious in the PCA of the original feature space (Fig. 1 left), where proteins with similar sequences cluster together. These clusters of proteins can be analyzed further for enrichment of Gene Ontology (GO) annotations.

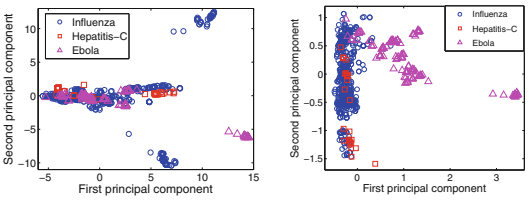


Fig. 1. First two components from Principal component analysis (PCA) of virus proteins. *Left*: PCA of original feature space. *Right*: PCA of projected subspace. Shape of the points indicates the virus the protein belongs to.

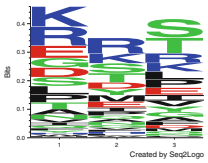


Fig. 2. Top trimer sequence motifs from virus proteins of all viruses.

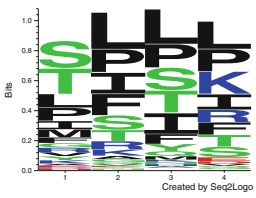


Fig. 3. Sequence motif constructed from the top four-mer features of virus proteins across all three viruses.

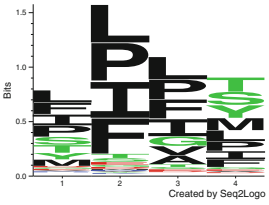


Fig. 4. Sequence motif constructed from the top four-mer features of human proteins

Sequence Motifs from Virus Proteins. In Figs. 2, 3 and 4, we show sequence motifs derived from the top k-mers that contribute to interactions. The significant entries of the model parameters U, V and $\{S_t\}$ were used to compute these motifs. The top positive-valued entries from the product UV^T indicate which pairs of features: $((f_v, f_h)$: virus protein feature, human protein feature) are important for interactions across all the virus-human PPI tasks. Analogously, the entries from S_t give us pairs of features important to a particular virus-human task 't'. We find that most of the top entries from UV^T correspond to linear virus features, whereas those from the various S_t involve bilinear features. We analyze the k-mers corresponding to the top 20 features from each of the matrices.

Note that our features do not directly correspond to a unique amino-acid k-mer (see Sect. 4.1): the virus feature f_v will map to several amino-acid sequences (for instance KKCC, KRCC, RRCC etc. all map to a single feature due to the molecular similarity between the amino acids K and R being both positively charged). Given the set of top virus features we can obtain the corresponding set of amino-acid k-mers, say AA_v, by reversing the feature-generation step. However most of the possible k-mers do not appear in the training data (ex: out of the 160,000 ($= 20^4$) possible 4-mers $\approx 24,000$ appear). Let AA_{tr} be the set of amino-acid k-mers that appear in the training data. Then, the intersection $I_v = AA_v \cap AA_{tr}$ gives us the important amino-acid k-mers from

virus proteins w.r.t interaction prediction. To summarize I_v, we use a popular tool Seq2Logo [19] to generate a sequence motif. The logos for the two-, three-, four-mers from I_v are generated independently. Since we only want to summarize, we use the Shannon logo type (which does not consider any background amino-acid distribution) with the following settings: clustering-method = None, weight-on-prior = 1 (pseudo-counts do not make sense in our analysis). Figures 2 and 3 show the motif that is common across viruses.

This procedure described above is used to analyze the most significant human protein features, obtained from the matrix UV^T, which are shown in Fig. 4. We observe that the shared trimer motif for virus proteins in Fig. 2 is dominated by hydrophilic amino acids (K, R, T, D, E). All other motifs seem to be dominated by hydrophobic residues (I, P, L, V, A, G) though S and T do appear in some motifs as well. The human protein motifs are shown in Fig. 4. Further analysis with trimers and tetramers specific to the pathogens is in the supplementary, Sects. S5 and S6. These task-specific features (i.e., k-mers) are obtained from the matrices S_{ebola}, S_{hepc} and S_{flu} respectively. In most cases, the first position of the trimer was less significant than the second and third, while for the tetramer all four positions show clear preferences.

Phosphorylation sites: We found the frequent occurrence of S and T and sometimes Y in the motifs striking and suspected this may be related to the amino acids being frequent targets of phosphorylation. Phosphorylated sites often serve as PPI sites, and databases such as Phosphosite [7] are repositories for known sites in human proteins. Since these are sites in human proteins, we searched for the patterns from the 4-mer motif in Fig. 4 and found several to be flanking known phosphorylation sites in human proteins: `LLLs`, `LLLt`, `ILLs`, `PPPs`, `PIPs`, `PIPt`, `LIPs`, `PLLt` (lower-case indicates the putative phosphorylation site). This observation also supports the notion that the motifs our method finds are biologically interesting.

Novel Interactions with Ebola Proteins. The top four Ebola-human PPI are all predictions for the Ebola envelope glycoprotein (GP) with four different human proteins (Note: GP is not in the gold standard PPIs). There is abundant evidence in the published literature [11] for the critical role played by GP in virus docking and fusion with the host cell. A list of interactions will be provided on the corresponding authors website.

References

1. Abernethy, J., Bach, F., Evgeniou, T., Vert, J.P.: A new approach to collaborative filtering: operator estimation with spectral regularization. J. Mach. Learn. Res. (JMLR) **10**, 803–826 (2009)
2. Candes, E., Recht, B.: Exact matrix completion via convex optimization. Found. Comput. Math. **9**(6), 717–772 (2008)
3. Chen, J., Liu, J., Ye, J.: Learning incoherent sparse and low-rank patterns from multiple tasks. ACM Trans. Knowl. Discov. Data (TKDD) **5**(4), 22 (2012)

4. Davis, J., Goadrich, M.: The relationship between precision-recall and roc curves. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 233–240 (2006)
5. Dyer, M.D., Murali, T.M., Sobral, B.W.: Computational prediction of host-pathogen protein-protein interactions. Bioinformatics **23**(13), i159–166 (2007)
6. Evgeniou, T., Pontil, M.: Regularized multi-task learning. In: ACM SIGKDD (2004)
7. Hornbeck, P.V., Zhang, B., Murray, B., Kornhauser, J.M., Latham, V., Skrzypek, E.: Phosphositeplus, 2014: mutations, ptms and recalibrations. Nucleic Acids Res. **43**(D1), D512–D520 (2015)
8. Jain, P., Dhillon, I.S.: Provable inductive matrix completion (2013). arXiv:1306.0626
9. Kshirsagar, M., Carbonell, J.G., Klein-Seetharaman, J.: Techniques to cope with missing data in host-pathogen protein interaction prediction. Bioinformatics **28**(18), i466–i472 (2012)
10. Kshirsagar, M., Carbonell, J.G., Klein-Seetharaman, J.: Multi-task learning for host-pathogen protein interactions. Bioinformatics **29**(13), i217–i226 (2013)
11. Nanbo, A., Imai, M., Watanabe, S., et al.: Ebolavirus is internalized into host cells via macropinocytosis in a viral glycoprotein-dependent manner. PLoS Pathog. **6**(9), e1001121 (2010)
12. Natarajan, N., Dhillon, I.S.: Inductive matrix completion for predicting genedisease associations. Bioinformatics **30**(12), i60–i68 (2014)
13. Qi, Y., et al.: Evaluation of different biological data and computational classification methods for use in protein interaction prediction. Proteins **63**(3), 490–500 (2006)
14. Qi, Y., Tastan, O., Carbonell, J.G., Klein-Seetharaman, J., Weston, J.: Semi-supervised multi-task learning for predicting interactions between HIV-1 and human proteins. Bioinformatics **6**(18), i645–i652 (2010)
15. Shen, J., et al.: Predicting protein-protein interactions based only on sequences information. PNAS **104**, 4337–4341 (2007)
16. Singh, R., Xu, J., Berger, B.: Struct2net: integrating structure into protein-protein interaction prediction. Pac. Symp. Biocomput. **11**, 403–414 (2006)
17. Tastan, O., et al.: Prediction of interactions between HIV-1 and human proteins by information integration. Pac. Symp. Biocomput. **14**, 516–527 (2009)
18. Tekir, S.D., Ali, S., Tunahan, C., Kutlu, O.U.: Infection strategies of bacterial and viral pathogens through pathogen-host protein protein interactions. Front. Microbio. Immunol. **3**, 46 (2012)
19. Thomsen, M.C.F., Nielsen, M.: Seq2logo: a method for construction and visualization of amino acid binding motifs and sequence profiles including sequence weighting, pseudo counts and two-sided representation of amino acid enrichment and depletion. Nucleic Acids Res. **40**(W1), W281–W287 (2012)
20. Widmer, C., Leiva, J., Altun, Y., Rätsch, G.: Leveraging sequence classification by taxonomy-based multitask learning. In: Berger, B. (ed.) RECOMB 2010. LNCS, vol. 6044, pp. 522–534. Springer, Heidelberg (2010)
21. Xu, Q., Xiang, W.E., Yang, Q.: Protein-protein interaction prediction via collective matrix factorization. In: International Conference on Bioinformatics and Biomedicine (2010)

pathTiMEx: Joint Inference of Mutually Exclusive Cancer Pathways and Their Dependencies in Tumor Progression

Simona Cristea[1,2], Jack Kuipers[1,2], and Niko Beerenwinkel[1,2(✉)]

[1] Department of Biosystems Science and Engineering, ETH Zürich, Basel, Switzerland
niko.beerenwinkel@bsse.ethz.ch
[2] Swiss Institute of Bioinformatics, Basel, Switzerland

Abstract. In recent years, high-throughput sequencing technologies have facilitated the generation of an unprecedented amount of genomic cancer data, opening the way to a more profound understanding of tumorigenesis. In this regard, two fundamental questions have emerged: (1) which alterations drive tumor progression? and (2) what are the evolutionary constraints on the order in which these alterations occur? Answering these questions is crucial for therapeutic decisions involving targeted agents, which are often based on the identification of early genetic events. Mainly because of interpatient heterogeneity, progression at the level of pathways has been shown to be more robust than progression at the level of single genes. Here, we introduce pathTiMEx, a probabilistic generative model of tumor progression at the level of mutually exclusive driver pathways. pathTiMEx employs a stochastic optimization procedure to jointly optimize the assignment of genes to pathways and the evolutionary order constraints among pathways. On cancer data, pathTiMEx recapitulates previous knowledge on tumorigenesis, such as the temporal order among pathways which include *APC*, *KRAS* and *TP53* in colorectal cancer, while also proposing new biological hypotheses, such as the existence of a single early causal event consisting of the amplification of *CDK4* and the deletion of *CDKN2A* in glioblastoma. The pathTiMEx R package is available at https://github.com/cbg-ethz/pathTiMEx. Supplementary Material for this article is available online.

1 Introduction

Over the last years, our basic understanding of tumorigenesis has increased substantially, primarily as a result of the large-scale use of high-throughput sequencing technologies. High-resolution genomic, epigenomic, transcriptomic and proteomic information from tens of thousands of cancerous samples are now publicly available [7–9], providing the medical and research communities with an

Electronic supplementary material The online version of this chapter (doi:10.1007/978-3-319-31957-5_5) contains supplementary material, which is available to authorized users.

M. Singh (Ed.): RECOMB 2016, LNBI 9649, pp. 65–82, 2016.
DOI: 10.1007/978-3-319-31957-5_5

unprecedented amount of genetic data. In this context, two fundamental questions have emerged in describing tumorigenesis [34]: (1) which alterations drive tumor progression? and (2) what are the evolutionary constraints on the order in which these alterations occur? Answering these questions is crucial not only for therapeutic decisions, but also for the basic understanding of carcinogenesis. Early genetic events, such as point mutations or copy number aberrations, have a particularly important role in tumor development, as they may induce the so-called *oncogenic addiction* [29,42], when some tumors rely on one single dominant oncogene for growth and survival [39]. The identification of these events can prioritize the validation of candidate drug targets.

In the context of identifying important events in carcinogenesis, genes which have a positive selective advantage and significantly contribute to tumor progression are termed *drivers*, while genes which are selectively neutral are termed *passengers*. Intuitively, drivers and passengers can be classified according to their marginal alteration frequencies in a cohort of patients. However, the highly diverse alteration landscape characteristic of malignant tumors is often poorly explained solely by the highly frequently altered genes [37]. A more sensitive and robust identification of driver events consists in analyzing the joint effect of multiple alterations performing the same functional role in tumor progression, commonly referred to as *pathways*. Once one of the members of a pathway is altered, tumors gain a significant selective advantage, which is not further increased by the alteration of additional pathway members. The clones with only the first alteration likely become dominant, rendering all alterations in the same pathway to display a mutually exclusive pattern. Therefore, in tumorigenesis, identifying groups of mutually exclusive genes can lead to identifying driver events.

Several computational approaches to identify mutually exclusive pathways either *de novo* [12,22,23,26,38,40,44] or based on literature-derived biological interaction networks [2,11] have recently been developed. Multidendrix [26] employs integer linear programming to find multiple driver pathways, improving on the tool Dendrix [40], which identifies a single pathway with both high coverage and high exclusivity. CoMEt [44] further improves the methodology of Multidendrix by proposing a generalization of Fisher's exact test for evaluating mutual exclusivity. Muex [38] is a statistical model in which members of mutually exclusive groups are required to have similar alteration frequencies, MEMCover [23] detects dysregulated mutually exclusive subnetworks across different cancer types, and Daisy [22] employs a data-driven approach for identifying synthetic lethality interactions in cancer, which lead to mutual exclusivity. Finally, TiMEx [12] is a generative probabilistic model which quantifies exactly the degree of mutual exclusivity for each identified pathway. TiMEx explicitly models tumorigenesis as a dynamic process, accounting for the interplay between the waiting times to alteration of each gene and the observation time. Except TiMEx, all other methods ignore the fact that mutually exclusive patterns occur over time, during disease progression.

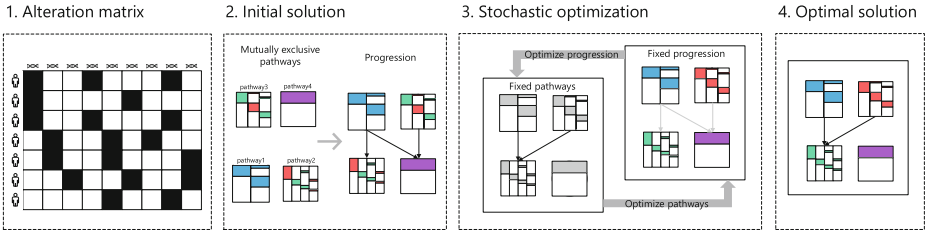


Fig. 1. Overview of pathTiMEx. In the first step, a cancer genomics dataset (consisting of, for example, point mutations or copy number aberrations for a cohort of patients) is preprocessed in the form of a binary alteration matrix, with rows representing patients and columns representing alterations. A black square encodes the presence of an alteration, and a white square encodes its absence. In the second step, on the basis of the binary matrix, mutually exclusive groups of alterations (pathways) are inferred with TiMEx [12], and the progression among pathways is inferred with CBN [19]. In the third step, the progression among pathways is used as the initial solution to a stochastic optimization routine, which consists of two parts. First, given the fixed optimal progression among pathways, the assignment of genes to pathways is optimized via a Markov Chain Monte Carlo (MCMC) approach [30]. Second, given the fixed optimal assignment of genes to pathways, the progression among pathways is optimized via simulated annealing [24]. The joint optimization is repeated until both the assignment of genes to pathways and the progression among pathways converge to the joint optimal solution.

In the context of evolutionary order constraints in cancer (see [4,15] for recent reviews on progression models), Fearon and Vogelstein [18] were the first to show that not all progression paths are equally probable. Based on genetic and clinical data, their model proposed the existence of a linear order among several mutations in colorectal cancer. Next, oncogenetic trees [14] generalized the idea of a single progression path, by allowing diverging temporal orderings of events, under the assumption that each event depends on a single parent. At the cost of increased computations, Bayesian Networks for cancer progression [3,5,19,21,35] addressed the restriction of tree-based methods of not allowing different progression paths to converge. For example, Conjunctive Bayesian Networks (CBN) [5,19] are generative models of tumor progression, defined by a partial order on a set of events, and in which the occurrence of each event can depend on more than one parent. Other approaches include RESIC [1], which explicitly considers the evolutionary dynamics of mutation accumulation, Progression Networks [16], which employ mixed integer linear programming, and CAPRESE [28] and CAPRI [32], which use a framework of probability causation.

Virtually all cancer progression models are designed to infer tumorigenesis using only *cross-sectional* data, *i.e.* single-time snapshots from multiple patients. However, because of the very high interpatient heterogeneity [41], carcinogenesis often follows different progression paths in different individuals. Even though the selective pressure in cancer is known to be stronger on pathways than on single

alterations [20,41], only three previous models describe tumorigenesis at the level of pathways [10,20,34]. Commonly, in a first step, single alterations are mapped to literature-derived pathways, and, in a second step, progression algorithms are used to infer the order constraints among pathways [10,20]. Besides the inherent disadvantages of using literature-derived pathways, such as lack of specificity and large pathway size, these approaches are assuming that the inference of either driver pathways or order constraints are independent of eachother. On the contrary, Raphael and Vandin [34] have formally shown that the single *a priori* knowledge of either pathways or progression is not sufficient for inferring the correct joint solution. However, their joint integer linear programming model of progression among mutually exclusive pathways assumes that cancer progresses via a single linear path, which is a very restricted representation of tumorigenesis.

Here, we introduce pathTiMEx, a probabilistic generative model of cancer progression at the level of driver pathways (Fig. 1). pathTiMEx directly generalizes TiMEx [12], a waiting time model for mutually exclusive cancer alterations, and CBN [19], a waiting time model for cancer progression at the level of single genes (Fig. 2). We assume that, in tumor development, alterations can either occur independently, or depend on eachother by being part of the same pathway or by following particular progression paths. By inferring these two types of potential dependencies simultaneously, pathTiMEx jointly identifies driver events and the evolutionary order constraints among them. In our approach, the structure among pathways is modeled as a directed acyclic graph (DAG), hence the model of Raphael and Vandin [34] is a special case of pathTiMEx, corresponding to the situation when the structure is fixed to a linear path.

On publicly available cancer data, pathTiMEx recapitulates previous knowledge on tumorigenesis, such as the temporal order among pathways which include *APC*, *KRAS* and *TP53* in colorectal cancer, while also proposing new biological hypotheses, such as the existence of a single early causal event consisting of the amplification of *CDK4* and the deletion of *CDKN2A* in glioblastoma. Our approach represents the first joint probabilistic generative model of cancer progression via multiple paths, at the level of mutually exclusive driver pathways.

2 Methods

2.1 Probabilistic Model

Preliminaries. Let $\mathcal{G} = \{1, \ldots, n\}$ and $\mathcal{P} = \{1, \ldots, k\}$ be two sets of genetic events, with associated random variables $\mathcal{T}_\mathcal{G} = (T_1, \ldots, T_n)$ and $\mathcal{U}_\mathcal{P} = (U_1, \ldots, U_k)$, where T_g represents the waiting time to occurrence of event $g \in \mathcal{G}$ and U_p represents the waiting time to occurrence of event $p \in \mathcal{P}$. Given $\mathcal{G}$, let the vector of random variables $\pi = (\pi_1, \ldots, \pi_n)$ be a partition of $\mathcal{G}$. In the context of tumor evolution, $\mathcal{G}$ is a set of n genes, and $\mathcal{P}$ is a set of k mutually exclusive pathways, to which the n genes are assigned via π. Given π, let $(\mathcal{P}, \prec)$ be a partially ordered set (*poset*), where $\prec$ is a reflexive, antisymmetric and transitive order relation on the events in $\mathcal{P}$. If two events $i, j \in \mathcal{P}$ are related as $i \prec j$, then event i occurs before or at the same time as event j,

which directly corresponds to an order restriction between their corresponding waiting times: $U_i < U_j$. We define the set of parents of any event $p \in \mathcal{P}$ to be $\text{pa}(p) := \{j \in \mathcal{P} \mid j \prec p \text{ and } \nexists\, k \in \mathcal{P} \text{ s.t. } k \neq p, j \text{ and } j \prec k \prec p\}$. We assume that each event can only occur once all its parents have occurred, and that the occurrence of an event represents the irreversible fixation of its alteration, referred to as *mutation*. Each tumor progresses from T_0, the onset of cancer, corresponding to the occurrence of the first cellular signal related to the growth of a malignant tumor, until T_{obs}, the observation time, corresponding to the time of tumor biopsy. Without loss of generality, T_0 is set to 0.

Generative Model. We assume that genes are uniformly assigned to pathways, subject to the restrictions that π is a partition and that the poset $(\mathcal{P}, \prec)$ is uniformly distributed over the space of all transitively reduced Bayesian Networks of size k, denoted here by $\mathbb{D}_k$. T_{obs} is regarded as a system failure time, hence it follows an exponential distribution with an unknown rate: $T_{\text{obs}} \sim \text{Exp}(\lambda_{\text{obs}})$. We extend $(\mathcal{P}, \prec)$ to include order relations among the events in $\mathcal{P}$ and the observation event *obs*, and we denote the extended poset by $(\mathcal{P}_{\text{obs}}, \prec)$, where $P_{\text{obs}} = \{\mathcal{P} \cup obs\}$.

The genes in the same pathway are assumed to contribute to the same biological function, such that, up to various degrees of mutual exclusivity, only one gene is necessary and sufficient to be mutated for cancer to progress. The mutual exclusivity interaction among the members of each pathway drives tumor progression via a mutation in the gene with the shortest waiting time. Therefore, the waiting time of pathway p is the minimum waiting time of its gene members: $U_p := \min_{g:\pi_g=p} T_g$. Once all the parents of pathway p have occurred, the waiting times of its members follow exponential distributions with rates equal to the rates of evolution specific to each gene. Hence, starting from the initial time T_0, the waiting time of gene g is $T_g \sim \max_{j \in \text{pa}(\pi_g)} U_j + \text{Exp}(\lambda_g)$, which is equivalent to $T_g \sim \max_{j \in \text{pa}(\pi_g)} \min_{k:\pi_k=j} T_k + \text{Exp}(\lambda_g)$. Following the mathematical framework of TiMEx [12], the parameter μ_p represents the degree of mutual exclusivity of pathway p and it is modeled as the fractional increase in fixation probability of the clones in which only the gene with the shortest waiting time mutates, provided that its waiting time is also shorter than the observation time. With probability μ_p, alterations in additional pathway members do not fixate before observation time. Hence, μ_p is the probability that p is perfectly mutually exclusive, *i.e.* at most one gene in p is mutated. Consequently, $1 - \mu_p$ quantifies deviations from perfect mutual exclusivity, such that, with probability $1 - \mu_p$, the temporal dynamics of the gene members of p are independent, conditioned on the observation time. In this case, a gene is mutated if and only if its waiting time is shorter than the observation time.

The true mutational statuses of the n genes are represented by the vector of binary random variables $\mathcal{X} = (X_1, \ldots, X_n)$. Any instantiation of $\mathcal{X}$, namely $x = (x_1, \ldots, x_n)$ is referred to as a *true genotype* and can be alternatively represented as a collection of sets $s = \{s^1, \ldots, s^k\}$, where each set $s^p := \{g \mid$

$\pi_g = p$ and $x_g = 1\}$, $\forall p \in \mathcal{P}$ consists of the genes mutated in pathway p. The mutational status of any gene g depends on its own waiting time, the waiting time of pathway π_g, and the observation time. Therefore, with probability μ_{π_g}, $X_g = \begin{cases} 1, & \text{if } U_{\pi_g} = T_g \text{ and } T_g < T_{\text{obs}} \\ 0, & \text{otherwise} \end{cases}$ and, with probability $1 - \mu_{\pi_g}$, $X_g = \begin{cases} 1, & \text{if } T_g < T_{\text{obs}} \\ 0, & \text{otherwise} \end{cases}$. Due to both biological and experimental noise, it can either happen that a mutation present in a gene is not observed (false negative), or that a gene is incorrectly labeled as mutated (false positive). We denote the *observed genotype* by the vector of random variables $Y = (Y_1, \ldots, Y_n)$, where each Y_g represents the observed mutational status of gene $g \in \mathcal{G}$. Y is generated from $\mathcal{X}$ by drawing from a Bernoulli distribution with $P(Y_g = X_g) = 1 - \varepsilon$, where $\varepsilon \in [0, 1)$.

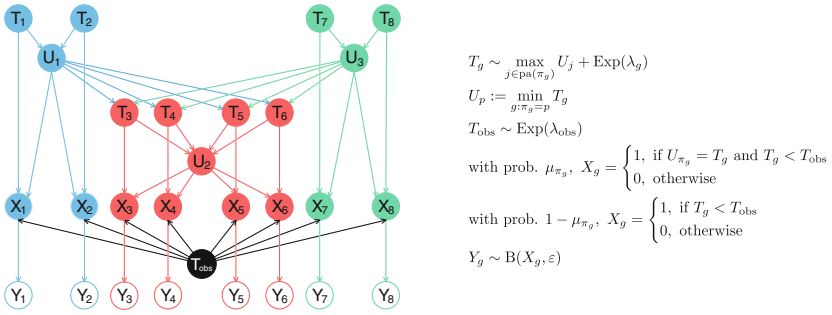


Fig. 2. The probabilistic graphical model pathTiMEx. Nodes represent variables and directed edges represent dependencies. The hidden variables are colored, while the observed ones are not, and the three colors correspond to three different pathways. Variables T_g are the waiting times to alteration of the genes, U_p are the waiting times to alteration of the pathways, T_{obs} is the observation time, X_g are the true mutational statuses of the genes, and Y_g are the observed ones. The corresponding generative distributions are depicted alongside. Additionally, the assignment of genes to pathways $\pi \sim \mathcal{U}\{\mathcal{P}\}$, where $\mathcal{P}$ is the set of k mutually exclusive pathways, the poset $(\mathcal{P}, \prec) \sim \mathbb{D}_k$, the intensity of mutual exclusivity of pathways $\mu_p \sim \mathcal{U}[0, 1]$, and the error probability per gene $\varepsilon \sim \mathcal{U}[0, 1)$. By $\mathbb{D}_k$, we denote the set of transitively reduced DAGs with k vertices.

Consequently, the pathTiMEx model θ (Fig. 2) is characterized by the poset $(\mathcal{P}_{\text{obs}}, \prec)$, together with $2n + k + 2$ additional parameters: $\theta = ((\mathcal{P}_{\text{obs}}, \prec), \lambda_1, \ldots, \lambda_n, \lambda_{\text{obs}}, \pi_1, \ldots, \pi_n, \mu_1, \ldots, \mu_k, \varepsilon)$, where $\lambda_i > 0$ are the exponential waiting time rates of the events in $\mathcal{G} \cup obs$, $\pi_g \in \mathcal{P}$ are the assignments of genes to pathways, $\mu_p \in [0, 1]$ are the degrees of mutual exclusivity of the pathways, and $\varepsilon \in [0, 1)$ is the probability of an erroneous observation at the level of genes. The set of random variables $\{\mathcal{T}_{\mathcal{G}}, U_{\mathcal{P}}, T_{\text{obs}}, \mathcal{X}, Y\}$, together with

the dependencies among its elements, form a Bayesian Network. pathTiMEx is a direct generalization of both TiMEx [12], which only models independent mutually exclusive pathways, and CBN [19], which only models progression among single genes. Similarly to the two models, pathTiMEx is unidentifiable up to λ_{obs}, since cross-sectional input datasets lack temporal information. After setting $\lambda_{\text{obs}} = 1$ (without loss of generality), equivalent to scaling the waiting time rates by λ_{obs}, the reparametrized model becomes identifiable.

μ_p, the degree of mutual exclusivity of pathway p, together with ε, the error probability per gene, represent different ways of capturing the effects of biological and technical noise acting either on each pathway or on their progression. In order to remove the redundancy of modeling two non-independent error processes, we assume that all pathways are perfectly mutually exclusive and that deviations from both mutual exclusivity and progression constraints are jointly explained by ε. In the remainder of this paper, we consider that $\mu_p = 1,\ \forall p \in \mathcal{P}$. The general expression of the likelihood when $\mu_p \in [0, 1]$ is given in Sect. S1.1.

Likelihood. The joint probability density of the waiting times is

$$\begin{aligned} f_{T_{\mathcal{G}},U_{\mathcal{P}},T_{\text{obs}}}\left(t_{\mathcal{G}}, u_{\mathcal{P}}, t_{\text{obs}} \mid \theta\right) &= \prod_{g\in\mathcal{G}} f_{T_g}\left(t_g \mid U_i = u_i, \forall i \in \text{pa}(\pi_g); \theta\right) \\ &\quad \times \prod_{p\in\mathcal{P}} f_{U_p}\left(u_p \mid T_j = t_j, \forall j \text{ s.t. } \pi_j = p; \theta\right) f_{T_{\text{obs}}}\left(t_{\text{obs}} \mid \theta\right) \\ &= \prod_{g\in\{\mathcal{G}\cup obs\}} \lambda_g e^{-\lambda_g\left(t_g - \max\limits_{i\in\text{pa}(\pi_g)} \min\limits_{j:\pi_j=i} t_j\right)} \\ &\quad \times \mathbb{1}_{t_g > \max\limits_{i\in\text{pa}(\pi_g)} \min\limits_{j:\pi_j=i} t_j} \qquad \forall g \in \mathcal{G} \end{aligned} \tag{1}$$

Conditioned on the observation time and on the waiting times of the pathways, the true mutational statuses of genes in separate pathways are independent. Their conditional probability distribution is

$$P\left(\mathcal{X} \mid \mathcal{T}_{\mathcal{G}}, \mathcal{U}_{\mathcal{P}}, T_{\text{obs}}, \theta\right) = \prod_{g\in\mathcal{G}} P\left(X_g \mid T_g, U_{\pi_g}, T_{\text{obs}}, \theta\right) = \prod_{p\in\mathcal{P}} P\left(S^p \mid T_k, \forall k \text{ s.t } \pi_k = p; T_{\text{obs}}; \theta\right) \tag{2}$$

In the absence of noise, in a given pathway p, a true genotype contains no mutations if the observation time is shorter than the waiting times of all pathway members. Alternatively, since $\mu_p = 1$, a single mutation (of gene j) is present in the set s^p if and only if p is perfectly mutually exclusive. The presence of additional mutations in p represents a deviation from mutual exclusivity and, in the absence of noise, has probability 0

$$P\left(s^p \mid T_k = t_k, \forall k \text{ s.t. } \pi_k = p; T_{\text{obs}} = t_{\text{obs}}; \theta\right) = \begin{cases} P\left(T_{\text{obs}} < \min\limits_{i:\pi_i=p} T_i\right) & \text{if } s^p = \emptyset \\ P\left(T_j < \min\limits_{i:\pi_i=p; i\neq j} (T_i, T_{\text{obs}})\right) & \text{if } s^p = \{j\} \\ 0 & \text{if } |s^p| \geq 2 \end{cases} \tag{3}$$

The likelihood of the true genotype X is the marginal probability

$$P\left(\mathcal{X} \mid \theta\right) = \int_{\mathbb{R}_{\geq 0}^{n+k+1}} \cdots \int f_{T_{\mathcal{G}}, U_{\mathcal{P}}, T_{\text{obs}}}\left(t_{\mathcal{G}}, u_{\mathcal{P}}, t_{\text{obs}} \mid \theta\right) P\left(\mathcal{X} \mid T_{\mathcal{G}}, U_{\mathcal{P}}, T_{\text{obs}}, \theta\right) \mathrm{d}t_{\mathcal{G}} \; \mathrm{d}u_{\mathcal{P}} \; \mathrm{d}t_{\text{obs}} \tag{4}$$

which can be decomposed into a sum over all linear extensions of the poset $(\mathcal{P}_{\text{obs}}, \prec)$ (proof in Sect. S1.1). Conditioned on the true genotype, the likelihood of the observed genotype is

$$P\left(Y \mid \mathcal{X}, \theta\right) = \prod_{g \in \mathcal{G}} P\left(Y_g \mid X_g, \theta\right) = \varepsilon^{\mathrm{d}(Y-\mathcal{X})} \left(1-\varepsilon\right)^{n-\mathrm{d}(Y-\mathcal{X})} \tag{5}$$

where $\mathrm{d}\left(\mathcal{X}, Y\right) = \sum_{g \in \mathcal{G}} \mid Y_g - X_g \mid$ is the Hamming distance between the true genotype $\mathcal{X}$ and the observed genotype Y. If we denote by $J\left(\mathcal{P}\right)$ the set of all true genotypes compatible with the poset $(\mathcal{P}_{\text{obs}}, \prec)$, it follows that the likelihood of the observed genotype Y is

$$L\left(\theta \mid Y\right) = P\left(Y \mid \theta\right) = \sum_{\mathcal{X} \in J(\mathcal{P})} P\left(Y \mid \mathcal{X}, \theta\right) P\left(\mathcal{X} \mid \theta\right) \tag{6}$$

The log likelihood of a dataset of N independent samples $\mathbf{Y} = \left(Y^1, \ldots, Y^N\right)$, where each Y^i is an observed genotype, is

$$l\left(\theta \mid \mathbf{Y}\right) = \sum_{i=1}^{N} \log L\left(\theta \mid Y^i\right) \tag{7}$$

2.2 Inference

Given a dataset of N independent samples $\mathbf{Y}$, the inference scheme of path-TiMEx aims to maximize the log likelihood in (7). The initial solution of our inference procedure is generated by first running TiMEx [12] with default parameters ($\mu_{\text{pair}} = 0.5$, $p_{\text{pair}} = 0.01$ and $p_{\text{group}} = 0.1$) iteratively on the input binary dataset. Each time, the largest and most significant pathway is retained and TiMEx is ran again on the dataset from which the members of the recently-identified pathway were excluded. This procedure is repeated until no more significantly mutually exclusive pathways are found. TiMEx identifies mutually exclusive groups as maximal cliques, and estimates the waiting time rates of the

genes λ_g, as well as the mutual exclusivity intensities of the pathways μ_p, by computing the maximum likelihood estimates (MLEs) numerically. In a second step, the CBN routine [19] is ran on the binarized matrix obtained by encoding a pathway as altered whenever at least one of its gene members is altered. CBN estimates the waiting time rates of the pathways λ_p, as well as the error probability ε, via a nested Expectation-Maximization (EM) algorithm [13], where, conditioned on an optimal fixed value of ε, λ_p are optimized and, conditioned on the optimal fixed values of λ_p, ε is optimized. The progression among pathways is inferred via simmulated annealing [24].

Starting from the initial solution, the joint optimization of pathways and structure follows a framework similar to an EM algorithm: conditioned on the optimal assignment π, the poset $(\mathcal{P}, \prec)$ is optimized and, conditioned on the optimized poset, a new optimal assignment is computed. The procedure is repeated until both π and $(\mathcal{P}, \prec)$ converge. Specifically, in iteration $i \geq 2$, given an optimal fixed structure $\left(\widehat{\mathcal{P}}_{i-1}, \widehat{\prec}_{i-1}\right)$, π_i is optimized via a Markov Chain Monte Carlo (MCMC) scheme [30]. This amounts to minimizing the number of contradictions due to both mutual exclusivity and progression E_i, which represents the number of ones that would need to be changed to zeroes to ensure consistency of the data $\mathbf{Y}$ with both π_i and $\left(\widehat{\mathcal{P}}_{i-1}, \widehat{\prec}_{i-1}\right)$. For fixed parameters $(\mathcal{P}, \prec)$, λ, and $\mu = 1$, $E_i = \sum_{j=1}^{N} \mathrm{d}\left(X^j, Y^j\right)$, where $\mathbf{X} = \left(X^1, \ldots, X^n\right)$ is the set of true genotypes. Therefore, mimizing E_i is equivalent to maximizing the likelihood in (7) as a function of ε, and $\widehat{E}_i = Nn\ \widehat{\varepsilon}_i$ (proof in Sect. S1.2). A detailed explanation of the MCMC scheme, including its computational complexity, is given in Sect. S1.2.

Given the optimal assignment $\widehat{\pi}_i$, $(\mathcal{P}_i, \prec_i)$ is optimized via simmulated annealing, by locally maximizing the likelihood function in (7), given a set of optimal parameters $\widehat{\lambda_p}$ and $\widehat{\varepsilon}$ found via the nested EM procedure. Starting from a poset P, a new acylic poset P' is proposed and it is accepted either if it increases the likelihood, or with probability $\exp\left(-\left[l\left(\widehat{\lambda_p}, \widehat{\varepsilon}, P\right) - l\left(\widehat{\lambda_p}, \widehat{\varepsilon}, P'\right)\right]/T\right)$. The temperature T reduces the risk of remaining in local optima.

3 Results

3.1 Simulations

We compared the performance of the iterative and the classic TiMEx procedures using simulated data, for various noise levels ε and number of samples N. Specifically, we considered two independent mutually exclusive groups consisting of two genes each, for which the waiting time rates λ were uniformly sampled between 0.1 and 2. According to this simulation scenario, the genes cover a wide range of alteration frequencies, from less than 1 % to more than 60 % [12]. In the absence of noise, all pathways were perfectly mutually exclusive, *i.e.* $\mu = 1$. With noise probability $\varepsilon \in \{0, 0.05, 0.1, 0.15\}$, each entry in the binary alteration matrix consisting of $N \in \{100, 400, 1000\}$ patients was flipped either from zero to

one or otherwise. We generated 100 datasets corresponding to each of the above configurations, and inferred mutually exclusive groups with either the iterative or the classic TiMEx procedures. Iterative TiMEx clearly outperformed Classic TiMEx for all tested sample sizes and noise levels (Fig. S1).

Additionally, we designed a simulation experiment assessing the performance and stability of pathTiMEx, for the noise levels ε and sample sizes N mentioned above. We simulated a progression model consisting of 12 genes assigned to 5 mutually exclusive pathways. The assignment of genes to pathways was random, with the sole restriction that each pathway contained at least one gene. 87 % of the 100 generated datasets included at least one pathway with a single member and 78 % of them included at least one pathway with at least four members. The order constraints among pathways were generated as a DAG with a number of edges uniformly sampled as to satisfy an edge density (number of existing edges divided by number of all possible edges) of between 0.4 and 0.8. In consequence, each poset had at least one unconnected pathway, and the majority of pathways were connected to either one or two other pathways. Our simulation scenarios resemble real cancer datasets in terms of number of pathways, alteration frequencies, noise levels and the presence of pathways either consisting of single genes or unconnected to other pathways [20,34]. On all simulated datasets, we jointly inferred mutually exclusive pathways and the order constraints among them with pathTiMEx. We compared our results with the initial solution of pathTiMEx, namely iteratively identifying mutually exclusive groups with TiMEx [12], followed by optimizing the order dependencies among the groups a single time with CBN [19]. As Rapahel and Vandin's tool [34] is not publicly available, we were not able to compare our results with their approach on simulated data.

The convergence of pathTiMEx increases with increasing sample size and decreasing noise levels (Table S1). For $\varepsilon = 0$, the algorithm converged in all cases for all sample sizes, while for $\varepsilon = 0.1$, it converged in 51 % of the runs for $N = 100$, 97 % for $N = 400$, and 98 % for $N = 1000$. Similarly, up to random fluctuations, the number of iterations required for convergence decreases with increasing sample size and decreasing noise levels. When $N = 400$, pathTiMEx converges in an average of 2.6 iterations for $\varepsilon = 0$, as compared to an average of 12.5 iterations when $\varepsilon = 0.15$. The runtime per iteration increases with increasing sample size, and remains largely uninfluenced by noise, with the exception of the largest sample size. The decrease in runtime per iteration with increasing noise level when $N = 1000$ can be explained by a slight, but noticeable, increase in false negative rate.

pathTiMEx identified the optimal assignment of genes to pathways in the largest percentage of cases (Fig. S2A). The average Rand index [33] between the true pathways and the estimated ones ranged from 95 % for $\varepsilon = 0$ to 84 % for $\varepsilon = 0.15$ in the case of $N = 100$ (with corresponding Jaccard indices [27] of 0.75 and 0.44), and from 99 % to 92 % when $N = 1000$ (with corresponding Jaccard indices of 0.94 and 0.68). pathTiMEx outperformed the naive initial solution in almost all situations, with the exception of large sample sizes and a large noise level. In these cases, as previously mentioned, the joint effect of large sample size

and low progression signal lead to an increase in both false positive and false negative rates. For all sample sizes and noise levels, among the cases in which the genes were correctly assigned to pathways (clustering similarity indices of 1), pathTiMEx further identified the correct progression structure in the large majority of cases (Fig. S2B). As expected, structure similarity increases with increasing sample size and decreasing noise levels.

3.2 Cancer Data

We used pathTiMEx to jointly infer mutually exclusive driver pathways and the order constraints among them in publicly available cancer data: a small colorectal cancer dataset [43], a large colorectal cancer dataset (TCGA, provisional) [9], and a large glioblastoma dataset (TCGA, provisional) [9]. We compared our results with the naive approach of decoupling the identification of mutually exclusive groups and the inference of their progression, which is the initial solution to our stochastic joint optimization scheme. Additionally, as Raphael and Vandin's approach [34] is a special case of pathTiMEx, we facilitated the direct comparison between the two tools by optimizing the assignment of genes to pathways under a fixed linear progression with an unspecified number of stages. Under these constraints, pathTiMEx is much faster than the algorithm in [34], as solely optimizing mutual exclusivity for a fixed progression involves one single iteration of the MCMC chain.

For the three datasets, we assessed the stability of the joint optimal solutions across 100 runs, and in each run the joint optimization procedure was iterated at most 100 times (Table S2). For each identified order dependency among pathways, we computed its weight, *i.e.* its frequency across runs. As the initial pathways were fixed in all cases, the edge weights in the initial solutions only evaluate the stability of the order constraints and are usually higher. Additionally, the estimation of the waiting time rates λ and the error probability ε was highly stable (less than 0.01 % variance across runs). According to our model, high rates of evolution λ indicate early events, while low λ indicate late events.

Colorectal Cancer (Wood et al. 2007 [43]). The colorectal cancer dataset published by Wood et al. [43], consisting of eight genes mutated with frequency above 5 % in 95 samples, has been previously used in [20,34] for analyzing cancer progression among pathways. The optimal structure inferred by pathTiMEx (Fig. 3A) reaffirmed the current knowledge on tumor progression in colorectal cancer: *APC* is an early event ($\lambda = 9.5$), followed by *KRAS* ($\lambda = 2.89$) and *TP53* ($\lambda = 1.92$). pathTiMEx immediately converged in all 100 runs to the same joint solution (Table S2), with an estimated $\widehat{\varepsilon}$ of 0.13.

None of the five identified mutually exclusive pathways were also identified by the naive approach, *i.e.* they were not part of the initial solution, which means that mutual exclusivity and progression have been jointly optimized. For example, the naive approach estimated that the group *TP53* and *PIK3CA* is independent and it occurs earlier in progression than *KRAS* and *TCF7L2*. The same temporal order was inferred by Raphael and Vandin in [34], and also by pathTiMEx if assuming a linear structure (Fig. 4A). On the contrary, pathTiMEx

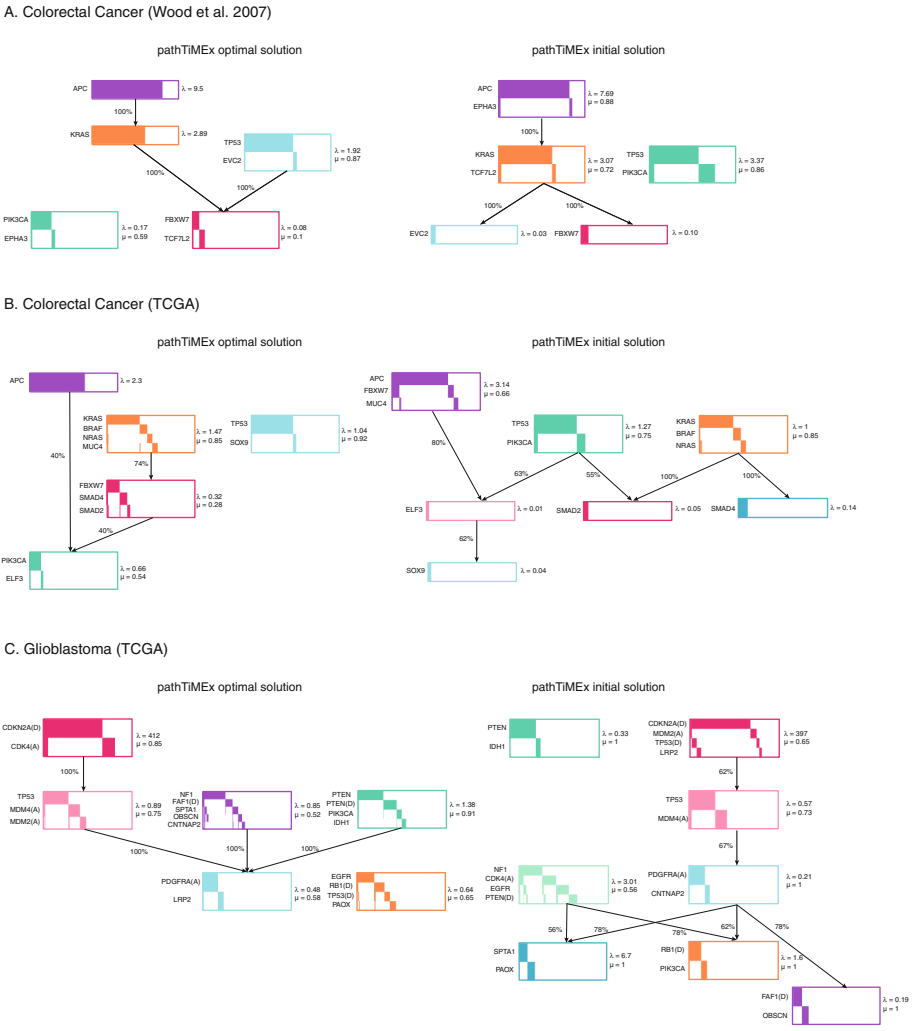


Fig. 3. Results of pathTiMEx on publicly available cancer data. The optimal solution inferred by pathTiMEx is shown on the left hand side, for three cancer datasets. The naive approach (right hand side) assumes that pathways and progression are decoupled, and it represents the initial solution of our stochastic joint optimization scheme. Each pathway is followed to its right by its waiting time rate estimate λ and by the estimate of the intensity of mutual exclusivity μ. The weight on each edge (as a percentage) shows how likely it is to infer a directed dependency between the same two groups, across 100 runs. The percentage is computed relative to all the cases in which the algorithm converged in less than 100 iterations (100/100 runs for the colorectal cancer dataset in Wood et al. 91/100 runs for the TCGA colorectal cancer dataset and 26/100 runs for glioblastoma). In the glioblastoma dataset, *(D)* indicates the copy number deletion of a gene, while *(A)* indicates its copy number amplification.

identified *TP53* and *EVC2* ($\lambda = 1.92$) as a later event than *KRAS*, which is consistent with the current knowledge on colorectal tumorigenesis [17,18]. Additionally, by enforcing a linear order among pathways, our model obtained a better score than Raphael and Vandin's approach [34] (Fig. 4A). The late events found by pathTiMEx were *PIK3CA* and *EPHA3* ($\lambda = 0.17$), together with *FBXW7* and *TCF7L2* ($\lambda = 0.08$). These findings, similar to the ones obtained if assuming a linear progression, confirm previous observations in [20].

The gene-level model of tumor progression proposed by Gerstung et al. in [20] also identified *APC* as an early event, followed by a pathway including *KRAS* and *TP53*, which was unconnected to other pathways. However, their pathway-level model of cancer progression employed large literature-derived pathways and did not attempt to identify pathways *de novo*.

Colorectal Cancer (TCGA). For 258 colorectal cancer samples, we analyzed point mutation data for 473 genes, either significantly recurrently mutated as identified by MutSig [25], or part of copy number aberrations as identified by GISTIC 2.0 [31]. In addition to the three mutually exclusive groups found by Iterative TiMEx [12], we introduced in our analysis four more genes with known involvement in colorectal tumorigenesis [34]: *ELF3*, *SOX9*, *SMAD2* and *SMAD4*. pathTiMEx converged in 91 % of the runs, with an estimated $\widehat{\varepsilon}$ of 0.16 (Table S2). The most likely progression structure (Fig. 3B) was found in 40 % (37) of the 91 runs. The average Jaccard index [27] between the pathways alternatively reported and the most likely pathways was 0.81 (with a minimum of 0.66). Across all runs, provided that the reported pathways were also the most likely ones, the dependencies among them were identical.

The optimal structure inferred by pathTiMEx (Fig. 3B) was in high accordance with our findings on the colorectal cancer dataset analyzed above (Fig. 3A), and also highly consistent with the current knowledge on colorectal tumorigenesis [17,18]. Specifically, *APC* ($\lambda = 2.3$) was the earliest event, followed by a mutually exclusive pathway including *KRAS* ($\lambda = 1.47$), and later followed by *TP53* and *SOX9* ($\lambda = 1.04$). Interestingly, on both colorectal cancer datasets, even if *APC* was initially part of a mutually exclusive group, in the optimal solution it is the single early starting event. This finding emphasizes the particular importance that *APC* has in the progression of colorectal cancer [18]. As previously, the naive approach identified *TP53* as an early event, and only following the joint optimization of mutual exclusivity and progression, mutations in *TP53* were reported as later events. Most of the genes part of the identified mutually exclusive groups are known interaction partners in colorectal cancer, such as the tumor suppressors *SMAD2* and *SMAD4* [34]. Interestingly, pathTiMEx identified*MUC4*, a gene which is aberrantly expressed in colorectal adenocarcinomas, but with an unknown prognostic value [36], as part of the same mutually exclusive group with three oncogenes in the Ras-Raf pathway, namely *BRAF*, *NRAS* and *KRAS* [34].

In conclusion, the results of pathTiMEx are highly consistent on both colorectal cancer datasets, and more consistent with the literature than previous

approaches. By imposing a linear order, our model estimated a progression of four stages, and obtained a better score than Raphael and Vandin's approach [34] (Fig. 4B). Hence, pathTiMEx offers a better explanation of colorectal tumorigenesis than the naive initial solution and the models in [20,34].

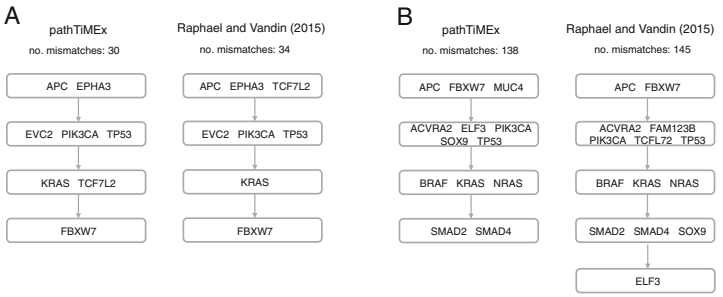


Fig. 4. Comparison between pathTiMEx and the approach of Raphael and Vandin (2015) [34], on (A) the colorectal cancer dataset from Wood et al. (2007) and (B) the colorectal cancer dataset from TCGA. In order to facilitate the direct comparison between the two models, we optimized the assignment of genes to pathways with pathTiMEx, assuming a fixed linear progression with an unspecified number of stages. We computed the scores E for the two models, *i.e.* the number of mismatches due to violation of both mutual exclusivity and progression constraints. pathTiMEx estimates four progression stages on both colorectal cancer datasets, and performs better than the model in [34], by identifying pathways with smaller numbers of mismatches. In (B), minor differences in data preprocessing led to the gene *MUC4* only being present in pathTiMEx, and the genes *FAM123B* and *TCFL72* only being present in the model from [34].

Glioblastoma (TCGA). We analyzed the glioblastoma dataset discussed in [26] and preprocessed as explained in [12], consisting of point mutations and copy number aberrations for 486 genes, in 261 patients. In glioblastoma, tumor progression is less known and more difficult to infer than in colorectal cancer [10]. Consequently, our algorithm converged slower than in the case of the two colorectal cancer datasets (Table S2), and the noise rate was estimated to $\widehat{\varepsilon} = 0.2$. pathTiMEx only reached a stable solution in 26 % of the runs in less than 100 iterations, to however always the same pathways and dependencies among them.

The optimal solution of pathTiMEx (Fig. 3C) consisted of fewer pathways than the initial solution. Unlike colorectal cancer, all members of the optimal pathways were part of large mutually exclusive groups, which points to the large variability of tumor progression in glioblastoma. Following the joint optimization scheme, pathTiMEx identified as a very early event ($\lambda = 412$) the group which included the deletion of *CDKN2A* and the amplification of *CDK4A*. These two genes are interaction partners and belong to the pathways CDC42 signaling

events and Cyclin D associated events in G1. This group is directly temporally related to a later event consisting of the copy number amplifications of *MDM4* and *MDM2*, together with the point mutation of *TP53* ($\lambda = 0.89$), which are all members of the p53 pathway [6] and were previously identified as playing a role in tumor progression in glioblastoma [10,34]. The point mutation of *TP53* and the amplification of *MDM4* were initially reported as a mutually exclusive pair by the naive approach. Following the joint optimization scheme, the pair was merged with the amplification of *MDM2*, a member of the same pathway. Interestingly, the point mutation and the deletion of *PTEN*, together with the point mutation of *PIK3CA* ($\lambda = 1.38$), which belong to the PI3K pathway, were identified as part of a mutually exclusive group which also included *IDH1*.

In conclusion, despite the high variability in the data, pathTiMEx offers new insights into tumorigenesis in glioblastoma, by jointly optimizing mutually exclusive pathways and the order constraints among them.

4 Conclusion

In this paper, we introduced pathTiMEx, a probabilistic model of tumor progression at the level of mutually exclusive driver pathways, together with an efficient stochastic joint optimization scheme. pathTiMEx is a step forward from the approaches which separately infer either mutually exclusive groups of alterations, or progression in tumorigenesis at the level of single genes. The simultaneous identification of driver pathways and the evolutionary order constraints among them may have important therapeutic implications, particularly by targeting members of early mutually exclusive pathways [42].

pathTiMEx is a direct generalization of both TiMEx [12], a waiting time model for independent mutually exclusive pathways, and CBN [19], a waiting time model for cancer progression at the level of single genes. It assumes that, in tumor development, alterations can either occur independently, or depend on eachother by being part of the same pathway or by following particular progression paths. By inferring these two types of potential dependencies simultaneously, pathTiMEx jointly addresses the two fundamental questions of identifying drivers and progression. pathTiMEx models the order constraints among pathways as a DAG, hence the only previous approach performing simultaneous inference [34] is a special case of pathTiMEx, corresponding to the situation when the structure is fixed to a linear path.

However, despite its advantages, pathTiMEx still models a very simplified representation of tumor progression. Future extensions of the model may aim to relax some of its assumptions, such as the hard assignment of genes to pathways, which doesn't allow for pathway cross-talk, or the irreversibility of mutations, which renders our approach not applicable to gene expression data. Moreover, model performance for large datasets with high levels of noise may improve by devising alternative ways of modeling temporal dependencies between waiting times, specifically accounting for false positive and false negative dependencies.

In applications on cancer datasets, pathTiMEx recapitulates previous knowledge on tumorigenesis, while also offering new insights on the order constraints

among pathways in cancer progression. The results of pathTiMEx are highly consistent on the two colorectal cancer datasets analyzed, and more consistent with the literature than previous approaches. In glioblastoma, pathTiMEx proposes the existence of a single early causal event consisting of the amplification of *CDK4* and the deletion of *CDKN2A*. These results clearly indicate that pathTiMEx is not only theoretically justified by its treatment of tumorigenesis on the level of pathways as a probabilistic generative process, but is also fruitfully applicable in practice.

Acknowledgements. The authors would like to thank Hesam Montazeri for helpful discussions.
Funding. Simona Cristea was financially supported by the Swiss National Science Foundation (Sinergia project 136247).

References

1. Attolini, C.S.O., Cheng, Y.K., Beroukhim, R., Getz, G., Abdel-Wahab, O., Levine, R.L., Mellinghoff, I.K., Michor, F.: A mathematical framework to determine the temporal sequence of somatic genetic events in cancer. Proc. Nat. Acad. Sci. **107**(41), 17604–17609 (2010)
2. Babur, Ö., Gönen, M., Aksoy, B.A., Schultz, N., Ciriello, G., Sander, C., Demir, E.: Systematic identification of cancer driving signaling pathways based on mutual exclusivity of genomic alterations. bioRxiv, p. 009878 (2014)
3. Beerenwinkel, N., Eriksson, N., Sturmfels, B.: Conjunctive bayesian networks. Bernoulli **13**(4), 893–909 (2007)
4. Beerenwinkel, N., Schwarz, R.F., Gerstung, M., Markowetz, F.: Cancer evolution: mathematical models and computational inference. Syst. Biol. **64**(1), e1–e25 (2015)
5. Beerenwinkel, N., Sullivant, S.: Markov models for accumulating mutations. Biometrika, p. asp023 (2009)
6. Brennan, C.W., Verhaak, R.G., McKenna, A., Campos, B., Noushmehr, H., Salama, S.R., Zheng, S., Chakravarty, D., Sanborn, J.Z., Berman, S.H., et al.: The somatic genomic landscape of glioblastoma. Cell **155**(2), 462–477 (2013)
7. Cancer Genome Atlas Network and others: Comprehensive molecular portraits of human breast tumours. Nature **490**(7418), 61–70 (2012)
8. Cancer Genome Atlas Research Network: Integrated genomic analyses of ovarian carcinoma. Nature **474**(7353), 609–615 (2011)
9. Cerami, E., Gao, J., Dogrusoz, U., Gross, B.E., Sumer, S.O., Aksoy, B.A., Jacobsen, A., Byrne, C.J., Heuer, M.L., Larsson, E., et al.: The cBio cancer genomics portal: an open platform for exploring multidimensional cancer genomics data. Cancer Discov. **2**(5), 401–404 (2012)
10. Cheng, Y.K., Beroukhim, R., Levine, R.L., Mellinghoff, I.K., Holland, E.C., Michor, F., et al.: A mathematical methodology for determining the temporal order of pathway alterations arising during gliomagenesis. PLoS Comput. Biol. **8**(1), e1002337 (2012)
11. Ciriello, G., Cerami, E., Sander, C., Schultz, N.: Mutual exclusivity analysis identifies oncogenic network modules. Genome Res. **22**(2), 398–406 (2012)
12. Constantinescu, S., Szczurek, E., Mohammadi, P., Rahnenfuhrer, J., Beerenwinkel, N.: TiMEx: a waiting time model for mutually exclusive cancer alterations. Bioinformatics (2015). doi:10.1093/bioinformatics/btv400

13. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Stat. Soc. Ser. B (Methodol.) **39**(1), 1–38 (1977)
14. Desper, R., Jiang, F., Kallioniemi, O.P., Moch, H., Papadimitriou, C.H., Schäffer, A.A.: Inferring tree models for oncogenesis from comparative genome hybridization data. J. Comput. Biol. **6**(1), 37–51 (1999)
15. Diaz-Uriarte, R.: Identifying restrictions in the order of accumulation of mutations during tumor progression: effects of passengers, evolutionary models, and sampling. BMC Bioinformatics **16**(1), 41 (2015)
16. Farahani, H.S., Lagergren, J.: Learning oncogenetic networks by reducing to mixed integer linear programming. Plos One **8**(6), e65773 (2013)
17. Fearon, E.R.: Molecular genetics of colorectal cancer. Annu. Rev. Pathol. **6**, 479–507 (2011)
18. Fearon, E.R., Vogelstein, B.: A genetic model for colorectal tumorigenesis. Cell **61**(5), 759–767 (1990)
19. Gerstung, M., Baudis, M., Moch, H., Beerenwinkel, N.: Quantifying cancer progression with conjunctive bayesian networks. Bioinformatics **25**(21), 2809–2815 (2009)
20. Gerstung, M., Eriksson, N., Lin, J., Vogelstein, B., Beerenwinkel, N.: The temporal order of genetic and pathway alterations in tumorigenesis. PloS One **6**(11), e27136 (2011)
21. Hjelm, M., Höglund, M., Lagergren, J.: New probabilistic network models and algorithms for oncogenesis. J. Comput. Biol. **13**(4), 853–865 (2006)
22. Jerby-Arnon, L., Pfetzer, N., Waldman, Y.Y., McGarry, L., James, D., Shanks, E., Seashore-Ludlow, B., Weinstock, A., Geiger, T., Clemons, P.A., et al.: Predicting cancer-specific vulnerability via data-driven detection of synthetic lethality. Cell **158**(5), 1199–1209 (2014)
23. Kim, Y.A., Cho, D.Y., Dao, P., Przytycka, T.M.: Memcover: integrated analysis of mutual exclusivity and functional network reveals dysregulated pathways across multiple cancer types. Bioinformatics **31**(12), i284–i292 (2015)
24. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., et al.: Optimization by simulated annealing. Science **220**(4598), 671–680 (1983)
25. Lawrence, M.S., Stojanov, P., Polak, P., Kryukov, G.V., Cibulskis, K., Sivachenko, A., Carter, S.L., Stewart, C., Mermel, C.H., Roberts, S.A., et al.: Mutational heterogeneity in cancer and the search for new cancer-associated genes. Nature **499**(7457), 214–218 (2013)
26. Leiserson, M.D., Blokh, D., Sharan, R., Raphael, B.J.: Simultaneous identification of multiple driver pathways in cancer. PLoS Comput. Biol. **9**(5), e1003054 (2013)
27. Levandowsky, M., Winter, D.: Distance between sets. Nature **234**(5323), 34–35 (1971)
28. Loohuis, L.O., Caravagna, G., Graudenzi, A., Ramazzotti, D., Mauri, G., Antoniotti, M., Mishra, B.: Inferring tree causal models of cancer progression with probability raising. Plos One **9**(10), e108358 (2014)
29. Luo, J., Solimini, N.L., Elledge, S.J.: Principles of cancer therapy: oncogene and non-oncogene addiction. Cell **136**(5), 823–837 (2009)
30. Madigan, D., York, J., Allard, D.: Bayesian graphical models for discrete data. In: International Statistical Review/Revue Internationale de Statistique, pp. 215–232 (1995)

31. Mermel, C.H., Schumacher, S.E., Hill, B., Meyerson, M.L., Beroukhim, R., Getz, G., et al.: GISTIC2.0 facilitates sensitive and confident localization of the targets of focal somatic copy-number alteration in human cancers. Genome Biol. **12**(4), R41 (2011)
32. Ramazzotti, D., Caravagna, G., Olde-Loohuis, L., Graudenzi, A., Korsunsky, I., Mauri, G., Antoniotti, M., Mishra, B.: CAPRI: efficient inference of cancer progression models from cross-sectional data. Bioinformatics, p. btv296 (2015)
33. Rand, W.M.: Objective criteria for the evaluation of clustering methods. J. Am. Stat. Assoc. **66**(336), 846–850 (1971)
34. Raphael, B.J., Vandin, F.: Simultaneous inference of cancer pathways and tumor progression from cross-sectional mutation data. In: Sharan, R. (ed.) RECOMB 2014. LNCS, vol. 8394, pp. 250–264. Springer, Heidelberg (2014)
35. Sakoparnig, T., Beerenwinkel, N.: Efficient sampling for bayesian inference of conjunctive bayesian networks. Bioinformatics **28**(18), 2318–2324 (2012)
36. Shanmugam, C., Jhala, N.C., Katkoori, V.R., Wan, W., Meleth, S., Grizzle, W.E., Manne, U.: Prognostic value of mucin 4 expression in colorectal adenocarcinomas. Cancer **116**(15), 3577–3586 (2010)
37. Stratton, M.R., Campbell, P.J., Futreal, P.A.: The cancer genome. Nature **458**(7239), 719–724 (2009)
38. Szczurek, E., Beerenwinkel, N.: Modeling mutual exclusivity of cancer mutations. PLoS Comput. Biol. **10**(3), e1003503 (2014)
39. Torti, D., Trusolino, L.: Oncogene addiction as a foundational rationale for targeted anti-cancer therapy: promises and perils. EMBO Mol. Med. **3**(11), 623–636 (2011)
40. Vandin, F., Upfal, E., Raphael, B.J.: De novo discovery of mutated driver pathways in cancer. Genome Res. **22**(2), 375–385 (2012)
41. Vogelstein, B., Papadopoulos, N., Velculescu, V.E., Zhou, S., Diaz, L.A., Kinzler, K.W.: Cancer genome landscapes. Science **339**(6127), 1546–1558 (2013)
42. Weinstein, I.B.: Addiction to oncogenes-the achilles heal of cancer. Science **297**(5578), 63–64 (2002)
43. Wood, L.D., Parsons, D.W., Jones, S., Lin, J., Sjöblom, T., Leary, R.J., Shen, D., Boca, S.M., Barber, T., Ptak, J., et al.: The genomic landscapes of human breast and colorectal cancers. Science **318**(5853), 1108–1113 (2007)
44. Wu, H.T., Leiserson, M.D., Vandin, F., Raphael, B.J.: Comet: A statistical approach to identify combinations of mutually exclusive alterations in cancer. Cancer Res. **75**(15 Supplement), 1936–1936 (2015)

Clonality Inference from Single Tumor Samples Using Low Coverage Sequence Data

Nilgun Donmez[1,2], Salem Malikic[1,2], Alexander W. Wyatt[2,3], Martin E. Gleave[2], Colin C. Collins[2,3], and S. Cenk Sahinalp[1,2,4(✉)]

[1] School of Computing Science, Simon Fraser University, Burnaby, BC, Canada
cenk@sfu.ca
[2] Vancouver Prostate Centre, Vancouver, BC, Canada
[3] Department of Urologic Sciences, University of British Columbia, Vancouver, BC, Canada
[4] School of Informatics and Computing, Indiana University, Bloomington, IN, USA

Abstract. Inference of intra-tumor heterogeneity can provide valuable insight into cancer evolution. Somatic mutations detected by sequencing can help estimate the purity of a tumor sample and reconstruct its subclonal composition. While several methods have been developed to infer intra-tumor heterogeneity, the majority of these tools rely on variant allele frequencies as estimated via ultra-deep sequencing from multiple samples of the same tumor. In practice, obtaining sequencing data from a large number of samples per patient is only feasible in a few cancer types such as liquid tumors, or in rare cases involving solid tumors selected for research. We introduce CTPsingle, which aims to infer the subclonal composition using low-coverage sequencing data from a single tumor sample. We show that CTPsingle is able to infer the purity and the clonality of single-sample tumors with high accuracy even restricted to a coverage depth of ∼30x.

Keywords: Intra-tumor heterogeneity · Cancer progression · DNA sequencing

1 Introduction

In the past decade, cancer genomics and sequencing revealed a striking degree of intra-tumor diversity in cancer. Molecular evidence increasingly suggests that this diversity has clinical implications. The pioneering work studying intra-tumor heterogeneity typically focused on a small number of selected genomic alterations from several tumor samples and involved manually reconstructing the phylogeny of these tumors [4,17]. Nevertheless, large-scale cancer sequencing efforts such as the PanCancer Analysis of Whole Genomes (PCAWG) require fully-automated methods [20].

Recently, we developed a tool named CITUP to tackle this problem in the existence of multiple samples from the same tumor. Using simulations and real data, we showed that CITUP is able to reconstruct the tumor phylogeny when

M. Singh (Ed.): RECOMB 2016, LNBI 9649, pp. 83–94, 2016.
DOI: 10.1007/978-3-319-31957-5_6

supplied with deep sequencing data on multiple samples from a single patient [12]. While targeted deep sequencing or high coverage exome sequencing are feasible alternatives to whole genome sequencing, obtaining multiple samples from solid tumors is a challenge in most clinical settings.

In fact, the majority of the tumor cohort currently analysed by PCAWG [20] have single sample, low to medium coverage sequencing data. Unfortunately, for CITUP and similar tools that exploit multiple samples to infer clonality, the ability to robustly determine the subclonal architecture of tumors deteriorate with decreasing number of samples per tumor [12]. To overcome this challenge and improve the purity and subclonal composition estimation in single-sample tumors, we introduce a new tool named CTPsingle that is specifically designed to work with low coverage sequencing data from a single sample.

CTPsingle features a robust clustering framework based on a beta-binomial mixture model and infers possible phylogenies using a fast mixed integer linear programming (mILP) formulation. Currently, CTPsingle is also used to infer clonality as a part of the Tumor Evolution and Heterogeneity working group of PCAWG [20]. The core functionality of CTPsingle is implemented in R using open source packages *DPpackage* [7] and *lpSolve* [1]. CTPsingle is freely available from https://github.com/nlgndnmz/CTPsingle.

1.1 Related Work

CTPsingle is partially based on CITUP, which uses a mixed Quadratic Integer Programming (mQIP) framework. Like CITUP, CTPsingle works on somatic single nucleotide variants (sSNVs) on copy neutral regions of the genome. However, unlike CITUP, which takes variant allele frequencies (VAFs) as input, CTPsingle takes reference and variant read counts as input and clusters sSNVs using a beta-binomial mixture model. This allows CTPsingle to infer the number of subclones in advance of phylogeny search and account for the higher noise in VAFs associated with low coverage. In addition, CTPsingle employs a simplified, iterative mILP formulation implemented using the freely available *lpSolve* library [1] and does not rely on any commercial libraries such as IBM CPLEX™.

CTPsingle is also related to TrAp [19], PhyloSub [8], rec-BTP [6], Clomial [21], BayClone [18], PyClone [16], LICHeE [14] and AncesTree [3]. The majority of these methods are designed to work with SNVs in copy neutral regions and are developed specifically for multiple samples, while a few of them can also work with single-sample datasets. Other relevant tools such as THetA2 [13], TITAN [5], CLONET [15] and PhyloWGS [2] are designed to work on copy number data, although some of them allow the use of additional type of mutation calls.

While rec-BTP is also exclusively designed for single-sample tumors, we had previously shown that this method has inferior performance compared to CITUP even on single-sample datasets [12]. Moreover, this tool does not report which mutations are assigned to which subclones, prohibiting us from calculating some of the evaluation measures we use in this paper. Instead, we compare CTPsingle to AncesTree, LICHeE and PyClone, which can also take single-sample data as

input. AncesTree has an integer linear programming framework where it formulates the problem of clonality inference as a variant allele factorization problem [3]. LICHeE works by constructing an evolutionary constraint network and finding the best scoring spanning trees [14]. While PyClone does not attempt to infer tree topologies, it has a similar clustering framework to CTPsingle that is based on a Dirichlet process [16]. We show that CTPsingle outperforms these methods even when they are supplied with more than one sample per tumor. In addition, we compare CTPsingle to CITUP and demonstrate that CTPsingle performs better than CITUP in low-coverage datasets.

2 Methods

2.1 Input Processing

As input, CTPsingle takes reference and variant read counts for single nucleotide variant (SNV) calls. These calls can be obtained from whole-genome, whole-exome or targeted sequencing data. CTPsingle expects only somatic SNVs in its input; all germline mutations should be discarded in advance. CTPsingle also expects the mutations to reside in copy number neutral regions. In other words, CTPsingle assumes all mutations to be heterozygous in diploid regions, although mutations on non-autosomal (i.e. X and Y) chromosomes can be included if the gender of the patient is given. Tri-allelic mutations should also be discarded from the input.

2.2 Robust Clustering Using Beta-Binomial Mixture Modelling

The clustering of mutations is performed via a beta-binomial model in CTPsingle. Suppose we have M mutations called in a tumor sample. Let y_i denote the number of variant read counts for mutation i and n_i denote the number of total reads covering the same position (i.e. reference + variant reads). The following assumes y_i is binomial distributed with an unknown (i.e. variable) probability of success p_i:

$$y_i|(n_i, p_i) \sim Binom(n_i, p_i); i = 1, 2, ..., M \tag{1}$$

We further assume that the probability parameter p_i is generated from a Dirichlet Process (DP) as given below:

$$p_i|G \sim G \tag{2}$$

$$G|(\alpha, G_0) \sim Dir(\alpha, G_0) \tag{3}$$

Above, the concentration parameter α can either be given as a user-defined input or further sampled from a Gamma distribution. The baseline distribution G_0 is taken to be the Beta distribution with parameters a_1 and b_1:

$$G_0 = Beta(a_1, b_1) \tag{4}$$

Since the prior Beta distribution is conjugate to the Binomial distribution, resulting in a beta-binomial posterior, inference can be performed using a standard Markov Chain Monte Carlo (MCMC) method [11].

Above, the model parameters α, a_1, b_1 are set to $0.001, 5.0, 5.0$ respectively in our implementation. These values were selected empirically based on our observation on real data, however, they can be modified by the user if desired. In addition to the estimated values p_i, the algorithm provides the inferred number of clusters and an assignment of the mutations to these clusters.

Mutations on Haploid Regions: Since clustering is performed in read count space rather than cellular prevalence space, read counts for trivially homozygous mutations (such as those in X and Y chromosomes in males) should be properly adjusted. In CTPsingle, this is done by adjusting the total read count of the position. Let v_i and r_i denote the variant and reference read counts respectively for mutation i. Suppose t_i is the copy count of the region containing mutation i in the tumor sample and h_i is the copy count of that region in the normal sample. Then, n_i is set to be:

$$n_i = \left(\frac{2}{t_i} v_i\right) + \left(\frac{2}{h_i} r_i\right) \tag{5}$$

In essence, this formulation mimics 'phantom' chromosomes that emit reads containing the reference allele whenever appropriate and can be applied to clonal single copy deletions in addition to non-autosomal chromosomes. Finally, $y_i = v_i$ for all mutations. Note that here we assume that the normal sample does not contain any chromosomal abnormalities. Such regions, if exist, should be removed from automated analysis and manually investigated.

2.3 Estimation of Tumor Purity and Phylogeny Inference

From the clustering stage, we obtain the number k of subclones (i.e. clusters) and the mean allelic frequency s_j for each subclone $j = 1, 2, ..., k$. Since we modify all mutations to be heterozygous, cellular frequency of each subclone s_j^* is simply calculated as $s_j^* = 2s_j$. Given the cellular frequency of subclones, we estimate the tumor purity p as the highest frequency of any subclone: $p = max(s_j^*); j = 1, 2, ..., k$. To standardise predictions and identify clonal mutations easily, we adjust the cellular frequencies with the estimated tumor purity to obtain cancer cell fractions (CCFs). That is, if f_j denotes the CCF of subclone j, we set $f_j = \frac{s_j^*}{p}$. Essentially, this ensures the ancestral cluster to always have a CCF of 1.0. Note that this formulation assumes that the cancer is uni-focal. Like most other tools, CTPsingle can not explicitly handle multi-focal tumors.

Similar to CITUP, CTPsingle considers all tree topologies. An mILP formulation is applied to each tree topology independently in order to find the optimal assignment of the clusters subject to phylogenetic constraints. As the number of

clusters k is determined in advance, CTPsingle only processes those trees with k nodes. The detailed mILP formulation is described in Appendix 1.

3 Results

3.1 Simulations

To evaluate CTPsingle in a controlled environment and compare its performance to AncesTree, LICHeE and PyClone, we performed two sets of simulations: (1) with low coverage (∼32x) as routinely observed in whole genome sequencing experiments; and (2) with ultra-high coverage (∼2800x) as typically obtained from deep sequencing experiments. As the other methods are primarily intended for multi-sample datasets, for each coverage-depth, we also generate two sets of simulations: (a) with a single sample per tumor; and (b) with two samples per tumor. Although CTPsingle can not exploit the additional information that can be obtained from the second sample, our results demonstrate that it achieves better results in all four experiment settings. For each experiment setting, we simulate 50 instances. The rest of the simulation details are given in Appendix 2.

We compare the performance of the tools using three measures: (i) estimated tumor purity; (ii) number of subclones predicted; and (iii) root mean square error (RMSE) of cancer cell fractions.

The RMSE measure is calculated using the cancer cell fractions of mutations reported by each method as follows. Let $PCF(m_i)$ denote the cancer cell fraction of the subclone where mutation i is assigned to and $TCF(m_i)$ denote the true cancer cell fraction of the subclone from which mutation i originates. Then RMSE is calculated as:

$$\sqrt{\frac{\sum_{i \in S}(PCF(m_i) - TCF(m_i))^2}{|S|}} \tag{6}$$

where S represents the set of mutations reported by the tool. Above, the cancer cell fractions for each tool is computed using the subclonal frequencies reported by the tool and adjusted by the tumor purity estimated in the same way as is done in CTPsingle. We note that both AncesTree and LICHeE discard a small number of mutations in some cases, possibly giving them an unfair advantage in the calculation of RMSE. CTPsingle and PyClone report all mutations. The calculation of tumor purity, number and frequency of the subclones, and the running parameters for the tools are described in Appendix 2.

Figure 1 shows the comparison of the true versus predicted tumor purities for each experiment. As can be seen from the plots, CTPsingle outperforms other methods in all experiments. For the low coverage datasets, both AncesTree and LICHeE tend to underestimate the purity, although AncesTree also calls near 1.0 purity in some cases. While the purity estimates are significantly improved for AncesTree in the case of deep coverage, LICHeE estimates show little difference for deep coverage. This is probably due to the fact that this method directly works with variant allele frequencies hence can not distinguish between high

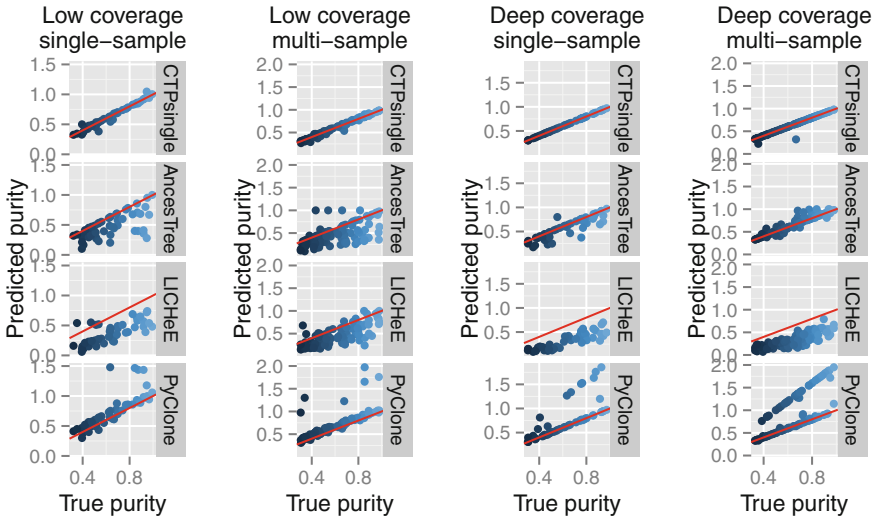


Fig. 1. Comparison of true versus predicted tumor purity across the simulation experiments. Each dot represents a distinct sample and is colored based on its real tumor purity. The red lines illustrate the $y = x$ line in each plot.

and low coverage data. In contrast, CTPsingle estimates purity with almost 100 % accuracy in the deep coverage samples. The two outliers in the case of multi-sample deep coverage experiment belong to the same tumor instance and it is due to the fact that this tumor contains a subclone consisting of only 2 mutations. We also note that PyClone appears to report a purity over 1.0 for some samples. This is because this tool reports allelic frequencies rather cellular frequencies. Thus, we multiply the frequencies reported by PyClone by a factor of 2 to calculate the purity estimated by this tool. However, for some samples the original frequencies seem to be closer to the true purity. Nevertheless, we choose to keep this practice as it improves the overall correlation between the predicted and true purities for this tool, especially for the low coverage samples. In addition, this problem does not affect the RMSE evaluation, as cancer cell fractions are already normalised (i.e. the highest CCF will always be 1.0).

Figure 2 shows the distribution of #subclones error for each method across the experiments. #subclones error is simply calculated as the absolute difference between the predicted number of subclones versus the true number of subclones. Once again, the plots show that CTPsingle outperforms the other methods in all experiment settings. In deep coverage datasets, CTPsingle correctly estimates the number of subclones in all but a few cases.

The histogram of the RMSE values for each sample is shown in Fig. 3. In the low coverage datasets, RMSE values are typically below 0.3 for CTPsingle, while this measure can range up to 0.8 for AncesTree and LICHeE. Note that the samples with low RMSE often represent the cases where the number of subclones are correctly identified, although RMSE can also be low if two subclones

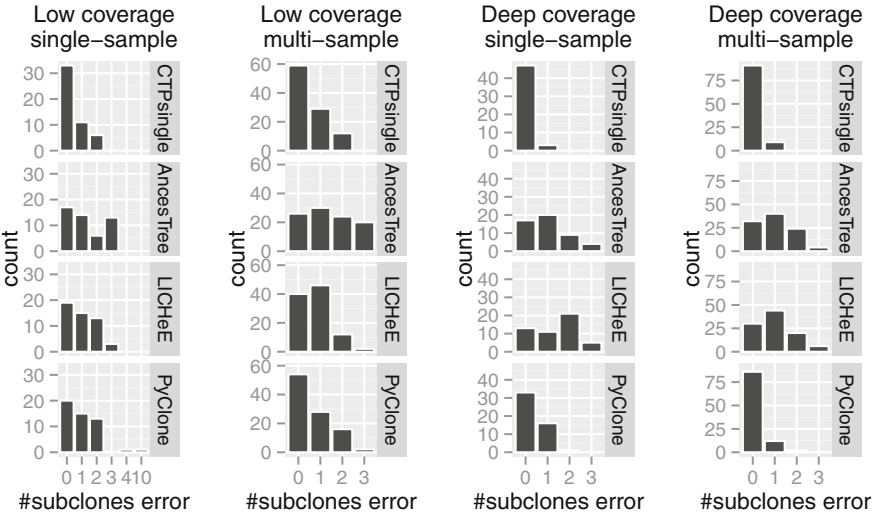


Fig. 2. Comparison of the absolute difference between the true and predicted number of subclones across the simulation experiments. The single-sample experiments contain 50 samples and the multi-sample experiments contain a total of 100 (50×2) samples.

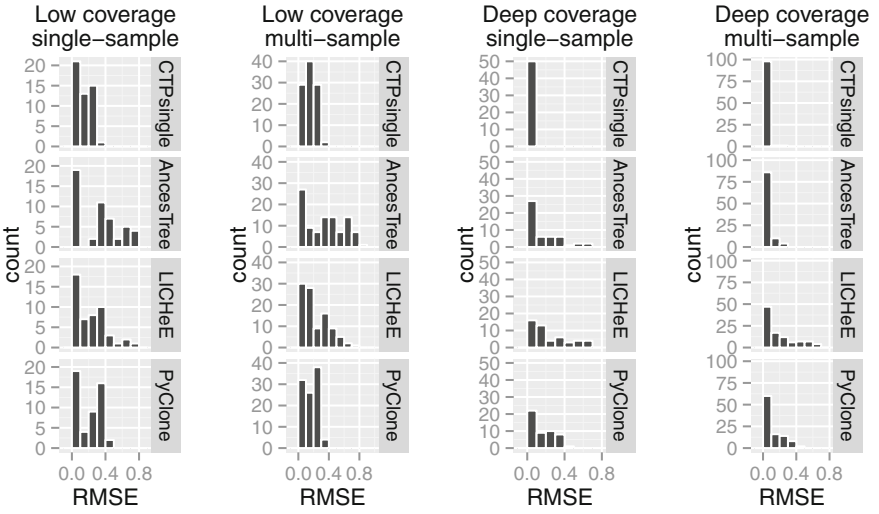


Fig. 3. The histogram of root mean square error (RMSE) values across the simulation experiments. For each sample, RMSE is calculated between the true cancer cell fraction (CCF) of mutations vs the predicted CCF based on the subclone assignments reported by the method. The bin size for the histogram is set to 0.1 for each plot. The single-sample experiments contain 50 samples and the multi-sample experiments contain a total of 100 (50×2) samples.

with very similar frequencies are merged into one cluster. Hence, this measure penalizes methods that tend to merge subclones with very different frequencies. For the deep coverage datasets, CTPsingle has less than 0.1 RMSE.

We also compare CTPsingle with CITUP on the same simulation datasets. To save computation time, we run CITUP only on trees with the correct number of nodes. Despite this advantage, we observe that CTPsingle performs better than CITUP on the low coverage datasets, although the performance of the two methods are very similar on the deep coverage datasets (Fig. S3; Appendix 2).

In terms of running time, we observe that LICHeE is the fastest tool and typically completed within a couple of minutes on our simulation datasets. While CTPsingle is slower than LICHeE, it completed all but a few samples within 30 min and did not exceed one hour on any sample. In contrast, AncesTree and PyClone took several hours on many samples.

3.2 CTPsingle Identifies Subclonal Populations in Prostate Tumors

To illustrate the possible use of CTPsingle in a real clinical setting, we also report the results of CTPsingle on two prostate tumors processed in house. Clinical details of these tumors are given in Table 1. For each patient, fresh frozen (FF) and formalin-fixed parafin embedded (FFPE) tissue taken at the time of radical prostatectomy are subjected to whole-exome sequencing. Since blood samples were not available for these patients, we also obtained whole-exome sequencing data from adjacent benign tissue. Sequencing and mutation calling details are given in Appendix 3. Full clinical details and genomic analysis for these tumours will be described elsewhere.

The percent of the genome with copy number alterations, the number of somatic SNVs and the tumor purity estimated by CTPsingle for each patient are given in Table 1. We remark that due to the high degradation level of FFPE samples, less somatic mutations are detected compared to FF samples. In addition, we determine the copy number alterations - thus copy number neutral regions - using the FF samples only since the copy number segmentations obtained from FFPE samples tend to be noisier.

Despite the difficulty of calling mutations from highly degraded FFPE tissues, we observed a reasonable consistency between the predictions in FF and

Table 1. *Clinical samples.* Below, the Gleason score and prostate specific antigen (PSA) levels are given as measured at diagnosis. Coverage is calculated as the mean total read depth of called mutations. The percent of the genome under copy number alterations (%CNA) are calculated based on the whole-exome sequencing on the fresh frozen sample for each patient. Purity denotes the estimated tumor purity by CTPsingle.

Patient ID	PSA	Gleason score	Clinical stage	Coverage FF (FFPE)	%CNA	#sSNVs FF (FFPE)	Purity FF (FFPE)
DP1566	5.48	9 (4 + 5)	T2c	55.0× (47.7×)	38	241 (166)	0.73 (0.60)
DP1570	5.4	9 (5 + 4)	T1c	61.3× (59.8×)	15	262 (208)	0.37 (0.55)

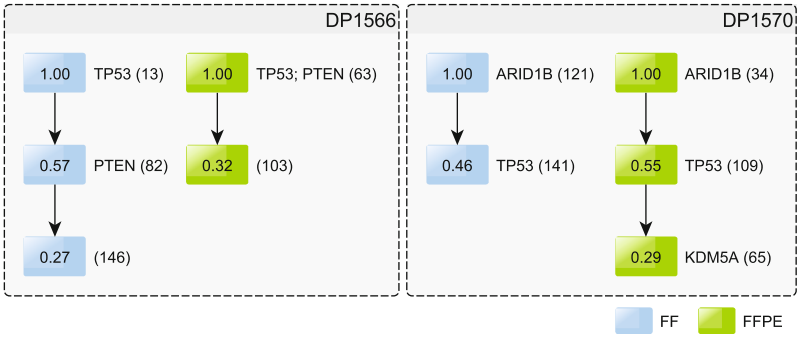


Fig. 4. Subclonal composition of two prostate tumors using fresh frozen and formalin-fixed parafin embedded (FFPE) samples as predicted by CTPsingle. Known cancer-related genes containing non-synonymous coding mutations are shown next to nodes based on their subclonal assignments. The numbers in parentheses give the total number of mutations assigned to each node including synonymous mutations. The PTEN mutations in the FF and FFPE samples of DP1566 represent distinct mutations. The rest of the mutations are identical between FF and FFPE in both patients.

FFPE samples from the same patient. We also remark that confidently calling somatic mutations is particularly challenging in this dataset due to a lack of pure normal sample obtained from blood. Since adjacent benign tissue can be contaminated with cancer cells, less stringent criteria have to be employed while calling mutations resulting in a higher number of false positives in addition to false negatives.

Figure 4 illustrates the subclonal composition of each sample as predicted by CTPsingle. While the number of inferred subclones and their frequencies differ between FF and FFPE samples, the figure shows that the order of the mutations are conserved as expected. Note that, although branching tree topologies are also possible in these samples, these can not be confidently determined without additional samples or information. Therefore, we only report the linear expansion topologies.

In the FF sample of patient DP1566, we see that a TP53 mutation is placed at the starting clone, while a PTEN mutation is placed at a secondary level. In the FFPE sample, in addition to the same TP53 mutation, a distinct PTEN mutation is also placed at the top level suggesting a convergent evolution in these two samples.

In patient DP1570, a TP53 mutation is placed at the secondary level in both FF and FFPE samples. Interestingly, for this patient, CTPsingle predicts an ARID1B mutation at the top level. ARID1B is a protein coding gene involved in transcriptional regulation of select genes by chromatin remodelling and has been previously implicated in pancreatic cancer [9]. This patient also contains a high-impact KDM5A mutation that results in a stop codon loss. KDM5A is a histone demethylase that specifically demethylates Lys-4 of histone H3 and is known to interact with many other proteins including retinoblastoma [10]. While this

mutation is only detected in the FFPE sample, this may be due to the very low tumor purity of the FF sample for this patient. An alternative explanation could be that this mutation is contained by a private subclone that is only present in the FFPE sample.

It is intriguing that this patient contains predicted mutations in two separate genes related to chromatin regulation. As chromatin remodelling is crucially involved in the fine-tuning of cell growth, DNA repair and chromosome segregation, mutations in these genes may act as cancer drivers. Indeed, a comparison of the epigenetic profiles of the tumor samples to that of the adjacent benign samples, as assayed by ChiP-on-ChiP or ChiP-Seq experiments, could provide valuable insight into the progression of this patient's cancer. Validating these mutations and investigating the epigenetic profile of this tumor is part of our plans as future work.

4 Conclusion

In this work, we introduce CTPsingle as a robust alternative to existing clonality inference methods in cases where obtaining multi-sample sequencing from tumors is unfeasible. Using realistic simulations, we show that CTPsingle achieves satisfactory results on single-sample datasets even with low sequencing depths and is able to reconstruct the subclonal composition of tumors with high accuracy on deep sequencing data. Our preliminary results on two prostate tumors also suggest that CTPsingle can help identify the presence of subclones on real exome sequencing data.

While our simulations include a wide range of parameters such as varying tumor purity and number of mutations (including highly disproportionate number of mutations and cancer cell fractions per subclone, Fig. S2; Appendix 2), we acknowledge that real clinical datasets have additional challenges such as DNA degradation and sequencing errors. However, some of these challenges can be mitigated by careful post-processing of sequencing data and are typically taken into account by state-of-the-art mutation calling software. Furthermore, we observe that CTPsingle is still able to accurately infer the subclonal composition of tumors in the presence of a small fraction of false positive SNVs (Fig. S4; Appendix 4). Nevertheless, we acknowledge that a larger cohort with experimental validation is necessary to further demonstrate CTPsingle's performance in a clinical setting.

Like several other tools, CTPsingle is limited to somatic SNVs in copy-neutral regions and hence is not applicable to tumors with high chromosome instability. On the other hand, our experiments suggest that its performance does not deteriorate significantly on moderately copy-number altered genomes (Fig. S4; Appendix 4). Nonetheless, we are currently working to extend our methods to include copy number alterations and hoping to release a comprehensive tool that is applicable to all tumors in the near future.

Acknowledgements. This project was funded by a Prostate Cancer Canada Movember Team grant and the Terry Fox Research Institute New Frontiers Program to CCC; NSERC Discovery Frontiers Grant on the Cancer Genome Collaboratory, Genome Canada Bioinformatics and Computational Biology Program Grant and NSERC Discovery Grant to SCS; NSERC CREATE (139277) fellowship and Vanier Canada Graduate Scholarship to SM.

Appendix

The supplementary material including additional figures are located at https://github.com/nlgndnmz/CTPsingle.

References

1. Buttrey, S.E., et al.: Calling the lp_solve linear program software from r, s-plus and excel. J. Stat. Softw. **14**(4), 1–13 (2005)
2. Deshwar, A.G., Vembu, S., Yung, C.K., Jang, G.H., Stein, L., Morris, Q.: Phylowgs: reconstructing subclonal composition and evolution from whole-genome sequencing of tumors. Genome Biol. **16**(1), 35 (2015)
3. El-Kebir, M., Oesper, L., Acheson-Field, H., Raphael, B.J.: Reconstruction of clonal trees and tumor composition from multi-sample sequencing data. Bioinformatics **31**(12), i62–i70 (2015)
4. Gerlinger, M., Rowan, A.J., Horswell, S., Larkin, J., Endesfelder, D., Gronroos, E., Martinez, P., Matthews, N., Stewart, A., Tarpey, P., et al.: Intratumor heterogeneity and branched evolution revealed by multiregion sequencing. N. Engl. J. Med. **366**(10), 883–892 (2012)
5. Ha, G., Roth, A., Khattra, J., Ho, J., Yap, D., Prentice, L.M., Melnyk, N., McPherson, A., Bashashati, A., Laks, E., et al.: Titan: inference of copy number architectures in clonal cell populations from tumor whole-genome sequence data. Genome Res. **24**(11), 1881–1893 (2014)
6. Hajirasouliha, I., Mahmoody, A., Raphael, B.J.: A combinatorial approach foranalyzing intra-tumor heterogeneity from high-throughput sequencing data. Bioinformatics **30**(12), i78–i86 (2014). Oxford Univ Press
7. Jara, A., Hanson, T., Quintana, F., Müller, P., Rosner, G.: DPpackage: Bayesian semi- and nonparametric modeling in R. J. Stat. Softw. **40**(5), 1–30 (2011). http://www.jstatsoft.org/v40/i05/
8. Jiao, W., Vembu, S., Deshwar, A., Stein, L., Morris, Q.: Inferring clonal evolution of tumors from single nucleotide somatic mutations. BMC Bioinform. **15**(1), 35 (2014)
9. Khursheed, M., Kolla, J., Kotapalli, V., Gupta, N., Gowrishankar, S., Uppin, S., Sastry, R., Koganti, S., Sundaram, C., Pollack, J., et al.: ARID1B, a member of the human SWI/SNF chromatin remodeling complex, exhibits tumour-suppressor activities in pancreatic cancer cell lines. Br. J. Cancer **108**(10), 2056–2062 (2013)
10. Klose, R.J., Yan, Q., Tothova, Z., Yamane, K., Erdjument-Bromage, H., Tempst, P., Gilliland, D.G., Zhang, Y., Kaelin, W.G.: The retinoblastoma binding protein RBP2 is an H3K4 demethylase. Cell **128**(5), 889–900 (2007)
11. MacEachern, S.N.: Computational methods for mixture of dirichlet process models. In: Dey, D., Müller, P., Sinha, D. (eds.) Practical Nonparametric and Semiparametric Bayesian Statistics, vol. 133, pp. 23–43. Springer, New York (1998)

12. Malikic, S., McPherson, A.W., Donmez, N., Sahinalp, C.S.: Clonality inference in multiple tumor samples using phylogeny. Bioinformatics **31**(9), 1349–1356 (2015)
13. Oesper, L., Satas, G., Raphael, B.J.: Quantifying tumor heterogeneity in whole-genome and whole-exome sequencing data. Bioinformatics **30**(24), 3532–3540 (2014)
14. Popic, V., Salari, R., Hajirasouliha, I., Kashef-Haghighi, D., West, R.B., Batzoglou, S.: Fast and scalable inference of multi-sample cancer lineages. Genome Biol. **16**(1), 91 (2015)
15. Prandi, D., Baca, S.C., Romanel, A., Barbieri, C.E., Mosquera, J.M., Fontugne, J., Beltran, H., Sboner, A., Garraway, L.A., Rubin, M.A., et al.: Unraveling the clonal hierarchy of somatic genomic aberrations. Genome Biol. **15**(8), 439 (2014)
16. Roth, A., Khattra, J., Yap, D., Wan, A., Laks, E., Biele, J., Ha, G., Aparicio, S., Bouchard-Côté, A., Shah, S.P.: PyClone: statistical inference of clonal population structure in cancer. Nat. Methods **11**(4), 396–398 (2014)
17. Schuh, A., Becq, J., Humphray, S., Alexa, A., Burns, A., Clifford, R., Feller, S.M., Grocock, R., Henderson, S., Khrebtukova, I., Kingsbury, Z., Luo, S., McBride, D., Murray, L., Menju, T., Timbs, A., Ross, M., Taylor, J., Bentley, D.: Monitoring chronic lymphocytic leukemia progression by whole genome sequencing reveals heterogeneous clonal evolution patterns. Blood **120**(20), 4191–4196 (2012)
18. Sengupta, S., Wang, J., Lee, J., Müller, P., Gulukota, K., Banerjee, A., Ji, Y.: Bayclone: Bayesian nonparametric inference of tumor subclones using NGS data. In: Pacific Symposium on Biocomputing, vol. 20, p. 467. World Scientific (2015)
19. Strino, F., Parisi, F., Micsinai, M., Kluger, Y.: TrAp: a tree approach for fingerprinting subclonal tumor composition. Nucleic Acids Res. **41**(17), e165 (2013). Oxford Univ Press
20. Weinstein, J.N., Collisson, E.A., Mills, G.B., Shaw, K.R.M., Ozenberger, B.A., Ellrott, K., Shmulevich, I., Sander, C., Stuart, J.M., Network, C.G.A.R., et al.: The cancer genome atlas pan-cancer analysis project. Nat. Genet. **45**(10), 1113–1120 (2013)
21. Zare, H., Wang, J., Hu, A., Weber, K., Smith, J., Nickerson, D., Song, C., Witten, D., Blau, C.A., Noble, W.S.: Inferring clonal composition from multiple sections of a breast cancer. PLoS Comput. Biol. **10**(7), e1003703 (2014)

Flexible Modelling of Genetic Effects on Function-Valued Traits

Nicolo Fusi(✉) and Jennifer Listgarten(✉)

Microsoft Research, One Memorial Drive, Cambridge, MA, USA
{fusi,jennl}@microsoft.com

Abstract. Genome-wide association studies commonly examine one trait at a time. Occasionally they examine several related traits with the hopes of increasing power; in such a setting, the traits are not generally smoothly varying in any way such as time or space. However, for function-valued traits, the trait is often smoothly-varying along the axis of interest, such as space or time. For instance, in the case of longitudinal traits like growth curves, the axis of interest is time; for spatially-varying traits such as chromatin accessibility it would be position along the genome. Although there have been efforts to perform genome-wide association studies with such function-valued traits, the statistical approaches developed for this purpose often have limitations such as requiring the trait to behave linearly in time or space, or constraining the genetic effect itself to be constant or linear in time. Herein, we present a flexible model for this problem—the Partitioned Gaussian Process—which removes many such limitations and is especially effective as the number of time points increases. The theoretical basis of this model provides machinery for handling missing and unaligned function values such as would occur when not all individuals are measured at the same time points. Further, we make use of algebraic re-factorizations to substantially reduce the time complexity of our model beyond the naive implementation. Finally, we apply our approach and several others to synthetic data before closing with some directions for improved modelling and statistical testing.

Keywords: Genome-wide association study · Longitudinal traits · Time-series traits · Functional traits · Function-valued traits · Linear mixed models · Gaussian process regression · Radial basis function

1 Introduction

Genome-wide association studies commonly examine one trait at a time. Occasionally they examine several related traits with the hopes of increasing power; in such a setting, the traits are not generally smoothly varying in any way such as time or space. However, with the advent of wearables for health and the "quantified self" movement; the broad deployment of cheap sensors in domains such as agriculture and breeding; and the approaching ubiquity of electronic health records, we shall soon see the ubiquity of function-valued traits. Longitudinal

M. Singh (Ed.): RECOMB 2016, LNBI 9649, pp. 95–110, 2016.
DOI: 10.1007/978-3-319-31957-5_7

traits are one example of function-valued traits—traits which can be viewed as a smooth function of some variable. For example, that variable could be time in a clinical history corresponding to a longitudinal trait, or it could be position in the genome, corresponding to a spatial trait such as chromatin accessibility [1]. Such function-valued traits offer new opportunities to dissect genetics. However, maximally benefiting from such opportunities requires that the rich, smoothly-varying structure within these traits can be leveraged by the statistical model of choice. Rich trait structure arises from constraints in the physical world such as that time moves forward and is smoothly varying, or that the correlation between positions on the genome is slowly decreasing according to genetic distance on the chromosome. Modelling approaches in these settings should take into account such constraints while still allowing for flexibility in the shapes of the traits. Furthermore, it stands to reason that the genetic effect might alter the functional form of a trait, such as the shape of a growth curve, a pattern of weight gain, bone loss, or electrocardiogram signal. Thus, flexible modelling beyond linear genetic effects is also one of our goals. Figure 1 shows a set of simple canonical traits and genetic effects that we would like to be able to detect. These canonical traits will also serve as the basis of our synthetic experiments for comparing the behaviour of several modelling approaches. In these examples, by design, a genetic effect which is constant or linear in time will fail to properly model the data. Although these traits are rather idealized, they present a good starting point with which to examine the problem.

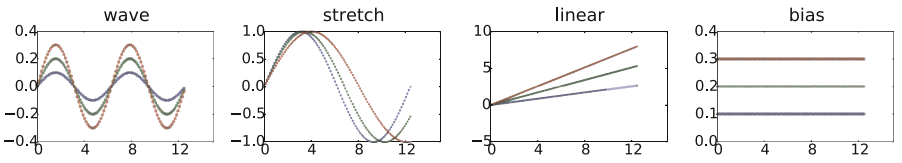


Fig. 1. Simulated traits with 100 time points taking on values uniformly spaced between 0 and 12. Each plot shows what the mean (noise-free) trait looks like for each of the SNP values 0 (blue), 1 (green) and 2 (red). The noise added (not shown) is *iid* with respect to both time and individuals. Note that we hear display the maximum genetic effect for each kind of trait for visual clarity.

The simplest problem one might tackle in our chosen setting is to find out which individual single-nucleotide polymorphisms (SNPs) are correlated to the trait of interest, a so-called marginal test. Those that are correlated are then assumed to have a reasonable probability of being causal for the trait, or of tagging a nearby SNP which is causal for the trait. While it is also of interest to test sets of SNPs jointly [2–4], we here focus on marginal SNP testing, leaving a generalization to set tests for future work. The solution to this marginal testing problem entails (1) proposing a statistical model of the data, and (2) obtaining some weight-of evidence of a genetic effect such as a p-value or Bayes factor. In

this work we focus primarily on the first task but discuss our future directions for the second task in concluding.

Numerous approaches for analyzing function-valued genetic associations have been proposed in recent years [1,4–15]. However, these do not necessarily make effective use of the rich trait structure to increase power because they often assume restrictive forms of the genetic effect or the trait itself. Also, in some cases the statistical efficiency does not scale well with the number of time points, which are expected to be quite numerous in the settings discussed earlier. Next we give a brief overview of some of these approaches and their weaknesses in tackling the kinds of problems we are interested in.

Sikorska *et al.* use an approximate linear mixed model that accounts for correlation in time and assumes that a trait evolves over time in a linear manner; they also assume that the SNP effect itself is additive. Musolf *et al.* first cluster the trait without accounting for genetics and then seek genetic effects on the cluster labels, thereby pre-supposing that all causal SNPs segregate the traits in a similar manner. Shim *et al.* first apply a wavelet-transform to the trait data, thereby transforming the traits to lie in a coordinate system based on (hierarchical) scales and locations; they then perform association testing in this new space. While this approach enables flexible functions of time to be modelled, the SNP effects are restricted to be linear because the wavelet transform itself is linear. Das *et al.* construct a different Legendre polynomial-based model to model the trait for each test SNP allele, learning each model in a largely independent manner. They then test whether the time-specific mean effects are different between the alleles, although it's not clear how they combine time points in their statistical testing framework. Also note that Das *et al.* remove SNPs with minor allele frequency (MAF) less than ten percent from their experiments since the MAF dictates the amount of data available to each allele-specific model. Finally, there has been some related work on detecting differential expression using Gaussian Process regression which shares many aspects of our approach, while differing in several respects including parameter sharing, independence among individuals, and substantial differences in time complexity in the case of aligned time points, partly owing to the use of a different noise model and inference algorithm [16].

In our work, we propose an extremely flexible approach for modelling function-valued traits with genetic effects. In particular, our approach, based on Gaussian Process (GP) regression with a Radial Basis Function (RBF) kernel [17] at its core, can in principle capture any smoothly-varying trait in time, where the smoothness is controlled by a "length scale" parameter. This length scale parameter is estimated using maximum likelihood, thereby effectively deducing the complexity of the trait functional form directly from the data. As for the genetic effect, similarly to Das *et al.*, our model has three components corresponding to three partitions of the data, yielding an extremely non-restrictive class of genetic effects since the GP for each allele can look completely different from the other alleles when no parameters are shared. In our experiments we assume that basic properties such as the noise level and length-scale are likely to be common to all alleles and hence tie these parameters together for more efficient statistical

estimation. However, the model need not be used in this manner. Furthermore, because the RBF kernel effectively integrates out the time points, the number of model parameters does not scale with the number of time points, but is instead fixed–a desirable property when many time points are observed. We call our model the *Partitioned GP* for partitioned Gaussian Process regression.

2 Partitioned Gaussian Processes

As already mentioned our model uses at its core GP regression [17], a class of models which encompasses linear mixed models, the more widely-used concept in genetics [18–21]. The GP regression literature contains results not typically found in the genetics community that we make use of including the use of RBF kernels and Kronecker-product-based refactorizations of matrix-variate normal probability distributions yielding computational efficiencies [22] in the case of aligned and non-missing time points. Also, although we have not yet implemented it, by virtue of using the GP machinery we can immediately access variational approximations to reduce computational time complexity [23,24] in the case of missing data or unaligned time points. We now formally introduce our null model, followed by an exposition of how to do efficient computations in it before introducing the alternative model and computation of p-values.

2.1 Null Model

Our null model, M_0, assumes that the SNP has no effect on the trait (and so does not enter the model), but does capture correlation in time by way of an RBF kernel. Let $\mathbf{Y}$ be the $N \times T$ matrix of traits for N individuals and T time points. Let $\mathbf{W}$ be the $NT \times 1$ times at which the traits were measured, and let $\text{vec}(\mathbf{Y})$ denote the unrolled version of $\mathbf{Y}$ into a vector of dimension $NT \times 1$,

$$\text{vec}(\mathbf{Y}) = \begin{pmatrix} y_{11} \\ y_{21} \\ \vdots \\ y_{NT} \end{pmatrix}$$

Then

$$M_0 : p(\text{vec}(\mathbf{Y})) = \mathcal{N}\left(\text{vec}(\mathbf{Y}) \,\middle|\, \mathbf{0}, \sigma_r^2 \mathbf{K}_{RBF}(\mathbf{W}, \mathbf{W}|l) + \sigma_e^2 \mathbf{I}_{NT}\right), \tag{1}$$

where $\mathcal{N}(a \,|\, b, \mathbf{C})$ is a Gaussian distribution in vector a with mean b and covariance $\mathbf{C}$; $\mathbf{I}_{NT}$ is the $NT \times NT$ identity matrix; σ_r^2 and σ_e^2 are scalar parameters which control the overall variance contributed by each kernel; $\mathbf{K}_{RBF}(l)$ is an $NT \times NT$ radial basis function kernel with length-scale parameter l and elements defined by $K_{RBF}(w_{ij}, w_{qp}|l) \equiv exp\left(-\frac{||w_{ij}-w_{qp}||}{2l^2}\right)$. The length-scale parameter determines the overall scale on which the trait varies within an individual. For very rapidly varying traits, it is small, and for slowly varying traits it is large.

The RBF kernel models the dependence in time while the identity kernel models the remaining environmental noise. Note that the RBF kernel here models not only correlation between time points within an individual but also equally across individuals. That is, we make the assumption that the trait at time point t is more correlated across individuals i and j than between time points t and $t + t_0$ for the same person (where t_0 is an offset in time). While at first this may seem a counterintuitive choice, it turns out that for the types of traits we are interested in, it is the correct thing to do. Namely, we are interested in settings in which the traits are the same across all individuals (or later for those with the same genetics), other than by virtue of noise. Examples of such traits are shown in Fig. 1. An example where this is might be a reasonable assumption would be growth curves where on average the curves look the same for a species, but with a particular mutation the curve suddenly changes trajectory. An example where this is an unreasonable assumption would be un-aligned electrocardiographic signals where no two people would in general look the same at time t unless their signals had been re-scaled and aligned. When the assumption of correlation in time between individuals is not believed to be reasonable, one can easily remove this restriction from the model, leaving time correlations only within an individual. In fact, as we explain in the next section, it is algebraically and computationally trivial to make such a change while retaining all efficient computations. However, by removing this assumption from the model one loses statistical power if the assumption is actually valid in the data. In fact, when conducting our synthetic experiments we found that removal of this assumption in the model substantially weakened the results (data not shown).

Note that for simplicity, we assume that covariates such as age and gender have been regressed out of the trait ahead of time, although these could easily be incorporated in to the model, by way of the Gaussian mean (*i.e.*fixed effects). All remaining expositions (other than for the pseudo-inputs and variational inference) can be readily extended to having covariates directly included with no change to the computational time complexity. We make a similar assumption about population structure and family relatedness, which can be regressed out using either principle components [25] or linear mixed models [21], although investigating the best way to do this for function-valued traits is an open area for investigation. Finally, in Eq. 1 we did not assume that traits for each person were measured at the same time points or that no trait values were missing. However, in the next section on efficient computations, we will need to make this assumption. In Sect. 2.4 we outline ways to relax this assumption.

Efficient Computation of the Likelihood. In order to obtain a p-value by way of statistical testing we need to estimate the maximum likelihood parameters of our model over and over, once per genetic marker. Computing the maximum likelihood over and over again for each hypothesis is a non-trivial goal in the sense that general kernel-based methods have time complexity which scales cubically in the dimension of the kernel (here NT), and space complexity which is quadratic in that dimension. However, in some cases, structure in the kernel

can be leveraged to gain substantial speed-ups (*e.g.* [21]). For Partitioned GPs such structure arises when there is no missing data and all traits are measured at the same time points for all individuals. In this case, the likelihood can be re-written with Kronecker products in the covariance term, yielding dramatically reduced time and space complexities. Later we discuss how to achieve speed-ups in the face of missing or unevenly-spaced time points using the Partitioned GP, which can require some approximations, whereas the present exposition requires no approximation.

The RBF kernel (dimension $NT \times NT$ in Eq. 1) is a specially structured kernel because of the repeating times across individuals. This structure means that we can re-write the Gaussian likelihood in Eq. 1 in matrix-variate form as follows [22],

$$M_0 : p(\mathbf{Y}) = \mathcal{N}\left(\mathbf{Y} \,\middle|\, \mathbf{0}, \sigma_r^2 \mathbf{K}_{RBF}(\mathbf{W}, \mathbf{W}|l) \otimes \mathbf{J}_N + \sigma_e^2 \mathbf{I}_{NT}\right), \tag{2}$$

where here we have overloaded $\mathbf{K}_{RBF}(\mathbf{W}, \mathbf{W}|l)$ to now indicate a $T \times T$ matrix, and where $\mathbf{J}_N$ is the square matrix of all ones of size N. The symbol $\otimes$ denotes the Kronecker product which produces a square matrix of dimensions $ab \times ab$ for $A \otimes B$ if A and B are square matrices of dimension a and b respectively. The computational time complexity of evaluating the likelihood in Eq. 1 is $O(N^3T^3)$ because one must compute the inverse and determinant of the covariance matrix of dimension $NT \times NT$. In contrast, using a spectral-decomposition-based re-factoring [22] of Eq. 2, the computational time complexity can be reduced to $O(T^3)$.[1] In particular, if one defines $\mathbf{U}_r \mathbf{S}_r \mathbf{U}_r^T$ as the spectral decomposition of the $T \times T$ matrix $\mathbf{K}_{RBF}(l)$, and $\mathbf{U}_j \mathbf{S}_j \mathbf{U}_j^T$ as the spectral decomposition of $\mathbf{J}_N$, then one can write the log likelihood of the null model as follows [22]:

$$\mathcal{L}_0 = -\frac{NT}{2}\ln(2\pi) - \frac{1}{2}\ln|\mathbf{S}_r \otimes \mathbf{S}_j| - \frac{1}{2}\mathrm{vec}(\mathbf{U}_r^T \mathbf{Y} \mathbf{U}_j)^T (\mathbf{S}_r \otimes \mathbf{S}_j)^{-1} \mathrm{vec}(\mathbf{U}_r^T \mathbf{Y} \mathbf{U}_j). \tag{3}$$

It is also easy to generalize this expression and its derivative when the mean of the Gaussian is non-zero; we do so to make one of the models we compare against (Furlotte *et al.*) significantly faster than in their original presentation (they could not do the same because they jointly model population structure) [5].

Note that the individuals are not identically and independently distributed (*iid*) in our null model because of the term $\mathbf{J}_N$. If we were to replace $\mathbf{J}_N$ with the identity matrix, then the individuals would be iid, which thus amounts to relaxing the assumption mentioned in the introduction wherein time points across individuals are correlated.

As described earlier, we have assumed that population structure and family structure have already been accounted for, but these could instead be incorporated in to the model by adding to $\mathbf{J}_N$ a genetic similarity matrix [21], incurring a time complexity of $O(N^3 + T^3)$ in the most general case.

[1] If $\mathbf{J}_N$ were an arbitrary matrix the time complexity would be $O(N^3 + T^3)$, but because the spectral decomposition of $\mathbf{J}_N$ can be computed once and cached, the complexity becomes $O(T^3)$. Moreover, because it is an all-ones matrix, its spectral decomposition can be computed more efficiently than in the general case.

For parameter estimation we use gradient descent to obtain the maximum likelihood solution in parameters $l, \sigma_r^2, \sigma_e^2$—all scalars. The reader is referred to Stegle *et al.* for the derivative expressions which have the same time complexity as Eq. 3 [22]. Because the log likelihood is not convex, we use multiple random re-starts, finding empirically that five restarts in our experiments yielded good results.

2.2 Alternative Model

Now that we have fully described the null model and how to efficiently compute its log likelihood, we generalize this model to an alternative model which handles a wide range of genetic effects. To do so, we create a separate GP for each partition of the data, where the partition is defined by the alleles of the test SNP (using whatever encoding of the data one desires, such as a $s = 0, 1, 2$ encoding of the number of mutant alleles across the two chromosomes),

$$M_A : p(\mathbf{Y}) = \sum_{s=1}^{S} \mathcal{N}\left(\mathbf{Y}_s \,\middle|\, \mathbf{0}, \sigma_{r_s}^2 \mathbf{K}_{RBF}(\mathbf{W}, \mathbf{W}|l) \otimes \mathbf{J}_{N_s} + \sigma_e^2 \mathbf{I}_{N_s T}\right), \qquad (4)$$

where S denotes the number of alleles in the SNP encoding, $\mathbf{Y}_s$ is the subset of trait data for which the individual has SNP value s, and where N_s is the number of such individuals. In principle, one could use a different length scale, l and variance parameters σ_e^2 for each partition s, but we have found that in our experiments, tying them together yielded good results and allowed us to test SNPs with much lower MAF owing to the data sharing offered by the shared parameters. While it may seem at first glance that this parameter tying might coerce the trait to look the same across SNP partitions, in fact, we are only coercing broad properties of the trait to be similar, such as the scale on which the signal changes, and only loosely at that. Because GP regression is a non-parametric model, the data itself plays a large role in defining the posterior distribution of functional forms; it is for this reason that our model is able to capture substantially different functional forms even with tied parameters.

The same efficient computations outlined earlier for the null model can just as well be applied to this alternative model, and so the time complexity of computing the alternative model likelihood has as an upper bound that of the null model, which happens only when all individuals are assigned to the same partition. Note too that the null model can computed just once and then cached across all SNPs tested for increased efficiency.

Beyond data sharing across partitions by virtue of shared parameters, the model has good statistical efficiency owing to the fact that GPs operate in the kernel space [17] where the number of parameters does not depend on the number of time points. All in all, we find in our experiments that as few as seven samples per partition appears to be sufficient, which with cohort sizes in the tens if not hundreds of thousands, imposes little restriction on the MAF.

2.3 Hypothesis Testing

Standard frequentist hypothesis testing uses a null model that is nested in the alternative model which then allows one to use a likelihood ratio or score test, for example. However, even when models are nested, these tests require that model assumptions are met, and typically that sample sizes are large enough for asymptotics to be valid. In cases where model or asymptotic assumptions are unmet, one can appeal to various forms of permutation testing to obtain calibrated p-values. Because our models are not nested, we cannot rely on standard theories to compute p-values, and could therefore turn to permutation testing. However, as it turns out, when we apply a standard χ^2 test to generate p-values for our Partitioned GP, we find that our type 1 error is controlled, albeit extremely conservatively even though the assumptions of this test are not here met (see Results). Furthermore, in the discussion, we outline a nested version of the Partitioned GP that we are currently working on.

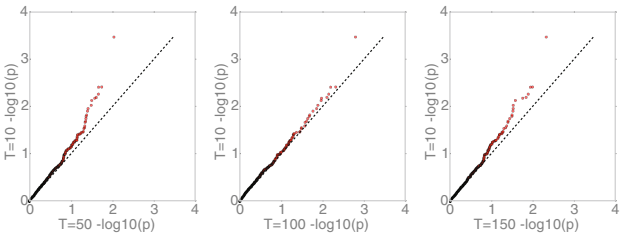


Fig. 2. Paired plot of the $-log$ p-values generated from the null distribution, for 10 time points versus each of 100 and 150 time points.

The precise way in which we apply a standard χ^2 test is that we compute the maximum likelihood of the data under the null and under the alternative models, $\mathcal{L}_A$ and $\mathcal{L}_0$, count the number of degrees of freedom different between them, d, and then apply the standard p-value computation. Our null model has no partitions and has three free scalar parameters: σ_r^2 and l, the overall-variance and length-scale for the time-based kernel, and σ_e^2 for the residual noise. Our alternative model shares all parameters across partitions except for the time-based kernel variances, $\sigma_{r^s}^2$ (one per SNP allele), leading to two more parameters than the null model. We count these two parameters as two extra degrees of freedom even though these parameters are constrained to be greater than zero and so are not truly full degrees of freedom—such miscounting can only lead to overly-conservative p-values in the case of properly nested models. Our test statistic is then twice the difference between the null and alternative maximum log likelihoods, $\Delta \equiv 2(\mathcal{L}_A - \mathcal{L}_0)$, from which we compute a p-value using a χ_d^2 test with $d = 2$ of freedom. While this p-value is uncalibrated, as we shall see in the Results section, it turns out to control type 1 error.

Table 1. Control of type 1 error at significance thresholds α for traits with 10 time points using 390,272 tests. Fraction of p-values below that threshold, with absolute numbers in parentheses.

Model	$\alpha = 10^{-2}$	$\alpha = 10^{-3}$	$\alpha = 10^{-4}$	$\alpha = 10^{-5}$
Partitioned GP	$1.1 \times 10^{-3}(434)$	$6.7 \times 10^{-5}(26)$	0.0(0)	0.0(0)
Inverse K score	$9.1 \times 10^{-3}(3568)$	$8.8 \times 10^{-4}(342)$	$5.6 \times 10^{-5}(22)$	$1.0 \times 10^{-5}(4)$
Inverse linreg	$9.8 \times 10^{-3}(3828)$	$9.4 \times 10^{-4}(366)$	$6.1 \times 10^{-5}(24)$	$1.0 \times 10^{-5}(4)$
Furlotte et al.	$9.2 \times 10^{-3}(3589)$	$9.3 \times 10^{-4}(362)$	$6.1 \times 10^{-5}(24)$	$1.5 \times 10^{-5}(6)$

2.4 Handling Traits with Missing Data or Which are Unevenly-Sampled Across Individual

In a model with a vector Gaussian likelihood, such as Eq. 1, missing trait data can readily be handled by simply removing any rows with missing data, because this procedure is equivalent to marginalization in a Gaussian [17]. In such a manner, if using Eq. 1, one could take T to be the number of uniquely observed time points across all individuals, even if many individuals were missing many of these time points. This procedure could also capture the case where different individuals were measured at different time points. However, in the Kronecker version of the likelihood written for computational efficiency gains (Eq. 2), one can no longer perform this arbitrary marginalization by simply removing an element of the phenotype vector, because with the Kronecker-factorized covariance matrix one would have to either remove all individuals missing a time point, or all time points missing an individual. Therefore, if one wants both computational efficiency and a means to readily marginalize over missing data, one must appeal to alternative formulations and/or approximations. The approach we propose is keep the Gaussian likelihood in vector form, as in Eq. 1, but to augment the model with latent *inducing inputs* [23,24], which are points in time (or space, depending on the type of trait) that are included in the model. Inducing inputs can be thought of as pseudo-observations in time (or space) that are included in the RBF kernel inputs; when conditioned on, these pseudo-observations make any observed data conditionally independent of each other. This has the effect of reducing the time complexity from $O((NT)^3)$ in Eq. 1 to $O(NTQ^2)$ for Q inducing inputs. In such a variational approach, only the number of pseudo-observations need be specified, not the locations, as these are learned as part of the parameter estimation procedure. Also note that if one uses as many pseudo-observations as there are uniquely observed time points, then the algorithm is exact. As a consequence, one could use this approach as an alternative to the efficient Kronecker product approach we described. We have not yet performed experiments with this approach, but these methods are well-studied and their application should be rather direct.

3 Results

As discussed in the introduction, many models have been developed to perform genome-wide association studies with function-valued traits. However, these models tend to have constraints on the type of genetic or time effect that can be recovered (*e.g.*, only constant or linear effect in time, or only linear in the SNP), or are limited to relatively few time points because the number of parameters scales with the number of time points. For our experiments we have chosen a set of baseline models to test particular hypotheses about what kinds of models work and where they fail, in the settings we care about—in particular, exploring what happens when there are a large number of time points such as would be collected be wearables and other sensors. The models we compare and their short-hand notation are:

1. *Partitioned GP*: As described above, using the (exact) Kronecker product implementation.
2. *Furlotte et al.*: A linear mixed model where correlation in time is modelled using an auto-correlation kernel (here we use an RBF as we do with our Partitioned GP), and where in the alternative model, the SNP is a fixed effect, shifting the trait at all time points by the same amount [5]. A standard LRT test is used for the one-degree-of-freedom test. Note that we here do not use the population structure kernel used in [5] as our experiments are not affected by such factors.
3. *Inverse linreg*: To examine how models for which the number of parameters increases with the number of time points, we use inverse linear regression model wherein the SNP is modelled as the dependent variable and each trait in time is an independent variable. Testing is done with a χ^2 test with T degrees of freedom (total number of time points, assumed to be the same for all individuals). Note that in place of inverse linear regression, we could have used inverse multinomial/"soft-max" regression. However, because preliminary results suggested the results were similar, we chose to experiment with only the linear model.
4. *Inverse K score*: This model can be viewed as a Bayesian equivalent to *Inverse linreg* where the time-effects are integrated out, yielding a linear mixed model. In this way, the model does not dependend on the number of time points. We then apply a score test to obtain a p-value (*e.g.* [2]).

We systematically explore each of these approaches on simulated phenotypic data where we know the ground truth, examining type 1 error control, power, and ability to rank hypotheses regardless of calibration. We based our simulated data on the actual SNPs in the CARDIA data set (dbGaP phs000285.v3.p2) which, after filtering out individuals missing more than 10 % of their SNPs, any SNPs missing more than 2 % of individuals, or with MAF less than 5 % left 1,441 individuals with 540,038 SNPs. The only covariate we use is an off-set, which we regress on as a pre-processing step before applying the models.

To simulate time-varying traits, we used a set of canonical functions that were representative of the types of signal we were interested in exploring. In

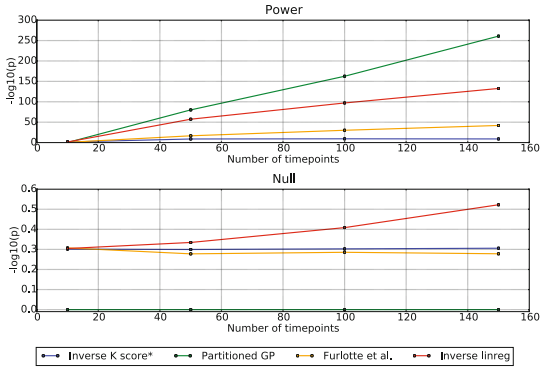


Fig. 3. Power curves as a function of time for all methods. On the vertical axis is the median $-log$ p-value for each method over eight thousand SNP tests. The top plot is for tests with SNP effect, and the lower plot is for those with no SNP effect. *As noted in the main text, the numerical routine used to get p-values for Inverse K score does not yield numeric values less than around 10^{-8}, thereby likely making this method appear worse than it might be; however, we get a better sense of its behaviour in Fig. 5.

particular, we used a *wave*, *linear*, *bias*, and a *stretch* as shown in Fig. 1. For null data, we generated noisy versions of these, where the noise was iid in time and individual. For non-null data we modified the noise-free trait in a smoothly varying way as a function of genotype before adding iid noise. For the wave (a *sin* wave), the amplitude increased as a linear function of the SNP; for the linear (a straight line), the slope changed as a linear function of the SNP; for the bias, the horizontal intercept changed as a linear function of the SNP; for stretch (a *sin* wave), the frequency changed as a linear function of the SNP. We varied both the SNP effect intensities and the amount of noise. One can summarize the strength of the SNP effect at each time point by the fraction of variance explained by the genetic signal at each time point (*i.e.*, the variance of the noiseless trait divided by total variance, all at a given time point) as shown in Fig. 4. Because we were interested specifically in seeing which models could handle many time points, we conducted experiments with 10, 50, 100, and 150 time points.

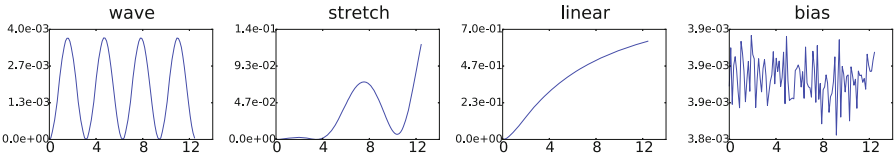


Fig. 4. Average fraction of variance accounted for by genetics at each time point in each canonical function over the range of settings used, for the traits with 100 time points.

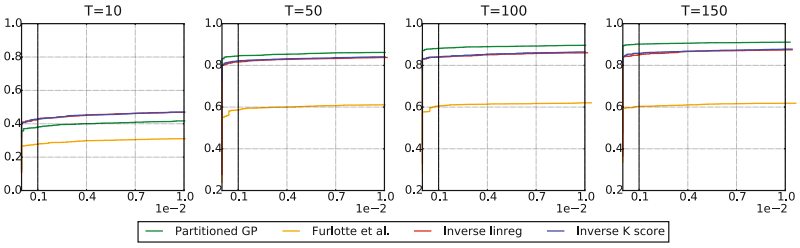
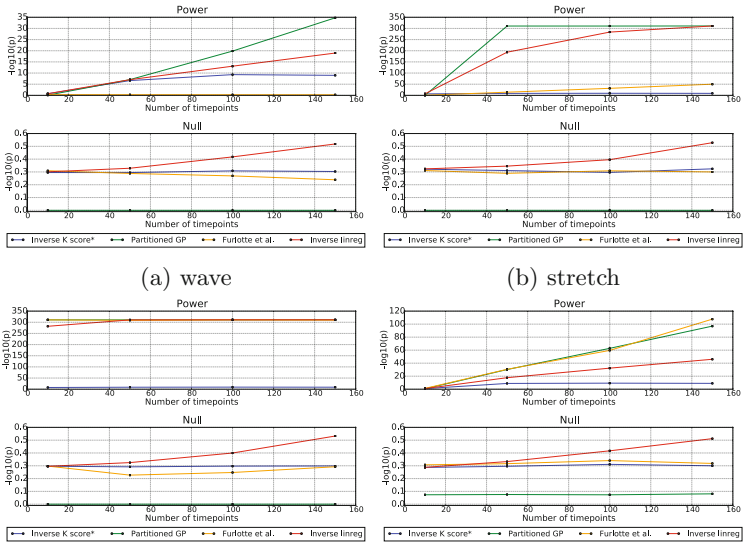


Fig. 5. ROC curves for the simulated data with T equal to 10, 50, 100 and 150 time points, for small False Positive Rates (less than 0.01). The vertical axis shows the False Positive Rate, and the horizontal axis, the True Positive Rate.

Our first goal was to establish whether our Partitioned GP controls type 1 error so that we could use its p-values at face value for power comparisons, even if they are not calibrated. First we used 8,000 tests at each of 10, 50, 100 and 150 time points, finding that the smallest number of time points (10) was always the least conservative (Fig. 2). Therefore, we ran much larger scale simulations of null-only data for 10 time points, obtaining 390,272 test statistics. With just under half a million tests, we had resolution to check for control of type 1 error up to a significance level of $\alpha = 10^{-5}$. As can be seen in Table 1, all methods control the type 1 error up to $\alpha = 10^{-5}$. Note that our method controls the type 1 error extremely conservatively, which could potentially hurt our method in a power comparison. However, as we see next, our method is still the most powerful overall in our experiments.

Having established that our method controls type 1 error, we next set out to see if it had more power to detect associations than the other methods. Figure 3 shows the median test statistic for both our null (lower plot) and non-null (upper plot) experiments, and demonstrates that our methods has maximum power for the traits and methods chosen. Because our type 1 error control experiments only went to $\alpha = 10^{-5}$, we chose to include the lower plot (Null). This null plot shows that while the inverse kernel score remains calibrated, the inverse linear regression becomes substantially inflated, failing to control the type 1 error. Our method, is extremely conservative in controlling the type 1 error, yet maintains maximal power. We also break down these plots by trait type in Fig. 6. Here we see that Furlotte *et al.*, despite only modelling a mean shift in the trait, is able to capture stretch, though not wave, for which the mean between alleles is identical. For stretch and wave, the Partitioned GP is the clear winner, while for linear, all method work equally well, and for stretch, Furlotte *et al.* and the Partitioned GP have the most power.

Note that the inverse kernel score test appears to have terrible power. However, this plot is perhaps misleading in the sense that this method uses a numerical routine (Davies method) which has limited precision, yielding many zeros for tiny p-values (usually those smaller than 10^{-8}). The only way to handle this was either to keep these at zero, which would give that method an unfair advantage, or to replace all zero p-values with 10^{-8}, which is what we chose to do, thereby



(a) wave (b) stretch

(c) linear (d) bias

Fig. 6. Power curves as in Fig. 3, but separated by trait types shown in Fig. 1. *Again, as noted in the main text, the numerical routine used to get p-values for Inverse K score does not yield numeric values less than around 10^{-8}, thereby likely making this method appear worse than it might be; however, we get a better sense of its behaviour in Fig. 5.

showing the model in a worse light with respect to power than we believe it may have if there were a way to compute p-values with more precision. As a consequence, we next investigated the ability of each model to discriminate true nulls from alternatives by using a Receiver Operating Characteristics (ROC) curve—a metric which does not depend in any way on calibration and may be less sensitive to p-value resolution.

Figure 5 shows the ROCs for each method, where we now see that the inverse kernel score test performs extremely well, though not as well as the Partitioned GP. Note that inverse linear regression, though showing inflated test statistics in the lower panel of Fig. 3, here demonstrates that it maintains the ability to properly rank the hypotheses from most to least significant, though again, not as well as the Partitioned GP. Note that the performance of Furlotte *et al.* is not terribly surprising since it is only able to capture shifts in the mean of the functional trait, whereas our simulation scheme is deliberately testing richer SNP effects.

4 Discussion

We have introduced a new method for performing GWAS on function-valued traits. Our model is extremely flexible in its capacity to handle a wide range of

functional forms. This flexibility is achieved by using a non-parametric statistical model based on RBF Gaussian processes. Computations in this model are efficient when time points are aligned and traits are not missing, scaling only cubically with the number of time points as opposed to cubic in the number of time points times individuals, as would be the case in a naive computation. We have also outlined how to do efficient computations even in the presence of missing trait data or unaligned samples. In a comparison against three other models on synthetic data, each with different characteristics and ways of handling the problem, we achieved maximal power, and maximal ability to discriminate null versus alternative tests as judged by an ROC curve. Our model is especially good at handling traits with many time points.

One downside of the model as presented is that the null model is not nested inside the alternative model, making computation of calibrated p-values without permutations most likely impossible. We were able to bypass this issue by demonstrating empirically that naive application of a likelihood ratio test controls the type 1 error, yielding extremely conservative p-values. However, we are currently investigating a version of the Partitioned GP model which has its null model nested in the alternative model and is therefore likely to yield calibrated p-values and therefore potentially a larger power gain. In this model, the partitions of the alternative model are all placed within a single Gaussian, with correlation parameters for each pair of alleles dictating how similar the GP for each allele should be. When these parameters are equal to one, we obtain the present alternative model. When these parameters are zero, we obtain the null model, thereby making it nested inside of the alternative. Other directions of interest are to extend this type of modelling approach to testing sets of SNPs rather than only single SNPs, and to incorporate model-based warping of the phenotype so as to coerce the data to better adhere to the Gaussian residual assumption [26].

Acknowledgments. We thanks to Leigh Johnston, Ciprian Crainiceanu, Bobby Kleinberg and Praneeth Netrapalli for discussion; the anonymous reviewers for helpful feedback, and Carl Kadie for use of his HPC cluster code. Funding for CARe genotyping was provided by NHLBI Contract N01-HC-65226.

References

1. Shim, H., Stephens, M.: Wavelet-based genetic association analysis of functional phenotypes arising from high-throughput sequencing assays. Ann. Appl. Stat. **9**(2), 665–686 (2015)
2. Wu, M.C., Kraft, P., Epstein, M.P., Taylor, D.M., Chanock, S.J., Hunter, D.J., Lin, X.: Powerful SNP-set analysis for case-control genome-wide association studies. Am. J. Hum. Genet. **86**(6), 929–942 (2010)
3. Listgarten, J., Lippert, C., Kang, E.Y., Xiang, J., Kadie, C.M., Heckerman, D.: A powerful and efficient set test for genetic markers that handles confounders. Bioinformatics **29**(12), 1526–1533 (2013)

4. He, Z., Zhang, M., Lee, S., Smith, J.A., Guo, X., Palmas, W., Kardia, S.L.R., Diez Roux, A.V., Mukherjee, B.: Set-based tests for genetic association in longitudinal studies. Biometrics **71**(3), 606–615 (2015)
5. Furlotte, N.A., Eskin, E., Eyheramendy, S.: Genome-wide association mapping with longitudinal data. Genet. Epidemiol. **36**(5), 463–471 (2012)
6. Smith, E.N., Chen, W., Kähönen, M., Kettunen, J., Lehtimäki, T., Peltonen, L., Raitakari, O.T., Salem, R.M., Schork, N.J., Shaw, M., Srinivasan, S.R., Topol, E.J., Viikari, J.S., Berenson, G.S., Murray, S.S.: Longitudinal genome-wide association of cardiovascular disease riskfactors in the Bogalusa heart study. PLoS Genet. **6**(9), e1001094 (2010)
7. Jaffa, M., Gebregziabher, M., Jaffa, A.A.: Analysis of multivariate longitudinal kidney function outcomes using generalized linear mixed models. J. Transl. Med. **13**(1), 192 (2015)
8. Das, K., Li, J., Wang, Z., Tong, C., Guifang, F., Li, Y., Meng, X., Ahn, K., Mauger, D., Li, R., Rongling, W.: A dynamic model for genome-wide association studies. Hum. Genet. **129**(6), 629–639 (2011)
9. Sikorska, K., Montazeri, N.M., Uitterlinden, A., Rivadeneira, F., Eilers, P.H.C., Lesaffre, E.: GWAS with longitudinal phenotypes: performance of approximate procedures. Eur. J. Hum. Genet. **23**, 1384–1391 (2015)
10. Ding, L., Kurowski, B.G., He, H., Alexander, E.S., Mersha, T.B., Fardo, D.W., Zhang, X., Pilipenko, V.V., Kottyan, L., Martin, L.J.: Modeling of multivariate longitudinal phenotypes in family geneticstudies with Bayesian multiplicity adjustment. BMC proceedings **8**(Suppl 1), S69 (2014)
11. Musolf, A., Nato, A.Q., Londono, D., Zhou, L., Matise, T.C., Gordon, D.: Mapping genes with longitudinal phenotypes via Bayesian posterior probabilities. BMC Proc. **8**(Suppl 1), S81 (2014)
12. Wang, T.: Linear mixed effects model for a longitudinal genome wideassociation study of lipid measures in type 1 diabetes linear mixed effectsmodel for a longitudinal genome wide association study of lipid measures in type 1 diabetes. Master's thesis, McMaster University (2012)
13. Zhang, H.: Multivariate adaptive splines for analysis of longitudinal data. J. Comput. Graph. Stat. **6**, 74–91 (1997)
14. Kendziorski, C.M., Cowley, A.W., Greene, A.S., Salgado, H.C., Jacob, H.J., Tonellato, P.J.: Mapping baroreceptor function to genome: a mathematical modeling approach. Genetics **160**(4), 1687–1695 (2002)
15. Chung, W., Zou, F.: Mixed-effects models for GAW18 longitudinal blood pressure data. BMC Proc. **8**(Suppl 1), S87 (2014)
16. Stegle, O., Denby, K.J., Cooke, E.J., Wild, D.L., Ghahramani, Z., Borgwardt, K.M.: A robust Bayesian two-sample test for detecting intervals of differential gene expression in microarray time series. J. Comput. Biol. J. Comput. Mol. Cell Biol. **17**(3), 355–367 (2010)
17. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press, Cambridge (2005)
18. Yu, J., Pressoir, G., Briggs, W.H., Vroh Bi, I., Yamasaki, M., Doebley, J.F., McMullen, M.D., Gaut, B.S., Nielsen, D.M., Holland, J.B., Kresovich, S., Buckler, E.S.: A unified mixed-model method for association mapping that accounts for multiple levels of relatedness. Nat. Genet. **38**, 203–208 (2006)
19. Kang, H.M., Zaitlen, N.A., Wade, C.M., Kirby, A., Heckerman, D., Daly, M.J., Eskin, E.: Efficient control of population structure in model organism association mapping. Genetics **178**(3), 1709–1723 (2008)

20. Listgarten, J., Kadie, C., Schadt, E.E., Heckerman, D.: Correction for hidden confounders in the genetic analysis of gene expression. Proc. Nat. Acad. Sci. **107**(38), 16465–16470 (2010)
21. Lippert, C., Listgarten, J., Liu, Y., Kadie, C.M., Davidson, R.I., Heckerman, D.: FaST linear mixed models for genome-wide association studies. Nat. Methods **8**(10), 833–835 (2011)
22. Stegle, O., Lippert, C., Mooij, J.M., Lawrence, N.D., Borgwardt, K.M.: Efficient inference in matrix-variate gaussian models with \iid observation noise. In: Shawe-Taylor, J., Zemel, R.S., Bartlett, P.L., Pereira, F., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 24, pp. 630–638. Curran Associates Inc. (2011)
23. Candela, J.Q., Rasmussen, C.E.: A unifying view of sparse approximate gaussian process regression. J. Mach. Learn. Res. **6**, 1939–1959 (2005)
24. Titsias, M.K.: Variational learning of inducing variables in sparse Gaussian processes. Artif. Intell. Stat. **12**, 567–574 (2009)
25. Price, A.L., Patterson, N.J., Plenge, R.M., Weinblatt, M.E., Shadick, N.A., Reich, D.: Principal components analysis corrects for stratification in genome-wide association studies. Nat. Genet. **38**(8), 904–909 (2006)
26. Fusi, N., Lippert, C., Lawrence, N.D., Stegle, O.: Warped linear mixed models for the genetic analysis of transformed phenotypes. Nature Communications, 5:4890 (2014)

MetaFlow: Metagenomic Profiling Based on Whole-Genome Coverage Analysis with Min-Cost Flows

Ahmed Sobih, Alexandru I. Tomescu(✉), and Veli Mäkinen

Helsinki Institute for Information Technology HIIT,
Department of Computer Science,
University of Helsinki, Helsinki, Finland
tomescu@cs.helsinki.fi

Abstract. High-throughput sequencing (HTS) of metagenomes is proving essential in understanding the environment and diseases. State-of-the-art methods for discovering the species and their abundances in an HTS sample are based on genome-specific markers, which can lead to skewed results, especially at species level. We present MetaFlow, the first method based on coverage analysis across entire genomes that also scales to HTS samples. We formulated this problem as an NP-hard matching problem in a bipartite graph, which we solved in practice by min-cost flows. On synthetic data sets of varying complexity and similarity, MetaFlow is more precise and sensitive than popular tools such as MetaPhlAn, mOTU, GSMer and BLAST, and its abundance estimations at species level are two to four times better in terms of ℓ_1-norm. On a real human stool data set, MetaFlow identifies *B.uniformis* as most predominant, in line with previous human gut studies, whereas marker-based methods report it as rare. MetaFlow is freely available at http://cs.helsinki.fi/gsa/metaflow.

1 Introduction

Microbes—microscopic organisms that cannot be seen by the eye, which include bacteria, archaea, some fungi, protists and viruses—are found almost everywhere, in the human body, air, water and soil, and play a vital role in maintaining the balance of ecosystems. For example, diazotrophs are solely responsible for the nitrogen fixation process on earth [9], and half of the oxygen on earth is produced by marine microbes [11].

Metagenomic sequencing allows studying microbes sampled directly from the environment without prior culture. A fundamental analysis of a metagenomic sample is finding what species it contains (the sample richness) and what are their relative abundances. This is a challenging task due to the similarity between the species' genomes, sequencing errors, and the incompleteness of reference microbial databases.

S. Ahmed and A.I. Tomescu—Equal contribution.

M. Singh (Ed.): RECOMB 2016, LNBI 9649, pp. 111–121, 2016.
DOI: 10.1007/978-3-319-31957-5_8

One of the first methods, applicable only at high taxonomic levels [7], focused on 16S ribosomal RNA marker genes. Its limitations have been mitigated by high-throughput sequencing of the entire genomes in a metagenomic sample, and a number of methods dealing with this data have been proposed. The simplest of them is to align the reads to a reference database, using e.g. BLAST [1], and to choose the best alignment for every read. This approach cannot break the tie between multiple equally-good alignments of a read, and it cannot detect false positive alignments. One way of avoiding alignment ties, but not false positive alignments, is to estimate the sample structure only at a high taxonomic level. For example, MEGAN [4] assigns each read with ties to the lowest common ancestor of its alignments in the reference taxonomic tree. Another method for breaking ties was proposed in PhymmBL [2], based on Interpolated Markov Models (IMMs). For each read, it combines the BLAST alignment score to a particular genome, with another score based on the probability of the read being generated by an IMM of that genome. This results in a single maximum-scoring alignment for every read, which improves over basic BLAST alignments. This method still cannot eliminate false alignments, and does not scale to high-throughput sequencing samples. For example, it takes one hour to classify 5,730 reads of length 100 bp [2].

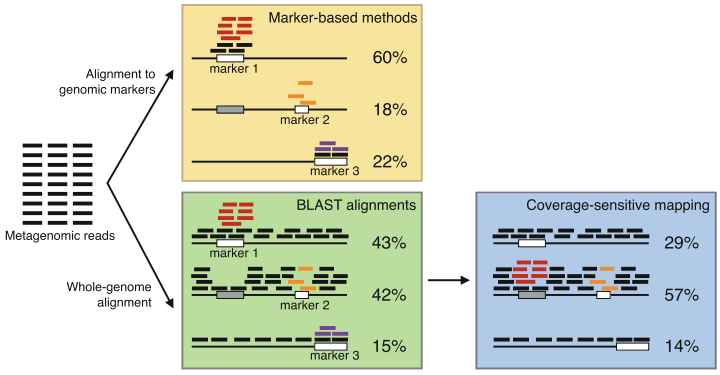


Fig. 1. Overview of the methods compared in this paper. In the yellow box, reads are aligned only to genomic markers, and the relative abundances are highly skewed: marker 1 receives more false mappings (in red) because it is similar with a subsequence of the second genome (gray); marker 2 has a drop in coverage due to a sequencing bias, and it is covered only by three reads (orange); marker 3 is covered also by some reads sequenced from a species not in the reference database (violet). In the green box, reads have whole-genome BLAST alignments, but the relative abundances are still skewed: the tie in the alignment of the red reads is not resolved, and the violet reads from an unknown species aligning to the third genome are not removed. In the blue box, the coverage-sensitive mapping of the reads: the red reads are correctly aligned to the second genome (in the gray sequence) and the violet reads from an unknown species are discarded from the third genome.

The current state-of-the-art approach is to construct a small curated reference database of genomic markers. These markers can be clade-specific marker genes (MetaPhlAn [12]), universal marker genes (mOTU [14]), or genome-specific markers not restricted to coding regions (GSMer [15]). The reads are aligned only to these marker regions, which makes it a fast process, and the estimations are more accurate at the species level because of the marker curation process. However, due to the short length of the marker regions, the abundance estimations can be extremely skewed in some cases. In addition, the markers are uniquely identifying the microbial genomes only among the currently known genomes, meaning that the entire database of markers must be re-computed with the addition of each new reference microbial genome.

In this paper, we propose a new method which addresses the problems of equally-good alignments and of false alignments, with accurate estimates at the species level. Our method takes into account the entire read alignment landscape inside all genomes in the reference database. The main idea is to exploit the assumption that if enough reads are sampled, then these reads will cover most of the genomic regions. In addition, not relying on specific genomic markers, there is no need to curate a reference database. Our problem formulation is based on a matching problem applied to a bipartite graph constructed from read alignments. This problem is NP-hard, but we give a practical strategy for solving it with min-cost network flows. See Fig. 1 for an overview of all methods.

We performed experiments on synthetic and real data sets, and compared MetaFlow with popular tools MetaPhlAn [12], mOTU [14], GSMer [15], and standard BLAST alignments. MetaFlow outperforms the other methods in its ability to correctly identify the species and their abundance levels. On synthetic data, MetaFlow's predictions are more precise and sensitive, and its abundance estimations are up to two to four times better in terms of ℓ_1-norm. On a real fecal metagenomic sample from the Human Microbiome Project, MetaFlow reports *B.uniformis* as predominant species, in line with previous human gut studies [8]. However, marker-based methods MetaPhlAn and mOTU assign it a low abundance.

2 Methods

We assume in input a set of BLAST hits of the metagenomic reads inside a collection of reference genomes, which we call *known genomes*. The output is the richness of the sample and the relative abundance of each known species. This is obtained by selecting, for every read, exactly one hit in a reference genome, or classifying it as originating from a species not in the reference database (we call such species *unknown*).

The optimal selection is the one simultaneously achieving the following three objectives: (1) few coverage gaps, (2) uniform coverage, and (3) agreement between BLAST scores and final mappings. Objective (1) allows the detection of outlier genomes that have only few regions covered. Objective (2) breaks ties between read alignments, and is also based on detecting abnormal read coverage patterns.

We model the above-mentioned input for this problem as a bipartite graph, such that the reads form one part of the bipartition, and the reference genomes form the other part of the bipartition. Objectives (1)–(3) are not modeled independently, but combined in a single objective function, as discussed in the next section. Our modeling is inspired by the interesting Coverage-sensitive many-to-one min-cost bipartite matching problem, introduced in [5] for mapping reads to complex regions of a reference genome. We extended this model to our metagenomic context, since reads can have mappings to more than one reference genome, or can originate from unknown species.

2.1 Problem Formulation and Computational Complexity

Assume that the reads have BLAST hits in the collection of reference genomes $\mathcal{G} = \{G^1, \ldots, G^m\}$. We partition every genome G^i into substrings of a fixed length L, which we call *chunks*. Denote by s_i the number of chunks that each genome G^i is partitioned into. We construct a bipartite graph $G = (A \cup B, E)$, such that the vertices of A correspond to reads, and the vertices of B correspond to the chunks of all genomes $G^1, \ldots, G^m$. Specifically, for every chunk j of genome G^i, we introduce a vertex y^i_j, and we add an edge between a read $x \in A$ and chunk $y^i_j \in B$ if there is a BLAST mapping of read x starting inside chunk j of genome G^i. This edge is assigned the cost of the mapping (BLAST scores can be trivially transformed to costs), which we denote here by $c(x, y^i_j)$. In order to model the fact that reads can originate from unknown species (whose genome is not present in the collection $\mathcal{G}$), we introduce an 'unknown' vertex z in B, with edges from every read $x \in A$ to z, and with a fixed cost $c(x, z) = \gamma$, where γ is appropriately initialized.

In the *coverage-sensitive metagenomic mapping* problem stated below, the tasks are: first, for each G^i, to find the number of reads sequenced (i.e., originating) from it, which we denote by r_i (r_i is 0 if G^i is an outlier); second, to select an optimal subset $M \subseteq E$ such that for every read $x \in A$ there is exactly one edge in M covering it (a mapping of x). These must minimize the sum of the following two costs:

(A) the sum, over all chunks of every genome G^i, of the absolute difference between r_i/s_i and the number of read mappings it receives from M (corresponding to Objective (2));
(B) the sum of all edge costs in M (corresponding to Objective (3)).

Our formal problem definition is given below. We use the following notation: n is the number of reads, m is the number of different genomes where the reads have BLAST hits, s_i is the number of chunks of each genome G^i, $i \in \{1, \ldots, m\}$, and in a graph $G = (V, E)$, $d_M(v)$ denotes the number of edges of a set $M \subseteq E$ incident to a vertex v.

Coverage-sensitive metagenomic mapping. Input:
- a bipartite graph $G = (A \cup B, E)$, where A is the set of n reads, $B = \{y_1^1, \ldots, y_{s_1}^1, \ldots, y_1^m, \ldots, y_{s_m}^m\} \cup \{z\}$ is the set of all genome chunks plus z, the 'dummy' node,
- a cost function $c : E \to \mathbb{Q}$,
- constants $\alpha \in (0, 1)$, $\beta, \gamma \in \mathbb{Q}_+$.

Tasks:
- find a vector $R = (r_1, \ldots, r_m)$ containing the number of reads sequenced (i.e., originating) from each genome G^i, $i \in \{1, \ldots, m\}$,
- find a subset $M \subseteq E$ such that $d_M(x) = 1$ holds for every $x \in A$ (i.e., each read is covered by exactly one edge of M)

which together minimize:

$$(1-\alpha) \sum_{\{x,y\} \in M} c(x,y) + \alpha \cdot \beta \cdot \sum_{i=1}^{m} \sum_{j=1}^{s_i} \left| \frac{r_i}{s_i} - d_M(y_j^i) \right| + \gamma d_M(z).$$

In the full version of the paper we show that the coverage-sensitive metagenomic mapping problem is NP-hard for all $\alpha \in (0, 1)$. Thus, we opt for the common *iterative refinement strategy*, akin to the strategy behind k-means clustering [13], or Viterbi training strategies with Hidden Markov Models [3]. In Sect. 2.2 we detail this approach; the main ideas are:

1. If the unknown vector $R = (r_1, \ldots, r_m)$ is fixed to some value $R = (a_1, \ldots, a_m)$, then the optimal mapping M can be found in polynomial time with min-cost flows. We show this in the full version of the paper.
2. For finding the optimal R, we start with a vector $R^0 = (a_1, \ldots, a_m)$, where a_i equals the number of reads with BLAST hits to G^i. We repeat the following process, until the vector R converges to a stable value. For each iteration j:
   (a) find the optimal mapping M^j by min-cost flows, with input R^j;
   (b) update R^j to R^{j+1}, a vector whose i-th component equals $mean_i \cdot s_i$; here $mean_i$ is the 20 %-trimmed mean read coverage of the chunks of genome G^i, obtained from M^j.

2.2 Overview of the Implementation

Our practical implementation is divided into five stages. These depend on some parameters, whose complete list is in the full version of the paper.

Stage 1: Removing outliers species. A genome $G^i \in \mathcal{G}$ is considered an outlier if at least one of the following conditions holds.

- The average read coverage of G^i (i.e., the number of reads with BLAST hits to G^i multiplied by the average read length and divided by the length of G^i) is lower than a given parameter.

- The average read mapping per chunk (i.e., the number of reads with BLAST hits to G^i divided by s_i) is lower than a given parameter.
- The percentage of chunks without any BLAST hit is more than a given parameter.

In this stage, we remove each outlier genome G^i and the reads that have BLAST hits only to G^i.

Stage 2: Breaking ties inside each genome. A read can have BLAST hits to different chunks of the same genome. In this stage, for each read remaining after Stage 1, we select only one BLAST hit in each genome, as follows. For each remaining genome $G^i \in \mathcal{G}$, we create a sub-problem instance $G = (A \cup B, E)$ where A consists only of the reads that have BLAST hits to G^i, and B consists only of the chunks of G^i (excluding the unknown vertex z, which will be dealt with in Stage 4). We fix the one-element vector R as $R = (r_1) = (|A|)$, and solve this sub-problem using min-cost flows. After this stage, every read has at most one hit to each genome, but it can still have hits to multiple genomes.

Stage 3: Breaking ties across all genomes. A read can be mapped to different species, due to the similarity between their genomes. In order to select only one read mapping across all genomes, we solve the coverage-sensitive metagenomic mapping problem on a graph $G = (A \cup B, E)$, as follows. The set A consists of all remaining reads, and the set B of all chunks of the remaining genomes. The set of edges E is the one obtained by the filtration done in Stage 2. Since this problem is NP-hard, we employ the iterative refinement strategy, coupled with min-cost flows, mentioned at the end of Sect. 2.1. After each iteration j, we use the resulting mapping M^j to remove outlier genomes, as in Stage 1. After this stage, each read is mapped to exactly one genome, and to only one of its chunks.

Stage 4: Identifying reads from unknown genomes. In this stage we identify reads originating from species whose reference genomes are not present in the reference database. We run the same min-cost flow reduction as in Stage 2, to which we add the unknown vertex z. If a read is mapped to z, then it will be marked as coming from an unknown genome and removed from the graph. Finally, we again remove outlier genomes, as in Stage 1.

Stage 5: Estimating richness and relative abundances. For every genome G^i, we compute its average read coverage $read\_cov(i)$, and its relative abundance $rel\_abun(i)$ as: $read\_cov(i) = \overline{r_i} \cdot R/length(i)$, $rel\_abun(i) = read\_cov(i)/\sum_{j=1}^{m} read\_cov(j)$, where $\overline{r_i}$ is the number of reads mapping to G^i after Stages 1–4, R is the average read length, and $length(i)$ is the length of G^i.

3 Experimental Setup

Our experimental setup measures the effect of the following three factors: (1) genotypic homogeneity between the species, (2) complexity of the sample (i.e. the

number of species), and (3) the presence of species unknown to the methods. We simulated two types of data sets: a *low complexity* type (LC), consisting of 4 M reads sampled from 15 different species, and a *high-complexity* type (HC), consisting of 40 M reads sampled from 100 different species. The goal of the simulated LC data sets is to evaluate the methods under different levels of genotypic homogeneity between the species. For example, species with high genotypic homogeneity are difficult to precisely identify and their presence usually leads to incorrect predictions. We explain these levels of similarity in the full version of the paper. The simulated HC data sets test how a large number of randomly chosen species influences the accuracy of the methods. As opposed to the LC data sets, they do not test the ability of the tools to deal with similar species. This experimental setup is in line with previous studies [6,12].

In both the LC and HC data sets, we had two experimental scenarios: one in which all species in the sample are known to the methods (LC-Known, HC-Known), and one in which a fraction of them are unknown (LC-Unknown, HC-Unknown). LC-Known is the "perfect-information" scenario, which though not realistic, shows the performance of the tools in the best possible conditions. HC-Unknown is the most realistic scenario. In total, these simulated experiments contain 48 data sets.

The abundances of the species in each data set were chosen log-normal distributed (with mean = 1, standard deviation = 1), also in line with previous experiments [12]. This presents a challenge in finding less abundant species, since the ratio between the most abundant species and the least abundant is 100 in most cases, and the top 10 % most abundant species represents about 35 % of the sample. We selected in total 817 bacterial species from the NCBI microbial genome database, and used Metasim [10] to create the data sets using the default 454-pyrosequencing error model (with mean read length = 250 and standard deviation = 1).

In order to evaluate the accuracy of the richness estimations, we evaluated the *sensitivity* (also called *recall*) and the *precision*. Sensitivity is defined as TP/NS, and precision is defined as $TP/(TP + FP)$, where TP is the number of species correctly identified by the tool, FP is the number of species not present in the sample but reported by the tool, and NS is the true number of species in the sample. In order to obtain a single measure of accuracy, we also computed the harmonic mean of precision and sensitivity, known as *F-measure*, and defined as $2{\cdot}precision{\cdot}sensitivity/(precision{+}sensitivity)$. To evaluate the accuracy of the relative abundance predictions, we measured the ℓ_1-norm of these abundances for each data set, expressed in percentage points. For the data sets with unknown genomes, we excluded the unknown genomes from the abundance vectors and we re-normalized the resulting relative abundances to the known genomes. Note that some methods, such as MetaPhlAn [12] and mOTU [14], give some abundance estimations for the unknown genomes at a higher taxonomic levels (e.g. genus). Our method is focused instead on the core problem of analyzing the known species, without estimating the relative abundances of the unknown genomes.

Table 1. Running times of the methods with the increase of data set size. MetaFlow starts the analysis from the BLAST alignments (column "BLAST"). The data sets of size 4 M and 40 M are the synthetic ones; the reported running times are averages over all data sets of the same size. The data set of size 280 M is the real one.

Data set size	BLAST	MetaFlow	MetaPhlAn	mOTU	GSMer
4 M	243 min	28 min	14 min	9 min	42 min
40 M	1572 min	459 min	132 min	84 min	364 min
280 M	3.5 days	2025 min	387 min	380 min	N/A

We compared the performance of MetaFlow against BLAST [1], MetaPhlAn [12], mOTU [14] and GSMer [15]. In the BLAST analysis, we always selected the best alignment; in case of multiple equally-good alignments, we randomly selected one of them; if the read coverage of a species is below 0.3×, then we considered it an outlier. GSMer does not provide relative abundances, so we compared only the accuracy of the richness estimations. For mOTU, some species among our known genomes were not covered in its database, so in our evaluation it received full marks on these species. On the other hand, some of the species chosen as unknown in our experiments already existed in mOTU's database. For these species, we removed mOTU's correct prediction.

We also ran our tool using a real data set. We merged 6 G_DNA_Stool samples of a female from Human Microbiome Project (5 samples were generated using Illumina, and one sample using LS454). Their accession numbers are in the full version of the paper. The read length of all reads was normalized to 100 bp. The total number of reads from all samples was 287,565,377, out of which 98,223,162 BLAST mapped to one or more species. Only alignments with identity $\geq$97 % were selected as an input for MetaFlow. In addition to the full reference genomes in NCBI's microbial database, we also used two other references: a supercontig of *B.uniformis* (accession number NZ_JH724260.1), because *B.uniformis* was previously reported as most abundant in fecal samples [8]; and the longest scaffold of *B.plebeius* (accession number NZ_DS990130.1) because MetaPhlAn and mOTU report *B.plebeius* as most abundant in this sample.

See Table 1 for the running times of the methods tested in this paper.

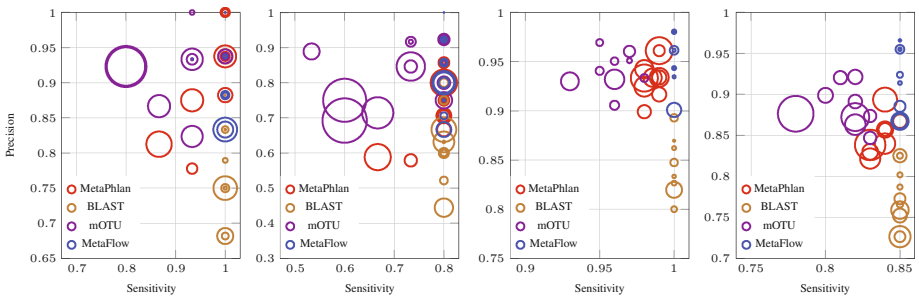
Table 2. Average over the F-measure and ℓ_1-norm in each experimental scenario. The 15 LC data sets contain 4 M reads from 15 species, and the 9 HC data sets contain 40 M reads from 100 species.

	LC-Known		LC-Unknown		HC-Known		HC-Unknown	
	F-measure	ℓ_1-norm	F-measure	ℓ_1-norm	F-measure	ℓ_1-norm	F-measure	ℓ_1-norm
MetaFlow	**0.971**	**10.41**	**0.825**	**17.87**	**0.976**	**4.86**	**0.883**	**8.01**
MetaPhlAn	0.946	26.42	0.770	31.48	0.958	18.61	0.844	19.13
BLAST	0.909	12.46	0.745	21.47	0.920	5.94	0.809	11.25
GSMer	0.218	N/A	0.163	N/A	0.327	N/A	0.259	N/A
mOTU	0.924	36.31	0.780	43.74	0.949	10.55	0.847	18.73

4 Discussion

Synthetic Data Sets. We summarize the experimental results on simulated data in Table 2 and in Fig. 2. The complete results on each LC data set are in the full version of the paper. Since BLAST reports alignments in all the reference genomes, its sensitivity is always the maximum achievable. However, this comes at the cost of low precision, since there is no proper strategy for breaking ties among alignments, or for properly removing outlier genomes. MetaPhlAn and mOTU have better precision and F-measure than BLAST, confirming that marker genes are a good way of distinguishing between similar species. However, the sensitivity of the marker-based methods suggests that such an approach is not always accurate in identifying all species present in the sample, especially at a high level of species similarity.

These results also confirm our hypothesis that taking into account the coverage across the entire genome improves the abundance estimation. For example, even though BLAST has a lower F-measure, it has better ℓ_1-norm than marker-based methods. This is due to the fact that marker regions are much shorter compared to the genome length, and thus slight variations in coverage in these regions can easily skew the abundance estimation. MetaFlow achieves the maximum sensitivity of BLAST, but it manages to highly improve the precision, and it also obtains the best F-measure among all the tested methods. Comparing the LC and HC scenarios, we can also observe that MetaFlow's richness estimations are robust with the increase in sample size. Moreover, since MetaFlow is



(a) LC-Known (b) LC-Unknown (c) HC-Known (d) HC-Unknown

Fig. 2. Results of the tools on all simulated data sets. The x-axis is the sensitivity and the y-axis is the precision; each circle is one experiment; inside each plot, the size of the circles is proportional with the ℓ_1-norm (smaller is better). In LC-Known data set: 15 data sets, each with 4 M reads from 15 species, all known. In LC-Unknown: 15 data sets, each with 4 M reads from 15 species, out of which 3 are unknown. In HC-Known: 9 data sets, each with 40 M reads from 100 species, all known. In HC-Unknown: 9 data sets, each with 40 M reads from 100 species, out of which 15 are unknown. The unknown species are among 31 bacterial species from the NCBI microbial genome database, published after 2014. GSMer's results are not included in the figures, since its precision and sensitivity were always much lower than the other methods (see Table 2).

also based on whole-genome read alignments, it gives a much better abundance estimation than marker-based methods, with an improvement of 2–4× in average ℓ_1-norm. Our problem formulation also filters out outlier species and false alignments, thus improving the abundance estimations over BLAST.

Finally, the results are always better on the HC data sets than on the LC data sets for all tools, because a small variation is more severe in a sample with 15 species than in a sample with 100 species. Recall also that the LC data sets were constructed to have similar species, whereas in the HC data sets the species were randomly chosen. The data sets with high genotypic homogeneity show that such scenario remains a difficult one: even though MetaFlow improves both the richness and abundance estimation of the competing methods, its precision drops to an average of 0.85 and its ℓ_1-norm increases to an average of 40.

Real Data Set. On the fecal metagenomic sample, the most abundant species reported by MetaFlow is *B.uniformis* (23.6 % relative abundance), which was also reported as the most abundant species in human feces [8]. This high abundance is also supported by the fact that 15,418,699 reads are mapped by BLAST only to *B.uniformis*. In the end, 10,721,492 are assigned by MetaFlow to *B.uniformis*, because of the uneven read coverage. This corresponds to an average read coverage of 220. Note also that the 10th most abundant species according to MetaFlow, *A.shahii*, has relative abundance 2.3 % and average read coverage 21. MetaPhlAn and mOTU assign *B.uniformis* abundances 1.7 % and 6.4 %, respectively.

The second most abundant species reported by MetaFlow is *B.vulgatus*, another common species in human feces [8]. MetaFlow's predicted abundance is 22.3 % (average read coverage 208), which is in line with MetaPhlAn's prediction of 17.7 % and, to an extent, mOTU's prediction of 11.9 %. In the full version of the paper we give the list of the top 10 prediction of MetaFlow, and their abundances reported by MetaPhlAn and mOTU. Four out of the top six species have also been reported by [8] as predominant in human feces, and they constitute 59 % of the sample according to MetaFlow (relative to the species known to MetaFlow).

The top abundant species in MetaPhlAn's and mOTU's predictions is *B.plebeius*, with 25 % and 16 % relative abundance, respectively. MetaFlow's reported abundance is 5.2 %. Note also that 62 species reported by MetaPhlAn are not available in the database of reference genomes given to MetaFlow (NCBI's database plus *B.uniformis* and *B.plebeius*). Since our predicted abundances are relative to the known genomes only (average read coverages are also outputted by MetaFlow), the abundance of *B.uniformis* relative to all species in the sample may be lower than 23.6 %, but it cannot be significantly lower than the one of *B.vulgatus*, as MetaPhlAn and mOTU predict.

Acknowledgement. We thank Romeo Rizzi for discussions about the computational complexity of our problem. This work was partially supported by the Academy of Finland under grants 284598 (CoECGR) to A.S. and V.M. and 274977 to A.T.

References

1. Altschul, S.F., et al.: Basic local alignment search tool. J. Mol. Biol. **215**(3), 403–410 (1990)
2. Brady, A., Salzberg, S.L.: Phymm and PhymmBL: metagenomic phylogenetic classification with interpolated Markov models. Nat. Methods **6**(9), 673–676 (2009)
3. Durbin, R., et al.: Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press, Cambridge (1998)
4. Huson, D.H., et al.: MEGAN analysis of metagenomic data. Genome Res. **17**(3), 377–386 (2007)
5. Lo, C., et al.: Evaluating genome architecture of a complex region via generalized bipartite matching. BMC Bioinform. **14**(S–5), S13 (2013)
6. Mavromatis, K., et al.: Use of simulated data sets to evaluate the fidelity of metagenomic processing methods. Nat. Methods **4**(6), 495–500 (2007)
7. Poretsky, R., et al.: Strengths and limitations of 16S rRNA gene amplicon sequencing in revealing temporal microbial community dynamics. PLoS One **9**(4), e93827 (2014)
8. Qin, J., et al.: A human gut microbial gene catalogue established by metagenomic sequencing. Nature **464**(7285), 59–65 (2010)
9. Raymond, J., et al.: The natural history of nitrogen fixation. Mol. Biol. Evol. **21**(3), 541–554 (2004)
10. Richter, D.C., et al.: MetaSim-A sequencing simulator for genomics and metagenomics. PLoS One **3**(10), e3373 (2008)
11. Rocap, G., et al.: Genome divergence in two Prochlorococcus ecotypes reflects oceanic niche differentiation. Nature **424**(6952), 1042–1047 (2003)
12. Segata, N., et al.: Metagenomic microbial community profiling using unique clade-specific marker genes. Nat. Methods **9**(8), 811–814 (2012)
13. Steinhaus, H.: Sur la division des corps matériels en parties. Bull. Acad. Polon. Sci. Cl. III. **4**, 801–804 (1956)
14. Sunagawa, S., et al.: Metagenomic species profiling using universal phylogenetic marker genes. Nat. Methods **10**(12), 1196–1199 (2013)
15. Tu, Q., et al.: Strain/species identification in metagenomes using genome-specific markers. Nucleic Acids Res. **42**, e67 (2014)

LUTE (Local Unpruned Tuple Expansion): Accurate Continuously Flexible Protein Design with General Energy Functions and Rigid-rotamer-like Efficiency

Mark A. Hallen[1], Jonathan D. Jou[1], and Bruce R. Donald[1,2,3(✉)]

[1] Department of Computer Science, Duke University, Durham, NC 27708, USA
[2] Department of Chemistry, Duke University, Durham, NC 27708, USA
[3] Department of Biochemistry, Duke University Medical Center, Durham, NC 27710, USA
brd+recomb16@cs.duke.edu

Abstract. Most protein design algorithms search over discrete conformations and an energy function that is residue-pairwise, i.e., a sum of terms that depend on the sequence and conformation of at most two residues. Although modeling of *continuous flexibility* and of *non-residue-pairwise energies* significantly increases the accuracy of protein design, previous methods to model these phenomena add a significant asymptotic cost to design calculations. We now remove this cost by modeling continuous flexibility and non-residue-pairwise energies in a form suitable for direct input to highly efficient, discrete combinatorial optimization algorithms like DEE/A* or Branch-Width Minimization. Our novel algorithm performs a local unpruned tuple expansion (LUTE), which can efficiently represent both continuous flexibility and general, possibly non-pairwise energy functions to an arbitrary level of accuracy using a discrete energy matrix. We show using 47 design calculation test cases that LUTE provides a dramatic speedup in both single-state and multi-state continuously flexible designs.

1 Introduction

Protein design algorithms compute protein sequences that will perform a desired function [5]. They generally do this by minimizing the energy of a desired binding or structural state (or some combination thereof [19,30]) with respect to sequence [4,5,7,9,13,15,25,27]. Given a model of the conformational space of a protein and its energy function (which maps conformations to their energies), this is a well-defined computational problem [5].

Previously, this minimization problem has been most efficient to solve if two restrictions are imposed on the model. First, the conformational space of the protein is modeled as discrete. Specifically, each residue takes on conformations

Electronic supplementary material The online version of this chapter (doi:10.1007/978-3-319-31957-5_9) contains supplementary material, which is available to authorized users.

M. Singh (Ed.): RECOMB 2016, LNBI 9649, pp. 122–136, 2016.
DOI: 10.1007/978-3-319-31957-5_9

from a discrete set (typically, experimentally observed sidechain conformations known as *rotamers* [23]). Hence, we optimize with respect to the amino-acid type and rotamer of each residue. Second, the energy function is assumed to be residue-pairwise, i.e., it is assumed to be a sum of terms that each depend on the amino-acid types and conformations of at most two residues.

A large body of efficient algorithms has been developed for this restricted case of the protein design problem, many of which offer provable accuracy. In particular, the dead-end elimination (DEE) algorithm [4] removes rotamers that provably cannot be part of the global minimum-energy conformation (GMEC). The A* algorithm from artificial intelligence [22] finds the optimal conformation using these unpruned rotamers [29]. This DEE/A* framework has been generalized to model free-energies for each sequence instead of simply GMECs [13,32] (the K^* algorithm). It has also been generalized to optimize combinations of stability and specificity by minimizing, with respect to sequence, a linear combination of the conformationally optimized energies of several bound and unbound states of a protein, instead of just the energy of a single state [19] (the COMETS algorithm). Several methods in addition to DEE/A* have also been used to address the protein design problem. Some of these, such as Metropolis Monte Carlo and simulated annealing [27,31], lack provable guarantees of accuracy, and thus may miss the optimal conformation significantly [41]. Other algorithms with provable accuracy are also available, largely building on techniques from integer linear programming [26,37] and weighted constraint satisfaction [37,45,46]. Notably, treewidth- and branch-width-based algorithms, such as TreePack [48] and BWM* [24], solve this problem with provable accuracy in polynomial time for systems whose residue interaction graph has treewidth or branch-width bounded by a constant [24].

However, proteins are actually continuously flexible, and continuous flexibility both in the sidechains [9] and backbone [21] has been shown to result in significantly lower energies and biologically better sequences [9,21]. Although a residue sidechain will usually be found in the *vicinity* of the modal conformation for a rotamer, its dihedral angles will often differ from this mode by 10° or more [23]. These continuous adjustments are often critical for determining what conformations are sterically feasible [9]. Thus, incorporation of continuous flexibility modeling substantially increases the accuracy of designs. The minDEE and iMinDEE methods [9,13] do this for continuous sidechain flexibility, and DEEPer does this [21] for simultaneous continuous sidechain and backbone flexibility. These methods replace the traditional discrete rotamers used in DEE/A* with voxels in the conformation space of each residue, called *residue conformations (RCs)*. An RC is defined as an amino acid type together with bounds on each of the conformational degrees of freedom of the residue (e.g., sidechain dihedrals) [21]. The modal conformation for a rotamer is usually found at the center of this voxel. In this model, the conformation space of an entire protein is a union of voxels, each of which is constructed as the cross-product of single-residue voxels. Thus, each voxel in the conformation space of the entire protein is represented by a list of RCs, one for each residue being modeled as flexible. RCs are constructed to be small enough that we can use local minimization to

find the optimal energy within the voxel. This applies to both the single-residue and entire-protein voxels.

However, the global minimum energy in this model could not previously be computed directly by DEE/A*. Instead, DEE/A* was used to enumerate *RC lists* (protein conformational voxels) in order of a *lower bound* on minimized energy [9,13,21]. Subsequently, the optimal energy for each RC list with a sufficiently low-energy lower bound was computed by minimization. The lower bound was computed from minimized pairwise interaction energies [13]. This minimization was accelerated significantly by precomputing polynomials to approximate the energy landscape, using the EPIC algorithm [20]. However, minimization was still the bottleneck in continuously flexible designs and prevented them from approaching the efficiency of designs with discrete flexibility. **In essence, these previous methods modeled continuously flexibility by modifying DEE/A* and making it do much more work. In contrast, LUTE achieves much greater efficiency by representing continuous flexibility in a form suitable for direct input into DEE/A***.

We must also address the question of the energy function. The energy landscape of a real protein is not residue-pairwise, or otherwise exactly described solely as the sum of local terms. There is, however, ample evidence that protein interactions are local in a more general sense [6,20,47,49]—i.e., that the cross-derivative of the energy with respect to conformational degrees of freedom of two residues will tend to zero fairly quickly as the distance between the residues increases. These properties are also observed for more realistic energy functions that return an energy for the entire protein, rather than breaking the energy into terms as molecular mechanics does. For example, the Poisson-Boltzmann model for implicit solvation [42] and quantum-chemical models return an energy for the entire system on which they are run. Thus, a viable approach to modeling protein energies more realistically is to infer local terms from full-protein energies. Vizcarra et al. [47] apply this approach to the Poisson-Boltzmann model, calculating pairwise energies from differences in full-protein conformational states and achieving a pairwise energy matrix that quite accurately matches the Poisson-Boltzmann energies of full conformations. However, their method can only accommodate rotamer pairs, does not support continuous flexibility, and can only be used when substituting a single rotamer into a conformation is possible while maintaining the conformation of the other residues. This is impossible when residues share conformational degrees of freedom, which is typically needed for backbone flexibility [12,21], and may also cause problems in the case of steric clashes. Also, DEE/A* has been generalized to accommodate higher-than-pairwise energy terms if these terms are computed explicitly for particular tuples, e.g., triples of residues [33]. However, most energy functions modeling higher-than-pairwise effects, including Poisson-Boltzmann, return a single energy for the entire system, rather than a sum of explicit local terms as required by algorithms such as those in Ref. [33].

Hence, today's protein and drug designers are faced with a choice. They can neglect continuous flexibility and energy terms that aren't explicitly local (e.g., explicitly pairwise), thus incurring significant error. Or they can pay a massive

overhead to incorporate them—by enumerating many conformations (for continuous flexibility) or searching exhaustively (for non-pairwise energy functions). We now offer a way around this dilemma. We construct an energy function that is an explicit, discrete sum of local energy terms, which are associated with tuples of RCs. This function maps RC lists, which represent voxels in the conformation space of a protein, to energies. But it will approximate, to arbitrary accuracy, the minimized voxel energy, which can be computed with *any* energy function: no need for residue-pairwiseness or any other local representation. Computing this approximation is a machine learning problem, and we attack it with a least-squares method. Our approach has some resemblance to cluster expansion methods, which have previously been used in quantum mechanics [3] and to represent optimized energies for protein sequences [17,18]. However, as discussed in Ref. [20], approximations of energy surfaces can be much more compact if unrealistically high-energy regions of conformational space are excluded from the approximation (and from the subsequent conformational search). Thus, unlike cluster expansion methods, we exclude *pruned* tuples of RCs, making our derived energy function a *local unpruned tuple expansion*, or LUTE. Because conformational and sequence search using the LUTE energy function is a discrete optimization problem of the type solved by DEE/A*, BWM*, and other very efficient algorithms, it allows designs to run quickly using these algorithms, while still approximating continuous flexibility and highly realistic energy functions to a high level of accuracy.

We have implemented LUTE in the OSPREY [10,13,14] open-source protein design package, which has yielded many designs that performed well experimentally—*in vitro* [2,8,11,16,36,40,43] and *in vivo* [8,16,36,40] as well as in non-human primates [40]. OSPREY contains a wide array of flexibility modeling options and provably accurate design algorithms [10,14], allowing LUTE to be used for many types of designs.

By presenting LUTE, this paper makes the following contributions:

1. A method to represent continuous flexibility and general energy functions to arbitrary accuracy in a local unpruned tuple expansion (LUTE) that can be used directly as input to discrete combinatorial search algorithms like DEE/A*.
2. A free implementation of LUTE in our laboratory's open-source OSPREY protein-design software package [2,8,13,14], available for download [14] upon publication as free software [14], supporting representation of both continuous sidechain and backbone flexibility and of molecular-mechanics and Poisson-Boltzmann energy functions.
3. Integration of LUTE with the DEE/A* [29], iMinDEE [9], BWM* [24], and COMETS [19] algorithms for sequence and conformational search.
4. Bounds on the time and space complexity of protein design calculations that model continuous flexibility and/or use energy functions with non-local terms. The time and space complexity are exponential merely in the branch-width w of the residue interaction graph, and thus the designs can be done in polynomial time for systems whose branch-width is bounded by a constant.
5. Experimental results for 47 computational design calculations on 36 protein structures using LUTE, which demonstrate its accuracy and efficiency

in single-state designs, multistate designs and for both n-body Poisson-Boltzmann and pairwise energy functions.

2 Methods

The basic strategy of LUTE is to create a discrete, quick-to-evaluate energy matrix that tells us everything we need to know for design purposes about the continuous energy landscape of a protein. We will now describe this energy matrix and how it works.

Our goals in protein design (both GMEC [9,29] and binding/partition function [13,32] calculations) can be posed in terms of a discrete function $E(\mathbf{r})$ that maps an ordered list $\mathbf{r}$ of RCs to an energy. The list $\mathbf{r}$ contains exactly one RC per residue and thus represents a voxel $V(\mathbf{r})$ in conformation space, where a vector $\mathbf{x}$ of sequence and conformational degrees of freedom satisfies $\mathbf{x} \in V(\mathbf{r})$ if the degree-of-freedom bounds defined by each RC in $\mathbf{r}$ are respected by every degree of freedom in $\mathbf{x}$. The conformational degrees of freedom in $\mathbf{x}$ will generally be continuous internal coordinates, e.g., sidechain dihedrals. We let $E'(\mathbf{x})$ denote the energy of the protein system, as a function of all its degrees of freedom.

For calculation of the GMEC energy E_g, we wish to minimize $E'(\mathbf{x})$ with respect to $\mathbf{x}$. Letting R be the set of all possible voxels, the domain over which we minimize is a finite union of voxels $\bigcup_{\mathbf{r}\in R} V(\mathbf{r})$:

$$E_g = \min_{\mathbf{x}\in \bigcup_{\mathbf{r}\in R} V(\mathbf{r})} E'(\mathbf{x}) = \min_{\mathbf{r}\in R} \min_{\mathbf{x}\in V(\mathbf{r})} E'(\mathbf{x}), \tag{1}$$

which can be expressed in the form $\min_{\mathbf{r}\in R} E(\mathbf{r})$ where

$$E(\mathbf{r}) = \min_{\mathbf{x}\in V(\mathbf{r})} E'(\mathbf{x}). \tag{2}$$

Similarly, partition function calculations seek to calculate the partition function

$$q = \int_{\bigcup_{\mathbf{r}\in R} V(\mathbf{r})} \exp\left(-\frac{E'(\mathbf{x})}{RT}\right) d\mathbf{x} = \sum_{\mathbf{r}\in R} \int_{V(\mathbf{r})} \exp\left(-\frac{E'(\mathbf{x})}{RT}\right) d\mathbf{x} \tag{3}$$

where R is the gas constant and T is the temperature. Letting

$$E(\mathbf{r}) = -RT \ln\left(\int_{V(\mathbf{r})} \exp\left(-\frac{E'(\mathbf{x})}{RT}\right) d\mathbf{x}\right) \tag{4}$$

we have a formulation of q in terms of the discrete free energy function $E(\mathbf{r})$:

$$q = \sum_{\mathbf{r}\in R} \exp\left(-\frac{E(\mathbf{r})}{RT}\right). \tag{5}$$

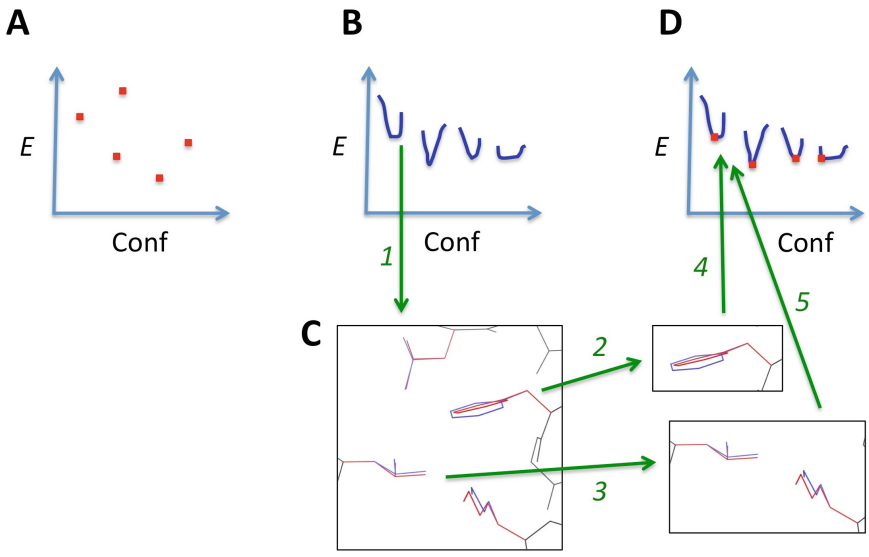


Fig. 1. LUTE makes continuously flexible design efficient by representing continuous flexibility using local, discrete energy terms. (A) Protein design with discrete flexibility searches over a discrete (albeit large) conformational space ("Conf"), looking for low-energy ("E") conformations. Highly efficient algorithms like DEE/A* are available for this problem. (B) Protein design with continuous flexibility must search over a large space of voxels (blue) in a continuous conformational space, but we are usually interested only in the minimum-energy point of each voxel. We thus want a way to search combinatorially over these minimum-energy points. (C) The minimized energy of a voxel in protein conformational space depends on all rotamers in the voxel (arrow 1). But we can expand this minimized energy as a sum of local contributions from low-order tuples (e.g., pairs) of residues (arrows 2, 3). (Minimized conformations shown in red, ideal rotamers in blue). (D) This expansion, known as LUTE, gives us a discrete combinatorial search problem of the same form as protein design with discrete flexibility (arrows 4, 5). But this new discrete problem searches over the minimum-energy points (red) of voxels in continuous conformational space (blue). We can solve this problem very efficiently. Figure shows Leu 29, Leu 51, Phe 55, and Lys 59 of the Atx1 metallochaperone (PDB id 1CC8 [39]).

Alternately, if we use the definition in Eq. (2) to define $E(\mathbf{r})$, then Eq. (5) gives us the approximation used in Refs. [13,32] for the partition function.

Because $\mathbf{r}$ is a discrete variable, the energy $E(\mathbf{r})$ can be decomposed as a sum of energies associated with tuples of RCs (Fig. 1). If all the RCs in a tuple are in the list $\mathbf{r}$, then that tuple's energy will contribute to $E(\mathbf{r})$. Most higher-order tuples of RCs consist of residues too far apart to have higher-order interactions, and thus do not contribute significantly to the energy (see Sect. 1 and Ref. [20]). We can reduce the number of tuples needed substantially further if we only try to represent favorable, non-clashing conformations. By eliminating high-energy conformations, this restriction of conformational space greatly reduces the

range of energy values over which $E(\mathbf{r})$ must be accurate. To achieve this, we prune tuples that cannot be part of favorable conformations, and consider only conformations whose tuples are all unpruned. Our expansion is much more efficient to compute after provably unfavorable tuples are pruned. Hence, we are able to represent the energy $E(\mathbf{r})$ as a local unpruned tuple expansion, or LUTE.

Let us consider a conformational space with continuous and discrete degrees of freedom, consisting of RCs, and a mapping $E(\mathbf{r})$ that we can readily calculate. For example, in a typical continuously flexible design, $E(\mathbf{r})$ is defined by Eq. (2), which we assume can be calculated by local minimization. Suppose we have a set T of tuples of RCs at different residue positions. T can contain pairs but also may contain triples, etc. We then define our local unpruned tuple expansion as a mapping $m : T \to \mathbb{R} \cup \{\perp\}$. m defines a real coefficient for each tuple $t \in T$, except for pruned tuples, for which $m(t) = \perp$. Let $T_{\mathbf{r}}$ denote the set of tuples in T that consist only of RCs in $\mathbf{r}$. For example, if T is the set of all possible RC pairs, then $T_{\mathbf{r}}$ will consists of all pairs of RCs in the list of RCs $\mathbf{r}$. Then LUTE predicts $\mathbf{r}$ to be a pruned conformation if $m(t) = \perp$ for any $t \in T_{\mathbf{r}}$, and otherwise it predicts $E(\mathbf{r}) = \sum_{t \in T_{\mathbf{r}}} m(t)$. We refer to the data structure representing the mapping m as the *LUTE energy matrix*. We call it an energy matrix because it takes a form similar to that of traditional pairwise energy matrices [4,9,21,29], although it contains significantly different numerical values when computed for the same design system.

The limiting behavior of LUTE is favorable. As we expand the set T, we must eventually approach perfect accuracy, because if T is the set of all tuples of RCs at different positions, then m can represent $E(\mathbf{r})$ for each full RC list $\mathbf{r}$ explicitly.

If we assume locality of $E(\mathbf{r})$ (see Sect. 1), we can expect inaccuracies to diminish fairly quickly with increasing size of T, because the component of $E(\mathbf{r})$ modeling the interactions of a residue i will depend only on the RCs assigned to residues fairly close in space to i. As a result, we expect a relatively compact LUTE expansion for any practical protein design problem. In practice, expansions in pairs and triples have worked well (see Sect. 3).

Most algorithms for protein design with discrete rotamers take a matrix of pairwise energies as input. By simply substituting a LUTE energy matrix for this pairwise energy matrix, we can convert any of these algorithms into an equally efficient design algorithm that searches a continuous search space instead of a discrete one, and/or that optimizes a non-pairwise energy function instead of a pairwise one. The LUTE energy matrix is computed once, before the search, which takes only polynomial time in the number of residues. For example, we need quadratic time to compute a LUTE matrix for which T is all pairs of RCs. Details of the computation by least squares of the LUTE matrix, and of the use of this matrix in search algorithms, are provided in the Supplementary Information (SI) which is available online at http://www.cs.duke.edu/donaldlab/Supplementary/recomb16/lute/

3 Results

We present here complexity results and computational experiments regarding the performance of LUTE. In Sect. 3.1, we show that the combination of LUTE with the BWM* [24] search algorithm is guaranteed to solve continuously flexible protein designs in polynomial time given a residue interaction graph with branch-width bounded by a constant. In Sects. 3.2 and 3.3, we present 30 single-state and 17 multistate protein design calculations using LUTE. We measure the gains in efficiency provided by LUTE and its ability to accurately and efficiently perform calculations that, due to their large amount of continuous flexibility (Sect. 3.2) or non-pairwise energy function (Sect. 3.3), are inaccessible to previous algorithms. These results include designs with both continuous sidechain and backbone flexibility. Sidechain dihedrals were allowed 9° of continuous motion in either direction relative to the modal value for each sidechain rotamer [23], while backbone flexibility (when present) was modeled as in Ref. [21].

3.1 Polynomial-Time Protein Design with Continuous Flexibility

Protein design in the general case is NP-hard [1,35]. In practice, however, many designs exhibit special properties that make them more tractable. For example, the residue interaction graph—the graph whose edges encode nonnegligible interactions between pairs of residues—of practical designs often has low branch-width. It has been previously shown that protein design with discrete rotamers can be performed in asymptotic time exponential only in the branch-width [24] w. Furthermore, these branch-widths can be small irrespective of the number of mutable residues [24]. Thus, for many protein designs with discrete rotamers the corresponding GMEC can be found in polynomial time. If one substitutes the LUTE matrix for the discrete pairwise energy matrix in this complexity result, then design with continuous flexibility and a constant-bounded branch-width can be solved in polynomial time as well. We can make this rigorous using the following theorem, whose proof is provided in SI Section E. In this theorem,

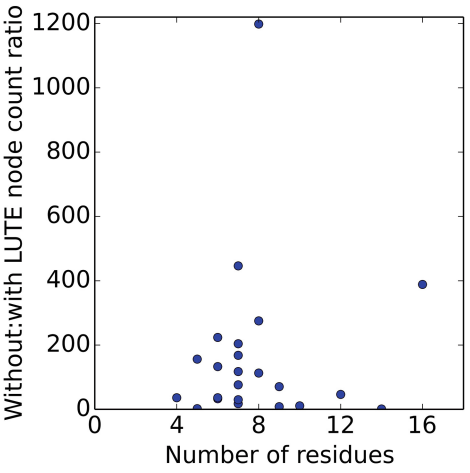


Fig. 2. LUTE markedly reduces the cost of continuously flexible conformational search. Ratios (without LUTE:with LUTE) of the number of nodes in the A* tree before enumeration of the GMEC (or of the last conformation if several conformations closely spaced in energy were calculated; see Ref. [20]), versus number of flexible residues. A 20-residue sidechain placement with node ratio 2×10^5 is not shown because it would break the scale.

a LUTE energy function is a function $E(\mathbf{r}) = \sum_{t \in T_\mathbf{r}} m(t)$, where $m : T \longrightarrow \mathbb{R} \cup \{\perp\}$ maps RC tuples to real coefficients (for this purpose, the coefficient $\perp$ of a pruned tuple is effectively ∞). Let n be the number of mutable residues, q be the maximum number of allowed RCs at any mutable residue position, and β_t and β_s be the time and space costs (respectively) to compute the branch-decomposition. This theorem establishes the complexity both of GMEC calculations and of enumeration of subsequent conformations in gap-free ascending order of energy. The latter is essential for calculation of partition functions, which can be used to account for entropy in predictions of binding [13,24,32].

Theorem 1. *For a LUTE energy function whose residue interaction graph has branch-width w, the GMEC can be computed in $\mathcal{O}(nw^2q^{\frac{3}{2}w} + \beta_t)$ time and $\mathcal{O}(nwq^{\frac{3}{2}w})$ space, and each additional conformation can be enumerated in order of LUTE energy in $\mathcal{O}(n \log q)$ time and $\mathcal{O}(n)$ space.*

3.2 Continuous Flexibility

LUTE single-state designs were run on 23 protein design systems from Ref. [20] with 4–16 mutable residues, as well as five larger systems (17–40 mutable residues), to measure the efficiency of LUTE and to observe the behavior of LUTE on the larger systems. Many of these larger systems are intractable by previous methods (except *post-hoc* minimization methods that do not account for continuous flexibility during the search). The results show that the discrete DEE/A* search with LUTE is dramatically more efficient even compared to EPIC, which offers previously state-of-the-art efficiency for continuously flexible design [20] (Fig. 2). They also demonstrate that LUTE can handle very large continuously flexible designs—including a 40-residue sidechain placement, which covers a large fraction of the residues in the Atx1 metallochaperone (Fig. 4, left), and a 20-residue design on the same structure with 5 amino-acid types allowed at every position. Furthermore, the LUTE energy matrix consistently represented the true energy landscape very closely (Fig. 3). Optimal sequences and conformations with LUTE differed significantly from the same designs run without continuous flexibility: the same top conformation was returned in only 2 of the 28 single-state designs. On average, 31 % of the RCs in the optimal conformations differed from each other. This is consistent with previous work showing that protein design calculations with and without continuous flexibility differ significantly in their results [9,21].

For many systems, LUTE achieved a fit with residual under 0.01 $(\text{kcal/mol})^2$ with only a pairwise expansion. In cases when the pairwise expansion's residual was higher, an expansion in sparse triples was performed instead. In all but one case, the triples expansion's residual was less than thermal energy at room temperature (0.59 kcal/mol, i.e., 0.35 $(\text{kcal/mol})^2$), and thus deemed insignificant.

The one outlier case was a 14-residue design on ponsin (PDB id: 2O9S). It exhibited significant local minimization errors, which caused even the matrix of pairwise lower-bound energies (computed before LUTE precomputation begins) to have errors of at least ∼10 kcal/mol. These errors indicate the failure of either our local minimizer or our assumption that local minimization suffices within RCs. As a result of these errors, the LUTE residual even with triples was 1.9 kcal/mol for this system, seven times worst than the next worst residual (the 40-residue Atx1 design). Our software now detects this problem and warns the user before the LUTE computation begins.

17 multistate protein designs were also performed, using a combination of LUTE with our COMETS [19] multistate protein design algorithm (see SI Sect. 1). The systems from these designs were taken from Ref. [19]; details are provided in SI Section G. The same designs were run with and without continuous flexibility, with LUTE used in the continuous case. As discussed in Ref. [19], COMETS provably returns the same results as exhaustive search over sequences, but it provides a speedup compared to that exhaustive search by (a) considering only a portion of the sequences in the search space explicitly, and (b) only performing a full conformational optimization for a small portion of the sequences in (a). However, previously [19] (a) was only significant in designs without continuous flexibility, and (b) was much more pronounced without continuous flexibility. LUTE brings continuously flexible COMETS designs up to speed with discrete designs on the same system (SI Fig. S3).

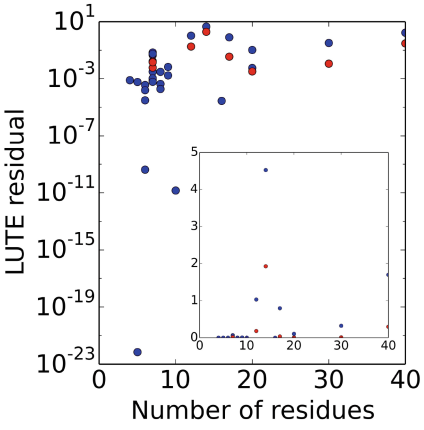


Fig. 3. LUTE accurately represents continuously minimized energies. Residuals for LUTE ((kcal/mol)^2) on the cross-validation data set, measuring the difference between the EPIC energy and a pairwise expansion (blue) or one with sparse triples (red; computed only if pairwise residual exceeded 0.01). x axis: number of flexible residues. Inset: All the same data plotted on a linear scale.

3.3 Designs that Provably Optimize Poisson-Boltzmann Energies

We also ran LUTE conformational optimization calculations on two proteins using the Poisson-Boltzmann energy function, which is non-pairwise. This energy was evaluated using Delphi [34,38] in place of the pairwise EEF1 [28] solvation energy that is used by default in OSPREY. Interestingly, triple energies did not provide significant benefit here, but LUTE was found to describe the Poisson-Boltzmann energy landscape with a high degree of accuracy. Previous

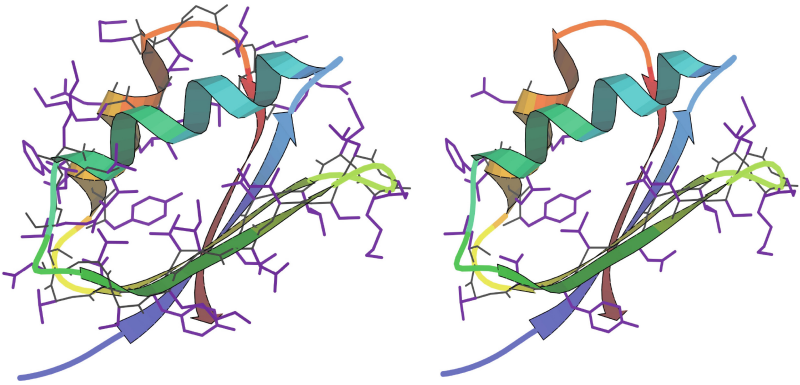


Fig. 4. LUTE enables very large provably accurate protein designs with continuous flexibility (left) and with Poisson-Boltzmann energy functions (right). Left: previously, protein designs with continuous flexibility only finished when performed with significantly fewer flexible residues, compared to designs with discrete rotamers. Even 20-residue designs were often intractable. But LUTE solved a sidechain placement problem with continuous flexibility in which 40 residues (purple) were made flexible in the Atx1 metallochaperone (PDB id 1CC8 [39]). Right: previous designs using the Poisson-Boltzmann energy function could not optimize this function directly, but only used Poisson-Boltzmann energies to rerank top hits from optimization of a simpler, pairwise energy function. But LUTE can optimize the Poisson-Boltzmann energy function directly—e.g., in a sidechain placement of 20 residues (purple) of Atx1.

work has shown that an accurate pairwise representation can be obtained for Poisson-Boltzmann energies of *discrete, rigid* rotamers [47], but our LUTE results show that a very accurate representation of *continuously minimized Poisson-Boltzmann energies* is possible as well. With continuous flexibility, a 6-residue sidechain placement on the unliganded TIR1/IAA7 complex (PDB code 2P1Q [44]) with continuous flexibility achieved a total residual of 6×10^{-4} and took about 4 days. Furthermore, a 20-residue sidechain placement without continuous flexibility on the bacterial metallochaperone protein Atx1 (PDB code 1CC8 [39]; Fig. 4, right) was solved in 2.5 h, with total residual 0.04 $(\text{kcal/mol})^2$. Unlike previous protein design calculations that use Poisson-Boltzmann energies, our new calculations provably return the minimum of the (LUTE-approximated) Poisson-Boltzmann energy over the entire conformational space, rather than simply over a set of top hits from an initial search that used a cheaper energy function.

4 Conclusions

The protein design problem enjoys a wide array of powerful algorithms for conformational and sequence search. These algorithms take a discrete energy matrix

and perform sequence optimizations, both in the single-state and multistate cases. At the same time, previous work in bioinformatics and quantum chemistry has made great progress toward quantitatively accurate modeling of the flexibility and energy landscapes of biomolecular systems. Uniting these fields to perform designs with highly realistic modeling would result in great biomedical impact, both in protein and drug design. However, because state-of-the-art flexibility and energy modeling methods do not produce a discrete matrix, there is a gap between these fields. LUTE offers a strategy to bridge this gap. By representing continuous flexibility and general energy functions in a discrete matrix, it greatly increases the realism of the modeling that discrete combinatorial optimization algorithms like DEE/A* can directly accommodate. We thus believe that LUTE can serve as a foundation for greatly improved biomolecular design protocols.

Acknowledgments. We would like to thank Drs. Kyle Roberts and Pablo Gainza for providing PDB files and scripts for testing; all members of the Donald lab for helpful comments; and the PhRMA and Dolores Zohrab Liebmann foundations (MAH) and NIH (grant R01-GM-78031 to BRD) for funding.

References

1. Chazelle, B., Kingsford, C., Singh, M.: A semidefinite programming approach to side chain positioning with new rounding strategies. INFORMS J. Comput. Comput. Biol. **16**(4), 380–392 (2004)
2. Chen, C.-Y., Georgiev, I., Anderson, A.C., Donald, B.R.: Computational structure-based redesign of enzyme activity. Proc. Nat. Acad. Sci. U.S.A. **106**(10), 3764–3769 (2009)
3. Čížek, J.: On the use of the cluster expansion and the technique of diagrams in calculations of correlation effects in atoms and molecules. In: Correlation Effects in Atoms and Molecules. Advances in Chemical Physics, vol. 14, pp. 35–90. Wiley (2009)
4. Desmet, J., de Maeyer, M., Hazes, B., Lasters, I.: The dead-end elimination theorem and its use in protein side-chain positioning. Nature **356**, 539–542 (1992)
5. Donald, B.R.: Algorithms in Structural Molecular Biology. MIT Press, Cambridge (2011)
6. Flocke, N., Bartlett, R.J.: A natural linear-scaling coupled-cluster method. J. Chem. Phys. **121**(22), 10935–10944 (2004)
7. Floudas, C.A., Klepeis, J.L., Pardalos. P.M.: Global optimization approaches in protein folding and peptide docking. In: Mathematical Support for Molecular Biology. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 47, pp. 141–172. American Mathematical Society (1999)
8. Frey, K.M., Georgiev, I., Donald, B.R., Anderson, A.C.: Predicting resistance mutations using protein design algorithms. Proc. Nat. Acad. Sci. U.S.A. **107**(31), 13707–13712 (2010)
9. Gainza, P., Roberts, K., Donald, B.R.: Protein design using continuous rotamers. PLoS Comput. Biol. **8**(1), e1002335 (2012)

10. Gainza, P., Roberts, K.E., Georgiev, I., Lilien, R.H., Keedy, D.A., Chen, C.-Y., Reza, F., Richardson, D.C., Richardson, J.S., Donald, B.R.: OSPREY: protein design with ensembles, flexibility, and provable algorithms. Methods Enzymol. **523**, 87–107 (2013)
11. Georgiev, I., Acharya, P., Schmidt, S., Li, Y., Wycuff, D., Ofek, G., Doria-Rose, N., Luongo, T., Yang, Y., Zhou, T., Donald, B.R., Mascola, J., Kwong, P.: Design of epitope-specific probes for sera analysis and antibody isolation. Retrovirology **9**(Suppl. 2), P50 (2012)
12. Georgiev, I., Donald, B.R.: Dead-end elimination with backbone flexibility. Bioinformatics **23**(13), i185–i194 (2007)
13. Georgiev, I., Lilien, R.H., Donald, B.R.: The minimized dead-end elimination criterion and its application to protein redesign in a hybrid scoring and search algorithm for computing partition functions over molecular ensembles. J. Comput. Chem. **29**(10), 1527–1542 (2008)
14. Georgiev, I., Roberts, K.E., Gainza, P., Hallen, M.A., Donald, B.R.: OSPREY (Open Source Protein Redesign for You) user manual, p. 94 (2009). www.cs.duke.edu/donaldlab/software.php
15. Georgiev, I.S., Rudicell, R.S., Saunders, K.O., Shi, W., Kirys, T., McKee, K., O'Dell, S., Chuang, G.-Y., Yang, Z.-Y., Ofek, G., Connors, M., Mascola, J.R., Nabel, G.J., Kwong, P.D.: Antibodies VRC01 and 10E8 neutralize HIV-1 with high breadth and potency even with Ig-framework regions substantially reverted to germline. J. Immunol. **192**(3), 1100–1106 (2014)
16. Gorczynski, M.J., Grembecka, J., Zhou, Y., Kong, Y., Roudaia, L., Douvas, M.G., Newman, M., Bielnicka, I., Baber, G., Corpora, T., Shi, J., Sridharan, M., Lilien, R., Donald, B.R., Speck, N.A., Brown, M.L., Bushweller, J.H.: Allosteric inhibition of the protein-protein interaction between the leukemia-associated proteins Runx1 and CBFβ. Chem. Biol. **14**, 1186–1197 (2007)
17. Grigoryan, G., Reinke, A.W., Keating, A.E.: Design of protein-interaction specificity affords selective bZIP-binding peptides. Nature **458**(7240), 859–864 (2009)
18. Grigoryan, G., Zhou, F., Lustig, S.R., Ceder, G., Morgan, D., Keating, A.E.: Ultrafast evaluation of protein energies directly from sequence. PLoS Comput. Biol. **2**(6), e63 (2006)
19. Hallen, M.A., Donald, B.R.: COMETS (Constrained Optimization of Multistate Energies by Tree Search): a provable and efficient algorithm to optimize binding affinity and specificity with respect to sequence. In: Przytycka, T.M. (ed.) RECOMB 2015. LNCS, vol. 9029, pp. 122–135. Springer, Heidelberg (2015)
20. Hallen, M.A., Gainza, P., Donald, B.R.: A compact representation of continuous energy surfaces for more efficient protein design. J. Chem. Theory Comput. **11**(5), 2292–2306 (2015)
21. Hallen, M.A., Keedy, D.A., Donald, B.R.: Dead-end elimination with perturbations (DEEPer): a provable protein design algorithm with continuous sidechain and backbone flexibility. Proteins: Struct., Funct., Bioinf. **81**(1), 18–39 (2013)
22. Hart, P.E., Nilsson, N.J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE Trans. Syst. Sci. Cybern. **4**(2), 100–107 (1968)
23. Janin, J., Wodak, S., Levitt, M., Maigret, B.: Conformation of amino acid sidechains in proteins. J. Mol. Biol. **125**(3), 357–386 (1978)
24. Jou, J.D., Jain, S., Georgiev, I.S., Donald, B.R.: BWM*: a novel, provable, ensemble-based dynamic programming algorithm for sparse approximations of computational protein design. J. Comput. Biol., 8 January 2016
25. Karanicolas, J., Kuhlman, B.: Computational design of affinity and specificity at protein-protein interfaces. Curr. Opin. Struct. Biol. **19**(4), 458–463 (2009)

26. Kingsford, C.L., Chazelle, B., Singh, M.: Solving and analyzing side-chain positioning problems using linear and integer programming. Bioinformatics **21**(7), 1028–1039 (2005)
27. Kuhlman, B., Baker, D.: Native protein sequences are close to optimal for their structures. Proc. Nat. Acad. Sci. U.S.A. **97**(19), 10383–10388 (2000)
28. Lazaridis, T., Karplus, M.: Effective energy function for proteins in solution. Proteins: Struct., Funct., Bioinf. **35**(2), 133–152 (1999)
29. Leach, A.R., Lemon, A.P.: Exploring the conformational space of protein side chains using dead-end elimination and the A* algorithm. Proteins: Struct., Funct., Bioinf. **33**(2), 227–239 (1998)
30. Leaver-Fay, A., Jacak, R., Stranges, P.B., Kuhlman, B.: A generic program for multistate protein design. PLoS One **6**(7), e20937 (2011)
31. Lee, C., Levitt, M.: Accurate prediction of the stability and activity effects of site-directed mutagenesis on a protein core. Nature **352**, 448–451 (1991)
32. Lilien, R.H., Stevens, B.W., Anderson, A.C., Donald, B.R.: A novel ensemble-based scoring and search algorithm for protein redesign and its application to modify the substrate specificity of the gramicidin synthetase A phenylalanine adenylation enzyme. J. Comput. Biol. **12**(6), 740–761 (2005)
33. LuCore, S.D., Litman, J.M., Powers, K.T., Gao, S., Lynn, A.M., Tollefson, W.T.A., Fenn, T.D., Washington, M.T., Schnieders, M.J.: Dead-end elimination with a polarizable force field repacks PCNA structures. Biophys. J. **109**(4), 816–826 (2015)
34. Nicholls, A., Honig, B.: A rapid finite difference algorithm, utilizing successive over-relaxation to solve the Poisson-Boltzmann equation. J. Comput. Chem. **12**(4), 435–445 (1991)
35. Pierce, N.A., Winfree, E.: Protein design is NP-hard. Protein Eng. **15**(10), 779–782 (2002)
36. Roberts, K.E., Cushing, P.R., Boisguerin, P., Madden, D.R., Donald, B.R.: Computational design of a PDZ domain peptide inhibitor that rescues CFTR activity. PLoS Comput. Biol. **8**(4), e1002477 (2012)
37. Roberts, K.E., Gainza, P., Hallen, M.A., Donald, B.R.: Fast gap-free enumeration of conformations and sequences for protein design. Proteins: Struct., Funct., Bioinf. **83**(10), 1859–1877 (2015)
38. Rochia, W., Sridharan, S., Nicholls, A., Alexov, E., Chiabrera, A., Honig, B.: Rapid grid-based construction of the molecular surface and the use of induced surface charge to calculate reaction field energies: applications to the molecular systems and geometric objects. J. Comput. Chem. **23**(1), 128–137 (2002)
39. Rosenzweig, A.C., Huffman, D.L., Hou, M.Y., Wernimont, A.K., Pufahl, R.A., O'Halloran, T.V.: Crystal structure of the Atx1 metallochaperone protein at 1.02 Å resolution. Structure **7**(6), 605–617 (1999)
40. Rudicell, R.S., Kwon, Y.D., Ko, S.-Y., Pegu, A., Louder, M.K., Georgiev, I.S., Wu, X., Zhu, J., Boyington, J.C., Chen, S., Shi, W., Yang, Z.-Y., Doria-Rose, N.A., McKee, K., O'Dell, S., Schmidt, S.D., Chuang, G.-Y., Druz, A., Soto, C., Yang, Y., Zhang, B., Zhou, T., Todd, J.-P., Lloyd, K.E., Eudailey, J., Roberts, K.E., Donald, B.R., Bailer, R.T., Ledgerwood, J., NISC Comparative Sequencing Program, Mullikin, J.C., Shapiro, L., Koup, R.A., Graham, B.S., Nason, M.C., Connors, M., Haynes, B.F., Rao, S.S., Roederer, M., Kwong, P.D., Mascola, J.R., Nabel, G.J.: Enhanced potency of a broadly neutralizing HIV-1 antibody *in vitro* improves protection against lentiviral infection *in vivo*. J. Virol. **88**(21), 12669–12682 (2014)

41. Simoncini, D., Allouche, D., de Givry, S., Delmas, C., Barbe, S., Schiex, T.: Guaranteed discrete energy optimization on large protein design problems. J. Chem. Theory Comput. **11**(12), 5980–5989 (2015)
42. Sitkoff, D., Sharp, K.A., Honig, B.: Accurate calculation of hydration free energies using macroscopic solvent models. J. Phys. Chem. **98**, 1978–1988 (1994)
43. Stevens, B.W., Lilien, R.H., Georgiev, I., Donald, B.R., Anderson, A.C.: Redesigning the PheA domain of gramicidin synthetase leads to a new understanding of the enzyme's mechanism and selectivity. Biochemistry **45**(51), 15495–15504 (2006)
44. Tan, X., Calderón-Villalobos, L.I.A., Sharon, M., Zheng, C., Robinson, C.V., Estelle, M., Zheng, N.: Mechanism of auxin perception by the TIR1 ubiquitin ligase. Nature **446**, 640–645 (2007)
45. Traoré, S., Allouche, D., André, I., de Givry, S., Katsirelos, G., Schiex, T., Barbe, S.: A new framework for computational protein design through cost function network optimization. Bioinformatics **29**(17), 2129–2136 (2013)
46. Traoré, S., Roberts, K.E., Allouche, D., Donald, B.R., André, I., Schiex, T., Barbe, S.: Fast search algorithms for computational protein design. J. Comput. Chem. (2016)
47. Vizcarra, C.L., Zhang, N., Marshall, S.A., Wingreen, N.S., Zeng, C., Mayo, S.L.: An improved pairwise decomposable finite-difference Poisson-Boltzmann method for computational protein design. J. Comput. Chem. **29**(7), 1153–1162 (2008)
48. Jinbo, X., Berger, B.: Fast and accurate algorithms for protein side-chain packing. J. ACM **53**(4), 533–557 (2006)
49. Zhang, D.W., Zhang, J.Z.H.: Molecular fractionation with conjugate caps for full quantum mechanical calculation of protein-molecule interaction energy. J. Chem. Phys. **119**(7), 3599–3605 (2003)

Improving Bloom Filter Performance on Sequence Data Using *k*-mer Bloom Filters

David Pellow[1], Darya Filippova[2], and Carl Kingsford[2(✉)]

[1] The Blavatnik School of Computer Science,
Tel Aviv University, 69978 Tel Aviv, Israel
[2] Computational Biology Department, School of Computer Science,
Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA, USA
carlk@cs.cmu.edu

Abstract. Using a sequence's k-mer content rather than the full sequence directly has enabled significant performance improvements in several sequencing applications, such as metagenomic species identification, estimation of transcript abundances, and alignment-free comparison of sequencing data. Since k-mer sets often reach hundreds of millions of elements, traditional data structures are impractical for k-mer set storage, and Bloom filters and their variants are used instead. Bloom filters reduce the memory footprint required to store millions of k-mers while allowing for fast set containment queries, at the cost of a low false positive rate. We show that, because k-mers are derived from sequencing reads, the information about k-mer overlap in the original sequence can be used to reduce the false positive rate up to 30× with little or no additional memory and with set containment queries that are only 1.3–1.6 times slower. Alternatively, we can leverage k-mer overlap information to store k-mer sets in about half the space while maintaining the original false positive rate. We consider several variants of such k-mer Bloom filters (kBF), derive theoretical upper bounds for their false positive rate, and discuss their range of applications and limitations. We provide a reference implementation of kBF at https://github.com/Kingsford-Group/kbf/.

Keywords: Bloom filters · Efficient data structures · k-mers

1 Introduction

Many algorithms central to biological sequence analysis rely, at their core, on k-mers — short substrings of equal length derived from the sequencing reads. For example, sequence assembly algorithms use k-mers as nodes in the de Bruijn graph [10,19]; metagenomic sample diversity can be quantified by comparing the sample's k-mer content against a database [17]; k-mer content derived from

D. Pellow—Work performed at the Language Technologies Institute, School of Computer Science, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA.

M. Singh (Ed.): RECOMB 2016, LNBI 9649, pp. 137–151, 2016.
DOI: 10.1007/978-3-319-31957-5_10

RNA-seq reads can inform gene expression estimation procedures [9]; k-mer-based algorithms can dramatically improve compression of sequence [1,11] and quality values [18].

A single sequencing dataset could generate hundreds of millions of k-mers making k-mer storage a challenging problem. Bloom filters [2] are often used to store sets of k-mers since they require much less space than hash tables or vectors to represent the same k-mer set while retaining the ability to quickly test for the presence of a specific k-mer at the cost of a low false positive rate (FPR) [4–8,10–16]. For example, Bloom filters allow for efficient k-mer counting [8], can be used to represent de Buijn graphs in considerably less space [10], and can enable novel applications like inexact sequence search over very large collections of reads [14].

The small size of Bloom filters has allowed algorithms to efficiently process large amounts of sequencing data. However, smaller Bloom filter sizes have to be traded off against higher false positive rates: a smaller Bloom filter will incorrectly report the presence of a k-mer more often. Sequencing errors and natural variation noticeably increase k-mer set sizes, with recent long read data driving k-mer set sizes even higher due to this data's lower overall quality profiles. To support large k-mer sets, researchers can either increase the Bloom filter size, choose a more costly function to compute set containment, or attempt to reduce false positive rate through other means.

To eliminate the effects of Bloom filter's false positives when representing a probabilistic de Bruijn graph [10] — where two adjacent k-mers represent an implicit graph edge — one can precompute the false edges in the graph and store them separately [4]. The results of querying the Bloom filter for a k-mer are modified such that a positive answer is returned only if the k-mer is not in the critical false positive set. The size of the set of critical false positives is estimated to be $6Nf$ where N is the number of nodes in the graph (and the number of k-mers inserted into the Bloom filter) and f is the false positive rate of the original Bloom filter [12]. Cascading Bloom filters lower the FPR by storing a series of smaller nested Bloom filters that represent subsets of critical k-mers [12].

While specific Bloom filter applications achieved improved false positive performance by using additional data structures, these applications assume the FPR of general-purpose Bloom filters derived in the paper that presented them originally [2]. This FPR is calculated based on the assumption that elements inserted into the Bloom filter are independent. In biological sequencing applications which store k-mers, the elements are not independent: if all k-mers of a sequence are stored, then there is a $k-1$ character overlap between adjacent k-mers. The information about the presence of the k-mer's neighbours can be used to infer the k-mer itself is part of the set — without the use of additional storage.

We use k-mer non-independence to develop *k-mer Bloom filters* (kBF) with provably lower false positive rates. We first consider a kBF variant where we are able to achieve more than threefold decrease in FPR with no increase in required storage and only a modest delay in set containment queries (1.2–1.3×

slower when compared to classic Bloom filter). We then consider a kBF with a stricter set containment criteria which results in more than 30-fold decrease in FPR with a modest increase in required storage and up to 1.9× delay in set containment queries.

Since the existence of k-mers in the Bloom filter can be inferred from the presence of neighbouring k-mers, we can also drop certain k-mers entirely, sparsifying the Bloom filter input set. We implement sparsifying kBFs and achieve k-mer sets that are 51–60 % the size of the original set with a slightly lower FPR at the cost of slower set containment queries.

The space and speed requirements vary between different kBF variants allowing for a multitude of applications. In memory-critical algorithms, such as sequence assembly [10] and search [14], sparse kBF can lower memory requirements allowing to process larger read collections. Applications relying on k-mers for error correction [15] or classification [17] may benefit from using kBF with guaranteed lower false positive rates to confidently identify sequencing errors and to distinguish between related organisms in the same clade.

2 Reducing False Positive Rate Using Neighbouring k-mers

When testing a Bloom filter for the presence of the query k-mer q, for example `AATCCCT` (see Fig. 1), the Bloom filter will return a positive answer — which could be a true or a false positive. However, if we query for the presence of neighbouring k-mers `xAATCCC` k-mers (where x is one of $\{A, C, G, T\}$) and receive at least one positive answer, we could be more confident that `AATCCCT` was indeed present in the Bloom filter. There is a non-zero chance that q is a false positive and its neighbour is itself a true positive, however, this is less likely than the chances of q being a false positive and thus lowers the false positive rate. We formally introduce the kBF and derive the probabilities of such false positive events below.

2.1 One-Sided k-mer Bloom Filter

We define a kBF that only checks for the presence of a single overlapping neighbour when answering a set containment query as a *one-sided kBF* (1-kBF). Each k-mer q observed in the sequence or collection of sequencing reads is inserted into a Bloom filter B independently in the standard way. To test for q's membership in a 1-kBF, the Bloom filter B is first queried for q. If the query is successful, then q is either in the true set of k-mers U, or is a false positive. If $q \in U$ and all k-mers in U were added to the Bloom filter, the set containment query for q's neighbour should return "true". We generate all eight potential left and right neighbours for q and test whether B returns true for any of them (see Algorithm 1). Under the assumption that every read or sequence is longer than k, every k-mer will have at least one neighbour in the right or left direction.

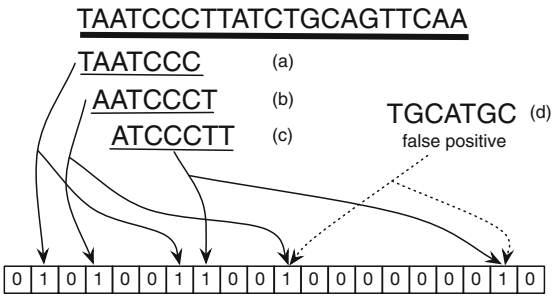


Fig. 1. The k-mers from a sequence are stored in a Bloom filter. False positives could occur when the bits corresponding to a random k-mer not in the sequence are set because of other k-mers which are in the Bloom filter. The true k-mers from the sequence all share sequence overlaps with other true k-mers from the sequence. We show how this overlap can be used to reduce the false positive rate and sparsify the set of k-mers stored in the kBF.

Algorithm 1. One-sided kBF *contains* functions

```
1: function ONE-SIDED_KBF_CONTAINS(query)
2:     if BF.CONTAINS(query) then
3:         return CONTAINS_SET(NEIGHBOUR_SET(query))
4:     return false

5: function CONTAINS_SET(set)
6:     for kmer ∈ set do
7:         if BF.CONTAINS(kmer) then return true
8:     return false
```

2.2 Theoretical FPR for a One-Sided k-mer Bloom Filter

We show that the theoretical upper bound for the FPR of a one-sided kBF is lower than that for the classic Bloom filter [2]. Suppose we inserted n unique k-mers into a Bloom filter of length m using h hash functions. Then the expected proportion of bits that are still 0 is $E = (1 - 1/m)^{hn}$ and the actual proportion of zeros, p, is concentrated around this mean with high probability [3]. The false positive rate f is:

$$f = (1-p)^h \approx (1-E)^h = \left(1 - (1 - 1/m)^{hn}\right)^h. \tag{1}$$

Let q' be q's neighbour that overlaps q by $k-1$ characters on either the left or the right side, and let t_k be a probability that a random k-mer is a true positive (i.e. is present in the set U). We assume further that events such as "q is a false positive" and "q' is a false positive" are independent since the false positives result from bits being set by uniform random hashes of other k-mers inserted into the Bloom filter. Then the chances that we get a false positive when testing

for the presence of a random k-mer q and one of its eight neighbours q' are:

$$\begin{aligned} f' &= f \cdot \Pr(\text{BF returns "True" for at least one of the adjacent } k\text{-mers}) && (2)\\ &= f \cdot (1 - \Pr(\text{BF returns "False" for every adjacent } k\text{-mer})) && (3)\\ &= f \cdot \left(1 - \Pr(\text{BF returns "False" for an adjacent } k\text{-mer})^8\right) && (4)\\ &= f \cdot \left(1 - (1 - \Pr(\text{BF returns "True" for an adjacent } k\text{-mer}))^8\right) && (5)\\ &= f \cdot \left(1 - (1 - (f + t_k))^8\right). && (6) \end{aligned}$$

Assuming that k-mers are uniformly distributed, we can estimate t_k as the chance of drawing a k-mer from the set U giving the set of all possible k-mers of length k, or $t_k = |U|/4^k$. For reasonably large values of $k \geq 20$, t_k will be much smaller than f allowing us to estimate an upper bound on f':

$$f' < f \cdot \left(1 - (1 - 2f)^8\right) \tag{7}$$

quantifying how much lower f' is than the false positive rate f for the classic Bloom filter.

2.3 Two-Sided k-mer Bloom Filter

The FPR could be lowered even more by requiring that there is a neighbouring k-mer extending the query k-mer in both directions in order to return a positive result. This is a two-sided k-mer Bloom filter, 2-kBF. However, this requires dealing with k-mers at the boundary of the input string. In Fig. 1, it can be seen that the first k-mer (`TAATCCC`) only has a right neighbour and no left neighbouring k-mers. We call this an *edge k-mer,* which must be handled specially, otherwise the 2-kBF would return a false negative.

To avoid this, 2-kBF maintains a separate list that contains these edge k-mers. We augment the Bloom filter with a hashtable EDGE_k-mer_set that stores the first and last k-mers of every sequence in order to handle edge cases. When constructing the kBF from a set of sequence reads, the first and last k-mer of each read are stored. Since reads can overlap, many of the read edge k-mers, will not be true edges of the sequence. After all the reads have been inserted into the Bloom filter, each of the stored k-mers is checked to see if it is an edge k-mer of the sequence and if it is, then it is saved in the final table of edge k-mers. The only k-mers that will be stored in the final edge table are those at the beginning and end of the sequence, or those adjacent to regions of zero coverage. Pseudocode for querying a two-sided k-mer Bloom filter is given in Algorithm 2.

2.4 Theoretical FPR for a Two-Sided k-mer Bloom Filter

Ignoring edge k-mers for simplicity and following the same assumptions and derivation as in Sect. 2.2, the FPR for two-sided kBF, f', is

$$\begin{aligned} f' = f \cdot \Pr(&\text{BF returns "True" for at least one of the left adjacent } k\text{-mers}) \cdot \\ &\Pr(\text{BF returns "True" at least one of the right adjacent } k\text{-mers}). \end{aligned} \tag{8}$$

Algorithm 2. Two-sided kBF *contains* function

```
1: function TWO-SIDED_KBF_CONTAINS(query)
2:     if BF.CONTAINS(query) then
3:         Contains_left ← CONTAINS_SET(LEFT_NEIGHBOUR_SET(query))
4:         Contains_right ← CONTAINS_SET(RIGHT_NEIGHBOUR_SET(query))
5:         if Contains_right == true and Contains_left == true then
6:             return true
7:         if Contains_right == true or Contains_left == true then
8:             if EDGE_k-mer_SET.contains(query) then
9:                 return true
10:    return FALSE
```

This leads to

$$f' = f \cdot \left(1 - (1 - (f + t_k))^4\right)^2. \tag{9}$$

An upper bound for this expression can be estimated as:

$$f' < f \cdot \left(1 - (1 - 2f)^4\right)^2. \tag{10}$$

3 Using Sequence Overlaps to Sparsify k-mer Sets

We can use the assumption that the set of k-mers to be stored, U, contains k-mers derived from an underlying string T to reduce the number of k-mers that must be stored in B without compromising the false positive rate. If we want to store a set U, we can choose a subset $K \subseteq U$ that will be stored in B. The idea is that every k-mer $u \in U$ will have some neighbours that precede it and some that follow it in the string T.

Let $L_u \subset U$ be a set of k-mers that occur before u in T, and let $R_u \subset U$ be a set of k-mers that occur after u in T. If we can guarantee that there is at least one k-mer of L_u and at least one k-mer of R_u that are close to u stored in B, then we can infer the presence of u from the presence of $v \in L_u$ and $w \in R_u$ without having to store $u \in B$. By reducing the k-mers that must be kept in B, we can maintain the set U using a smaller filter B. For example, in Fig. 1 the k-mer `AATCCCT` is preceded by `TAATCCC` and followed by `ATCCCTT`. If these two k-mers are stored in B then the presence of the middle k-mer `AATCCCT` in the sequence can be inferred without having to store it in the Bloom filter.

More formally, define P_{vu} to be the set of positions of k-mer v occurring before u in T, and let A_{uw} be the set of positions of k-mer w occurring after u in T. We then define, for $v \in L_u$ and $w \in R_u$, the set of all distances between occurrences of these k-mers that span u:

$$S_u(v, w) = \{i_w - i_v \mid i_v \in P_{vu}, i_w \in A_{uw}\}.$$

For some *skip length* s, if we can guarantee that $\min S_u(v, w) \leq s$ for some $v, w \in K$ then we can infer the presence of u without storing it in the Bloom filter by searching for neighbouring k-mers that satisfy $\min S_u(v, w) \leq s$. This leads to the following k-mer sparsification problem:

Problem 1. *Given a set of k-mers U, find a small subset $K \subset U$ such that for all $u \in U$, either $u \in K$ or there is a k-mer $v \in K \cap L_u$ and $w \in K \cap R_u$ with* $\min S_u(v, w) \leq s$.

We call this problem the *relaxed* k-mer sparsification problem. When we require exactly s skipped k-mers between those k-mers chosen for K we have the *strict* k-mer sparsification problem:

Problem 2. *Given a set of k-mers U, find a small subset $K \subset U$ such that for all $u \in U$, either $u \in K$ or there is a k-mer $v \in K \cap L_u$ and $w \in K \cap R_u$ with $s \in S_u(v, w)$.*

These sparsification problems would be easy if we could observe T — a solution would be to select every sth k-mer (Approach 3 below). However, we assume that we see only short reads from T, and must select K as best as possible. Below, we propose three solutions to the k-mer sparsification problem that are appropriate in different settings.

Approach 1: Best index match per read sparsification (kmers come from reads; arbitrary s). K will be built greedily by choosing k-mers from each read. Given a read r, we choose every sth k-mer starting from a particular index i_r, choosing i_r such that the set of k-mers K_r chosen for this read has the largest intersection with the set of k-mers K chosen so far.

Approach 2: Sparsification via approximate hitting set (k-mers come from reads; s = 1). When $s = 1$, the relaxed k-mer set sparsification problem can also be formulated as a minimal hitting set problem: For each k-mer $k \in U$, create a set L_k which includes k and every k-mer which immediately preceded it in some read, and a set R_k which includes k and every k-mer which immediately followed it in some read. Let $L = \{L_k : k \in U\}$ and $R = \{R_k : k \in U\}$. A solution to the minimal hitting problem choses a minimally sized set K such that at least one k-mer from K is in every set in R and L: $\forall N \in \{R \cup L\}\ \exists k \in K$ s.t. $k \in N$ and $|K|$ is minimized. We use a greedy approximation for the hitting set problem to choose K, the sparse set of k-mers. In each step of the greedy approximation, we add the k-mer which hits the most sets in $L \cup R$ to K.

Approach 3: Single sequence sparsification (kmers come from a known sequence T; arbitrary s). In the special case where input sequences are non-overlapping (e.g. a genome or exome) rather than multiple overlapping sequences (e.g. the results of a sequencing experiment), we solve the strict k-mer sparsification problem by taking each k-mer starting from the beginning of the sequence and then skipping s k-mers. This is a simple and fast way to choose the sparse set K, but restricted only to this special case, and will not work for sparsifying the k-mers from a set of reads generated in a sequencing experiment. It is useful, for example, if the input sequences are a reference genome which will be queried against.

Once K has been chosen, a sparse kBF can be queried for k-mers from U using the pseudocode in Algorithm 3. Two different query functions are

given: RELAXED-CONTAINS for when K satisfies the conditions of Problem 1, and STRICT-CONTAINS for when K satisfied the conditions of Problem 2. We call two k-mers with s skipped k-mers between them s-distant neighbours. The helper functions CONTAINS_NEIGHBOURS determine whether k-mers neighbouring the query k-mer at specified distances to the left and right are present and DECIDE_PRESENT determines whether the query is present depending on whether it has neighbouring k-mers or is an edge. Note that the sparse kBF also maintains a set of edge k-mers which is queried when a k-mer has neighbours in one direction but not the other.

4 Results and Discussion

We test the performance of the proposed kBFs on a variety of sequencing experiments and compare to classic Bloom filters. For each test, we store the k-mers from the input file in the kBF and create a query set by mutating k-mers from the input. We test on multiple species and types of experiments that could typically be used in applications that require Bloom filters over a range of input file sizes. The different data sets are summarized in Table 1.

Table 1. Read sets on which *k*bf variants were tested. Only reads without "N" bases were included.

Accession	Type	Read count	Read length
ERR233214_1	WGS of *P. aeruginosa*	7,571,879	92
SRR1031159_1	Metagenomic, WGS	674,989	101
SRR514250_1	Metagenomic, WGS	44,758,957	100
SRR553460	Human RNA-seq	66,396,200	49
chr15	Human chromosome (hg19)	1	81 Mbp

For all tests we used a k-mer length of $k = 20$ and 2 hash functions in the underlying Bloom filter. This choice of k is long enough that only a fraction of all possible k-mers are present in reasonably large datasets and shorter than all read lengths. The Bloom filter length is 10 times the number of k-mers inserted into it for each of the input files. For one-sided kBF and two-sided kBF, the underlying Bloom filter will be exactly the same as the classic Bloom filter they are compared to. For sparse kBF, the smaller sparse k-mer set is stored, so the underlying Bloom filter is smaller. The sparse kBFs use a skip length of $s = 1$. The implementations of the kBF variants described wrap around the basic Bloom filter implementation from libbf (http://mavam.github.io/libbf), which is used for the classic Bloom filter.

To create a query k-mer set for testing we randomly select (uniformly, with replacement) 1 million k-mers from the input file and mutate one random base. This creates a set of k-mers that are close to the real set, and will therefore have

Algorithm 3. Sparse kBF *contains* functions

```
1: function DECIDE_PRESENT(query, Contains_left, Contains_right)
2:     if Contains_right == true and Contains_left == true then
3:         return true
4:     if Contains_right == true or Contains_left == true then
5:         if EDGE_k-mer_SET.contains(query) then
6:             return true
7:     return false

8: function STRICT-CONTAINS_NEIGHBOURS(query, left_dist, right_dist)
9:     Contains_left ← CONTAINS_SET(
           S_DISTANT_LEFT_NEIGHBOUR_SET(query, left_dist))
10:    Contains_right ← CONTAINS_SET(
           S_DISTANT_RIGHT_NEIGHBOUR_SET(query, right_dist))
11:    return DECIDE_PRESENT(query, Contains_left, Contains_right)

12: function RELAXED-CONTAINS_NEIGHBOURS(query, l_dist, r_dist)
13:    Contains_left ← CONTAINS_SET(
           ⋃_{i≤l_dist} S_DISTANT_LEFT_NEIGHBOUR_SET(query, i))
14:    Contains_right ← CONTAINS_SET(
           ⋃_{i≤r_dist} S_DISTANT_RIGHT_NEIGHBOUR_SET(query, i))
15:    return DECIDE_PRESENT(query, Contains_left, Contains_right)

16: function STRICT-CONTAINS(query, s)
17:    if BF.CONTAINS(query) then
18:        if STRICT-CONTAINS_NEIGHBOURS(query, s, s) then
19:            return true
20:    for i ← 0 to s − 1 do
21:        if STRICT-CONTAINS_NEIGHBOURS(query, i, s − (i + 1)) then
22:            return true
23:    return false

24: function RELAXED-CONTAINS(query, s)
25:    if BF.CONTAINS(query) then
26:        if RELAXED-CONTAINS_NEIGHBOURS(query, s, s) then
27:            return true
28:    else
29:        for i ← 0 to s − 1 do
30:            if RELAXED-CONTAINS_NEIGHBOURS(query, i, s − (i + 1)) then
31:                return true
32:    return false
```

realistic nucleotide sequences while still providing many negative queries to test the false positive rate. For one experiment (SRR1031159) we also query with 1 million true queries (not mutated) to determine the worst-case impact on query time.

4.1 One-Sided and Two-Sided *k*BF Performance

The one-sided and two-sided *k*BF implementations achieve substantially better FPRs than the classic Bloom filter (Table 2) at the cost of some query time overhead (Fig. 2). For the one million mutated queries, only about one quarter of the queries are true positives, and one-sided *k*BF and two-sided *k*BF take 1.3 and 1.6 times as long to perform the queries, respectively. In the worst case, when all of the queries are true positives (SRR1031159 TP) one-sided *k*BF and two-sided *k*BF are 3.3 and 5.8 times slower respectively, while the speed of the classic Bloom filter does not change.

One-sided *k*BF has a FPR less than one third of the classic Bloom filter FPR at a cost of an extra one third the query time. Query times are extremely low, and this extra cost totals less than half a second to perform one million.

Table 2. False positive rates. Comparison of FPRs for classic Bloom filters, and the different *k*BF implementations. The theoretical FPRs are also shown in the last row (calculated according to Eqs. 1, 7, and 10). Hitting set sparsification uses the relaxed CONTAINS function, while best match uses the strict CONTAINS function. The sparse hitting set results for SRR514250_1 are missing since the method never completed on this data set.

Accession	Classic	1-*k*BF	2-*k*BF	Sparse	
				Best match	Hitting set
ERR233214_1	0.0329	0.0104	0.0009	0.0284	0.0311
SRR1031159_1	0.0329	0.0104	0.0009	0.0279	0.0306
SRR514250_1	0.0329	0.0106	0.0010	0.0290	—
SRR553460	0.0329	0.0104	0.0009	0.0285	0.0314
Chr15	0.0328	0.0104	0.0009	0.0284	0.0309
Theoretical FPR:	0.0328	<0.0138	<0.0019	—	—

The two-sided *k*BF requires a special data structure which stores the set of edge *k*-mers that may not be found because there is no adjacent *k*-mer on one side. The total number of *k*-mers and the number of *k*-mers in the edge set of each file are compared in Table 3. There is also extra memory and speed overhead during the two-sided *k*BF creation: as the sequence file is read in and split into *k*-mers, a set of *k*-mers at the edges of all reads (which could potentially be sequence edges that need to be stored separately) is maintained. After *k*-mers are inserted into the Bloom filter, the edges are checked, and only the true sequence edges, which do not have neighbouring *k*-mers on both sides, are stored. We do not optimize the *k*BF implementations for the one time cost of creating the *k*-mer set and populating the Bloom filter but note that the potential edge set could be pruned on the fly, keeping it smaller than the maximum size achieved here. We report the number of potential edge *k*-mers stored in the edge set, and the amount of extra time to check the edges in Table 3. In all tested cases,

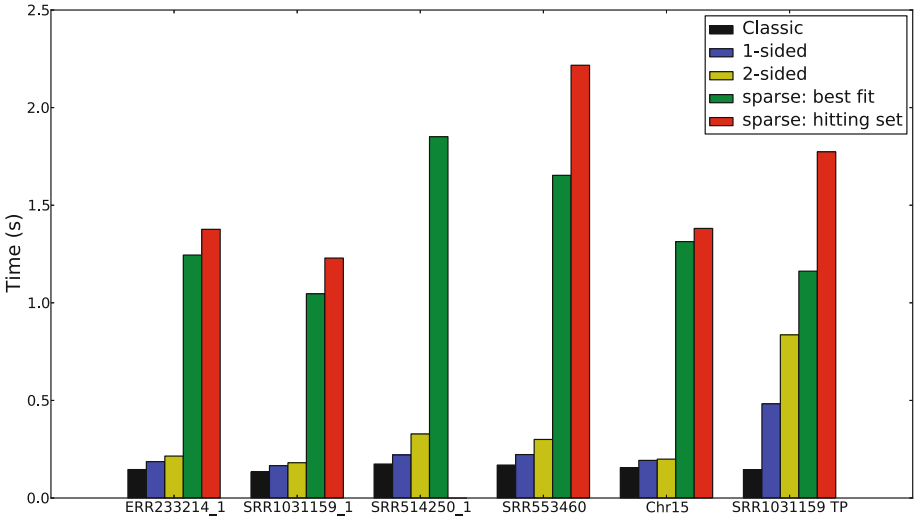


Fig. 2. Query times. Comparison of the time to query 1 million k-mers in classic BF and the different kBF implementations (average of 10 runs)

the number of edge k-mers stored is a small fraction of the total number of k-mers, reaching at most 6 % of the total. It is smallest when there are very few true sequence edges (in the single chromosome) and can be large if there are many reads with errors in the edge k-mers or many areas with zero coverage. If this overhead can be tolerated, applications could use two-sided kBF to achieve significantly lower FPRs.

Two-sided kBF provides a FPR that is 30× smaller than classic Bloom filters with a small query time penalty. Two-sided kBF also has a one-time cost of initialization to keep track of all potential edge k-mers and then determine the true edges. The extra time for this is only a small fraction of the total initialization cost. However, a large number of potential edge k-mers are stored during initialization. This number depends on the number of unique reads, and for the data sets with few long reads, i.e. the single chromosome, there is very little overhead, while when there are many reads, the first and last k-mer of each read could be stored.

4.2 Sparse kBF Performance

The sparse kBF implementations achieve slightly better FPRs than the classic Bloom filters while using a smaller filter. We report the FPRs for the best index match and hitting set implementations of sparse kBF in Table 2. We do not report specific results for single sequence sparsification (Approach 3) since we found in practice the results are the same as for best match sparsification in the cases where it is relevant. When a sparse set of k-mers is used, sparse kBF is

Table 3. Two-sided *k*bf overhead. The number of extra edge k-mers that are stored for two-sided kBF is compared to the total number of k-mers. The one-time initialization overhead includes keeping track of all potential edge k-mers and extra time to query which are the true edge k-mers. The number of potential edge k-mers is compared to the number of true edge k-mers as well as the total number of k-mers. The initialization time for two-sided kBF is shown as a fold-change over populating the classic Bloom filter with the k-mer set (average of 10 runs).

Accession	# of k-mers	# of edge k-mers	# potential edge k-mers	Init. time (fold change)
ERR233214_1	41,766,273	1,134,617	6,310,923	1.632×
SRR1031159_1	29,937,099	632,996	1,088,645	1.162×
SRR514250_1	442,498,904	6,656,205	53,063,633	1.813×
SRR553460	196,863,538	12,271,956	38,806,654	2.453×
Chr15	70,240,374	1	2	0.909×

able to use the sequence overlap to recover a similar FPR for this smaller set of k-mers.

The sparsification performance of the different implementations is compared in Table 4. The sparsification methods perform well, with the best match achieving close to the ideal size of one half the number of k-mers. The hitting set sparsification method does not perform as well, choosing a k-mer set that is roughly 10 % larger than the best match method.

Table 4. Number of *k*-mers selected by Sparse *k*bf. Comparison of the number of k-mers in the sparsified k-mer set for the different implementation methods and for the classic Bloom filter.

Accession	# k-mers Classic	# k-mers best match	# k-mers hitting set
ERR233214_1	41,766,273	21,783,670	23,635,764
SRR1031159_1	29,937,099	15,120,795	16,992,976
SRR514250_1	442,498,904	237,264,629	—
SRR553460	196,863,538	102,224,726	115,593,454
Chr15	70,240,374	36,064,290	39,152,979

Sparse kBF queries are significantly slower than classic Bloom filter queries. The speeds to perform 1 million queries for the classic BF and the different sparse kBF implementations are shown in Fig. 2. The time overhead of querying neighbouring k-mers is about ten times that for a classic Bloom filter, but is still only around 1–2 s for one million queries. In memory-constrained applications, it could be worth paying this timing penalty for smaller k-mer sets. The time overhead will grow exponentially as s is increased, but even very small s (such

Table 5. Sparse *k*bf overhead. Comparison of the one-time overhead for the initialization of the sparse kBF implementations. The one time cost of splitting the sequences into k-mers, choosing the k-mer set, checking the edge k-mers, and inserting them into the Bloom filter is reported. The hitting set implementation for SRR514250_1 used up all available memory and did not complete running. Results are the averages over 10 runs.

Accession	Initialization memory overhead (GB)			Initialization time overhead (sec)		
	2-kBF	Best match	Hitting set	2-kBF	Best match	Hitting set
ERR233214_1	3.45	4.26	42.00	121.9901	192.7540	857.8472
SRR1031159_1	1.62	2.07	28.67	21.7156	21.5318	373.4421
SRR514250_1	29.54	38.41	-	1342.1470	1856.5640	-
SRR553460	17.85	22.55	198.18	699.4796	932.5057	6012.9880
Chr15	3.94	4.49	67.04	43.5708	25.7702	844.1708

as $s = 1$ shown in our experiments) significantly reduces the size of the stored k-mer set. Similarly, as s increases, the FPR will increase, but as we showed here, for small s, the FPR is comparable to the FPR of a classic Bloom filter.

The hitting set sparsification implementation has a very large memory footprint and takes a lot of time to choose the sparse k-mer set. For the largest file (SRR514250), the implementation uses up all available RAM and does not complete even after running for several days. Table 5 compares the total time to split the input sequences into k-mers, choose the k-mer set, determine the edge k-mers and populate the Bloom filter for the different sparsification implementations and two-sided kBF. We compare with two-sided kBF because it also has edge k-mers, making it the most similar non-sparse implementation to the sparse kBF implementations. The memory overhead of initialization (measured as the maximum resident set size of the process) is also compared in Table 5.

The relaxed *contains* function, which must be used when K is selected using the hitting set formulation, needs to check more possible neighbouring k-mers, making the hitting set sparsification queries slower than the other implementations. The hitting set implementation also does not do as good a job of sparsifying the original set of k-mers. Hitting set sparsification also takes orders of magnitude more memory and time than the other methods and than the non-sparse kBF implementation.

In contrast to the hitting set sparsification, best match sparsification achieves close to one half of the original k-mer set with little extra overhead in initialization time or memory. The strict *contains* function for sparse kBF also has a better FPR than the relaxed version and takes less time to perform one million queries. In practice, there is little difference between the best match sparsification and single sequence sparsification, since they will both yield approximately the same k-mer set in a case where single sequences are being sparsified. These results mean that best match sparsification is the simplest and best way to sparsify any set of sequences, without having to determine whether it is a special case of non-overlapping sequences.

5 Conclusion

Together, the possibilities of drastically reducing the false positive rate or reducing the size of the Bloom filter have the potential to enable continued performance improvements in many applications that use Bloom filters to store k-mers from sequences. These performance improvements are necessary to allow biological sequence applications to continue to scale to larger and many more experiments.

Acknowledgments. The authors want to thank Dr. Geet Duggal and Hao Wang for the many helpful discussions. This research is funded in part by the Gordon and Betty Moore Foundation's Data-Driven Discovery Initiative through Grant GBMF4554 to Carl Kingsford, by the US National Science Foundation (CCF-1256087, CCF-1319998) and by the US National Institutes of Health (R21HG006913, R01HG007104). C.K. received support as an Alfred P. Sloan Research Fellow.

References

1. Benoit, G., Lemaitre, C., Lavenier, D., Drezen, E., Dayris, T., Uricaru, R., Rizk, G.: Reference-free compression of high throughput sequencing data with a probabilistic de Bruijn graph. BMC Bioinform. **16**(1), 288 (2015)
2. Bloom, B.H.: Space/time trade-offs in hash coding with allowable errors. Commun. ACM **13**(7), 422–426 (1970)
3. Broder, A., Mitzenmacher, M.: Network applications of Bloom filters: a survey. Internet Math. **1**(4), 485–509 (2004)
4. Chikhi, R., Rizk, G.: Space-efficient and exact de Bruijn graph representation based on a Bloom filter. Algorithms Mol. Biol. **8**(22), 1 (2013)
5. Heo, Y., Wu, X.L., Chen, D., Ma, J., Hwu, W.M.: BLESS: Bloom filter-based error correction solution for high-throughput sequencing reads. Bioinformatics **30**, 1354–1362 (2014)
6. Holley, G., Wittler, R., Stoye, J.: Bloom filter trie – a data structure for pan-genome storage. In: Pop, M., Touzet, H. (eds.) WABI 2015. LNCS, vol. 9289, pp. 217–230. Springer, Heidelberg (2015)
7. Malde, K., O'Sullivan, B.: Using Bloom filters for large scale gene sequence analysis in Haskell. In: Gill, A., Swift, T. (eds.) PADL 2009. LNCS, vol. 5418, pp. 183–194. Springer, Heidelberg (2008)
8. Marçais, G., Kingsford, C.: A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. Bioinformatics **27**(6), 764–770 (2011)
9. Patro, R., Mount, S.M., Kingsford, C.: Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms. Nat. Biotechnol. **32**(5), 462–464 (2014)
10. Pell, J., Hintze, A., Canino-Koning, R., Howe, A., Tiedje, J.M., Brown, C.T.: Scaling metagenome sequence assembly with probabilistic de Bruijn graphs. Proc. Nat. Acad. Sci. **109**(33), 13272–13277 (2012)
11. Rozov, R., Shamir, R., Halperin, E.: Fast lossless compression via cascading Bloom filters. BMC Bioinform. **15**(Suppl 9), S7 (2014)
12. Salikhov, K., Sacomoto, G., Kucherov, G.: Using cascading Bloom filters to improve the memory usage for de Brujin graphs. In: Darling, A., Stoye, J. (eds.) WABI 2013. LNCS, vol. 8126, pp. 364–376. Springer, Heidelberg (2013)

13. Shi, H., Schmidt, B., Liu, W., Müller-Wittig, W.: Accelerating error correction in high-throughput short-read DNA sequencing data with CUDA. In: IEEE International Symposium on Parallel and Distributed Processing (IPDPS 2009), pp. 1–8. IEEE (2009)
14. Solomon, B., Kingsford, C.: Large-scale search of transcriptomic read sets with sequence bloom trees. bioRxiv, p. 017087 (2015)
15. Song, L., Florea, L., Langmead, B.: Lighter: fast and memory-efficient sequencing error correction without counting. Genome Biol. **15**(11), 1–13 (2014)
16. Stranneheim, H., Käller, M., Allander, T., Andersson, B., Arvestad, L., Lundeberg, J.: Classification of DNA sequences using Bloom filters. Bioinformatics **26**(13), 1595–1600 (2010)
17. Wood, D.E., Salzberg, S.L.: Kraken: ultrafast metagenomic sequence classification using exact alignments. Genome Biol. **15**(3), R46 (2014)
18. Yu, Y.W., Yorukoglu, D., Berger, B.: Traversing the k-mer landscape of NGS read datasets for quality score sparsification. In: Sharan, R. (ed.) RECOMB 2014. LNCS, vol. 8394, pp. 385–399. Springer, Heidelberg (2014)
19. Zerbino, D.R., Birney, E.: Velvet: algorithms for de novo short read assembly using de Bruijn graphs. Genome Res. **18**(5), 821–829 (2008)

Safe and Complete Contig Assembly Via Omnitigs

Alexandru I. Tomescu[1(✉)] and Paul Medvedev[2(✉)]

[1] Helsinki Institute for Information Technology HIIT,
Department of Computer Science,
University of Helsinki, Helsinki, Finland
tomescu@cs.helsinki.fi
[2] The Pennsylvania State University, State College, USA
paul.medvedev@psu.edu

Abstract. Contig assembly is the first stage that most assemblers solve when reconstructing a genome from a set of reads. Its output consists of contigs – a set of strings that are promised to appear in any genome that could have generated the reads. From the introduction of contigs 20 years ago, assemblers have tried to obtain longer and longer contigs, but the following question was never solved: given a genome graph G (e.g. a de Bruijn, or a string graph), what are *all* the strings that can be safely reported from G as contigs? In this paper we finally answer this question, and also give a polynomial time algorithm to find them. Our experiments show that these strings, which we call *omnitigs*, are 66 % to 82 % longer on average than the popular unitigs, and 29 % of dbSNP locations have more neighbors in omnitigs than in unitigs.

1 Introduction

The genome assembly problem is to reconstruct the sequence of a genome using reads from a sequencing experiment. It is one of the oldest bioinformatics problems; nevertheless, recent projects such as the Genome 10 K have underscored the need to further improve assemblers [8]. Current algorithms face numerous practical challenges, including scalability, integration of new data types (e.g. PacBio), and representation of multiple alleles. While these are extremely important, assemblers do not produce optimal results even in very simple and idealized scenarios. Several papers have thus developed better theoretical underpinnings [9,21,22,25,36,40], often resulting in improved practical assemblers [1,31,37,41].

In most theoretical studies, the assembly problem is formulated as finding a genomic reconstruction, i.e. a single string that represents the sequence of the genome. However, the presence of repeats means that a unique genomic reconstruction usually does not exist. In practice, assemblers instead output

The full version, containing more information and proofs, is available as a pre-print at http://arxiv.org/abs/1601.02932

M. Singh (Ed.): RECOMB 2016, LNBI 9649, pp. 152–163, 2016.
DOI: 10.1007/978-3-319-31957-5_11

several strings, called *contigs*, that are "promised" to occur in the genome. We refer to this restatement of the genome assembly problem as *contig assembly*. Contigs can then be used to answer biological questions (e.g. about gene content) or perform comparative genomic analysis. When mate pairs are available, contigs can be fed to later assembly stages, such as scaffolding [2,18,33] and then gap filling [3,34].

Assemblers implement different strategies for finding contigs. The common strategy is to find *unitigs*, an idea that can be traced back to 1995 [13]. Unitigs have the desired property that they can be mathematically proven to occur in *all* possible genomic reconstructions, under clear assumptions on what "genomic reconstruction" means. We will refer to strings that satisfy such a property as being *safe* (Definition 3), and will say that a contig assembly algorithm is *safe* if it outputs only safe strings. Though most assemblers have a safe strategy at their core, they also incorporate heuristics to handle erroneous data and extend contig length (e.g. bubble popping, tip removal, and path disambiguation). Properties of such heuristics, however, are difficult to prove, and this paper will focus on core algorithms that are safe.

While the unitig algorithm is safe, it does not identify *all* possible safe strings (see Fig. 1). An improved safe algorithm was used in the EULER assembler [31], and further improvements were suggested based on iteratively simplifying the assembly graph [10,15,20,31]. However, these algorithms still do not always output all the safe strings. In fact, since the initial consideration of contig assembly 20 years ago, the fundamental question of finding *all* the safe strings of a graph is still open.

In this paper, we finally answer this question by giving a polynomial-time algorithm for outputting *all* the safe strings in the common genome graph models (de Bruijn and string graphs, Sect. 4). The key ingredient for this result is a graph-theoretic characterization of the walks that correspond to safe strings (Sect. 3). We call such walks *omnitigs* and our algorithm the *omnitig algorithm*. In our experiments on de Bruijn graphs built from data simulated according to our assumptions, maximal omnitigs are on average 66 % to 82 % longer than maximal unitigs, and 29 % of dbSNP locations have more neighbors in omnitigs than in unitigs.

Our results are naturally limited to the context of our model and its assumptions. Intuitively, we assume that (i) the sequenced genome is circular, (ii) there are no gaps in coverage, and (iii) there are no errors in the reads. A mathematically precise definition of our model is in Sect. 2, were we also argue that such a model is necessary if we want to prove even the simplest results about unitigs. Similar to previous studies, we also do not deal with multiple chromosomes or the double-strandedness of DNA and assume the genome is represented by a covering walk. As with previous papers that developed better theoretical underpinnings [9,22,25,30], it is necessary to prove results in a somewhat idealized setting. While this paper falls short of analyzing real data, we believe that omnitigs can be incorporated into practical genome analysis and assembly tools

– similar to the way that error-free studies of de Bruijn [30] and paired de Bruijn graphs [22] became the basis of practical assemblers [1,31,40].

Related Work. The number of related assembly papers is vast, and we refer the reader to some surveys [23,27]. For an empirical evaluation of the correctness of several state-of-the-art assemblers, see [35]. Here, we discuss work on the theoretical underpinnings of assembly.

There are many formulations of the genome assembly problem. One of the first asks to reconstruct the genome as a shortest superstring of the reads [13,14,29]. Later formulations referred to a graph built from the reads, such as a de Bruijn graph [9,31] or a string graph [25,37]. In an (edge-centric) de Bruijn graph, the reconstructed genome can be modeled as a circular walk covering every edge exactly once—Eulerian—[31], or at least once [11,20,21,26]. In a string graph, the reconstructed genome can be modeled as a circular walk covering every node exactly once—Hamiltonian—[19,28], or at least once [26]. These models have also been considered in their weighted versions [20,26,28], or augmented to include other information, such as mate-pairs [12,22,32]. Each such notion of genomic reconstruction brought along questions concerning its validity. For example, under which conditions on the sequencing data (e.g., coverage, read length, error rate) is there at least one reconstruction [17,24], or exactly one reconstruction [4,16,31]. If there are many possible reconstructions, then what is their number [7,15] and in which aspects one is different from all others [7]. In contrast to the framework of this paper, all these formulations deal with finding a single genomic reconstruction as opposed to a set of safe strings (i.e. contigs).

The most commonly employed safe strings are the ones spelled by maximal *unitigs*, where *unitigs* are paths whose internal nodes have in- and out-degree one. Figure 1 show an example of the output of the unitig algorithm, and also illustrates that it does not identify all safe strings. The EULER assembler [31] takes unitigs a step further and identifies strings spelled by paths whose internal nodes have out-degree equal to one (with no constraint on their in-degree). It can be shown that such strings are also safe. However, the most complete characterization of safe strings that we found is given by the *Y-to-V algorithm* [10,15,21]. Consider a node v with exactly one in-neighbor u and more than one out-neighbors $w_1, \ldots, w_d$. The *Y-to-V reduction* applied to v removes v and its incident edges from the graph and adds nodes $v_1, \ldots, v_d$ with edges from u to v_i and from v_i to w_i, for all $1 \leqslant i \leqslant d$. The Y-to-V reduction is defined symmetrically for nodes with out-degree exactly one and in-degree greater than one. The Y-to-V algorithm proceeds by repeatedly applying Y-to-V reductions, in arbitrary order, for as long as possible. The algorithm then outputs the strings spelled by the maximal unitigs in the final graph (see Fig. 1d for an example). The Y-to-V algorithm can also be shown to be safe, but, as we will show in Fig. 1, it does not always output *all* the safe strings. We are not aware of any study that compares the merits of Y-to-V contigs to unitigs, and we therefore perform this analysis in Sect. 5.

Basic Definitions. Given a string x and an index $1 \leqslant i \leqslant |x|$, we define $\mathrm{pre}(x, i)$ and $\mathrm{suf}(x, i)$ as its length i prefix and suffix, respectively. If x and y are two strings, and $\mathrm{suf}(x, k) = \mathrm{pre}(y, k)$ for some $k \leqslant |x| - 1$, then we define $x \oplus^k y$ as $x[1..|x| - k]$ concatenated with y. A k-mer of x is a substring of length k. Let R be a set of strings, which we equivalently refer to as *reads*. The *node-centric de Bruijn graph built on* R, denoted $\mathrm{DB}^k_{\mathrm{nc}}(R)$, is the graph whose set of nodes is the set of all k-mers of R, in which there is an edge from a node x to a node y iff $\mathrm{suf}(x, k-1) = \mathrm{pre}(y, k-1)$ [6]. The *edge-centric de Bruijn graph built on* R, denoted $\mathrm{DB}^k_{\mathrm{ec}}(R)$ is defined similarly to $\mathrm{DB}^k_{\mathrm{nc}}(R)$, with the difference that there is an edge from x to y iff $\mathrm{suf}(x, k-1) = \mathrm{pre}(y, k-1)$ and $x \oplus^{k-1} y$ is a substring of some string in R [9]. The *weight* of the edges of $\mathrm{DB}^k_{\mathrm{nc}}(R)$ and $\mathrm{DB}^k_{\mathrm{ec}}(R)$ is $k - 1$.

We denote by n and m the number of nodes and edges, respectively, of an arbitrary graph G. We use $N^-(v)$ to denote the set of in-neighbors of a node v. A *walk* w is a sequence $(v_0, e_0, v_1, e_1, \ldots, v_t, e_t, v_{t+1})$ where $v_0, \ldots, v_{t+1}$ are nodes, and each e_i is an edge from v_i to v_{i+1}, and $t \geqslant -1$. Its *length* is its number of edges. A *path* is a walk where the nodes are all distinct, except possibly the first and last nodes. Walks of length at least one are called *proper*. A walk whose first and last nodes coincide is called *circular* walk. A path with first node u and last node v will be called a *u-v path*. A walk is called *node-/edge-covering* if it passes at least once through each node/edge.

Let ℓ be a function labeling the nodes of G and let c be a function giving weights to the edges (intuitively, c should represent the length of overlaps). One can apply the notion of string spelled by a walk by defining the string *spelled by* w as $\mathrm{spell}(w) = \ell(v_0) \oplus^{c(e_0)} \ell(v_1) \oplus^{c(e_1)} \cdots \oplus^{c(e_t)} \ell(v_{t+1})$.

2 Problem Formulation

There are various theoretical approaches to formulating the assembly problem. Here, we adopt a model that captures the most popular ones: the node-centric de Bruijn graph, the edge-centric de Bruijn graph, and the string graph [25]. We generalize these using a notion of *genome graph*:

Definition 1 (Genome graph). *A graph G with edge-weights given by c and node-labels is a* genome graph *if and only if (1) for every edge $e = (u, v)$, $\mathrm{suf}(u, c(e)) = \mathrm{pre}(v, c(e))$, and (2) for any two walks w_1 and w_2, w_1 is a subwalk of w_2 if and only if $\mathrm{spell}(w_1)$ is a substring of $\mathrm{spell}(w_2)$.*

Both node- and edge-centric de Bruijn graphs are genome graphs, directly by their definition. Similarly, the interested reader can verify that string graphs, as commonly defined in [21,25,26,36], are genome graphs. Intuitively, the first condition states that the edge-weights represent the length of overlaps between strings, while the second condition prohibits a certain redundancy in the graph. It can be broken if, for example, there are nodes with duplicate labels, or if some labels are substrings of others. Or, for strings graphs, it can be broken if transitive edges are not removed from the graph [25]. We now augment a genome graph with a rule defining a "genomic reconstruction."

Definition 2 (Graph model). *A* graph model $\mathcal{G}$ *is defined by*

- *An algorithm transforming a set of reads R into a genome graph $\mathcal{G}(R)$.*
- *A rule determining if a walk in $\mathcal{G}(R)$ is a* genomic reconstruction.

Intuitively, a genomic reconstruction spells a genome that could have generated the observed set of reads R. In this paper, we consider two graph models. In the *edge-centric* model, a genomic reconstruction is a circular edge-covering walk; its underlying genome graph can be e.g. an edge-centric de Bruijn graph. In the *node-centric* model, a genomic reconstruction is a circular node-covering walk; its underlying genome graph can be a node-centric de Bruijn graph or a string graph. We assume, without explicitly stating it onwards, that $\mathcal{G}(R)$ contains at least one genomic reconstruction, and additionally that $\mathcal{G}(R)$ is always different from a single cycle. We now define the safe strings:

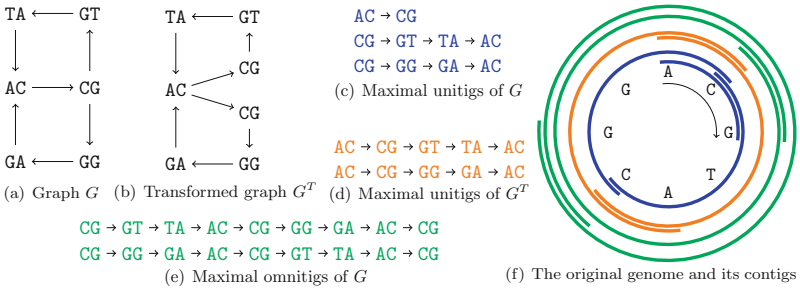
Definition 3 (Safe string). *Given a set of reads R and a graph model $\mathcal{G}$, a string s is said to be a* safe string for $\mathcal{G}(R)$ *if for every genomic reconstruction w of $\mathcal{G}(R)$, s is a substring of* spell(w).

In particular, for a node-centric (respectively, edge-centric) graph model $\mathcal{G}$, a string s is safe if for every circular node-covering (respectively, edge-covering) walk w, s is a substring of spell(w). Solving the following problem gives all the safely-retrievable information from a graph model.

Definition 4 (The safe and complete contig assembly problem). *Given a set of reads R and a graph model $\mathcal{G}$, find* all *safe strings for $\mathcal{G}(R)$.*

In this paper we solve this problem for the node- and edge-centric models defined above. Due to space limitations, we will focus on the edge-centric model here, and leave the node-centric model for the full version of the paper [38]. As a technical aside, our algorithms will output only *maximal* safe strings, in the sense that they are not a substring of any other safe string. In fact, this is desirable in practice, and moreover, the set of all safe strings is the set of all substrings of the maximal ones.

A note on assumptions: Our model makes three implicit assumptions, as outlined at the end of the Introduction. Here, we observe that such assumptions are necessary to prove even the simplest desired property: that the unitig algorithm outputs only safe strings. Let $w = (v_0, e_0, v_1, e_1, v_2)$ be a unitig in an edge-centric de Bruijn graph G built from the $(k+1)$-mers of a genome S. If the genome is not circular (assumption (i)), then e.g. the last k-mer of S can be v_0, its first k-mer can be v_1, the string $v_0 \oplus^k v_1$ can appear inside S, but $v_0 \oplus^k v_1 \oplus^k v_2$ does not have to appear in S. If there are gaps in coverage (assumption (ii)), then both an in-neighbor v' and an out-neighbor v'' of v_1 may be missing from G making w look safe whereas in reality $v_0 \oplus^k v_1 \oplus^k v_2$ may not be a substring of S. A sequencing error (assumption (iii)) creates a bubble in G with one of its paths being a unitig not spelling a substring of S.



(a) Graph G (b) Transformed graph G^T (c) Maximal unitigs of G (d) Maximal unitigs of G^T (e) Maximal omnitigs of G (f) The original genome and its contigs

Fig. 1. The output of the three algorithms on the edge-centric de Bruijn graph G from (a), built from the circular string in (f). Each contig is drawn as an arc on the wheel in (f). (c): the maximal unitigs of G; (b): the Y-to-V reduction is applied to node CG and the resulting graph G^T is shown; no more reductions are applicable and G^T has two maximal unitigs, shown in (d); (e): the maximal omnitigs of G. Note that this example illustrates that the Y-to-V algorithm does not always output all safe strings, because its output (d) does not contain the strings of (e).

3 Characterization of Safe Strings: Omnitigs

Definition 5 (Omnitig, edge-centric model). *Let G be a directed graph and let $w = (v_0, e_0, v_1, e_1, \ldots, v_t, e_t, v_{t+1})$ be a walk in G. We say that w is an* omnitig *iff for all $1 \leqslant i \leqslant j \leqslant t$, there is no proper v_j-v_i path with first edge different from e_j, and last edge different from e_{i-1}.*

The following theorem states that the omnitigs spell all the safe strings. Its proof is left for the full version of the paper [38].

Theorem 1. *Given an edge-centric graph model $G = \mathcal{G}(R)$ built for a set of reads R, a string s is safe for G iff s is spelled by an omnitig in G.*

4 Omnitig Algorithm

In this section, we use Theorem 1 to give the omnitig algorithm (Algorithm 1) and prove that it runs in polynomial time (Theorem 2). The algorithm finds all maximal omnitigs of $\mathcal{G}(R)$, which, by Theorem 1, are exactly the maximal safe strings of $\mathcal{G}(R)$. This is based on the following observation, which follows directly from the definition of omnitigs:

Observation 1. *Consider a walk $w' = (v_0, e_0, \ldots, e_{t-1}, v_t, e_t, v_{t+1})$ of length at least two, and consider its subwalk $w = (v_0, e_0, \ldots, e_{t-1}, v_t)$. Then w' is an omnitig if and only if (i) w is an omnitig and (ii) for all $0 \leqslant i \leqslant t-1$, there is no proper v_t-v_i path with first edge different from e_t and last edge different from e_{i-1}.*

Algorithm 1. Omnitig algorithm

```
1  extend(w)
2      denote w = (v_0, e_0, v_1, e_1, ..., v_{t-1}, e_{t-1}, v_t);
3      foreach edge e = (v_t, y) out-going from v_t do
4          X := (N^-(v_1) ∪ ··· ∪ N^-(v_t)) \ {v_0, ..., v_t};
5          let G' equal G minus the edge e;
6          if there is no path in G' from v_t to a node of X then
7              extend((v_0, e_0, v_1, e_1, ..., v_{t-1}, e_{t-1}, v_t, e, y));
8      if w was never extended then
9          W := W ∪ {w};
10 W := ∅;
11 foreach edge e = (u, v) of G do
12     extend((u, e, v));
13 remove from W any walk that is a subwalk of another walk in W;
14 return {spell(w) : w ∈ W};
```

The idea of the algorithm is to start a depth-first traversal of G from every edge (Lines 11–12), which by definition is an omnitig, and to keep traversing edges as long as the current walk is an omnitig. An omnitig w is thus recursively constructed, by possibly extending to the right with each edge e out-going from its last vertex (Lines 3–7). If w extended with e is not an omnitig, then we abandon this extension because Observation 1 tells us that no further extension could be an omnitig. To check if this extension is an omnitig or not, it is enough to check whether condition (ii) of Observation 1 is satisfied (Lines 4–6). Condition (i) is automatically satisfied because of the structure of the algorithm–we extend only walks that are omnitigs.

Next, we show that the algorithm runs in polynomial time. First, we show that the number of omnitigs included in W, prior to removal of non-maximal ones, is polynomial (proof left for the full version [38]):

Lemma 1. *Let W be a set of omnitigs in an edge-centric graph model $\mathcal{G}(R)$. If no omnitig in W is a prefix of another omnitig in W, then $|W| \leqslant nm$ and the length of any omnitig in W is at most nm.*

Note that Line 8 guarantees that W, prior to removal of subwalks in Line 13, satisfies the prefix condition of Lemma 1. Lemma 1 then implies that reporting one omnitig by our algorithm takes polynomial time, and there are only polynomially many omnitigs reported. Thus:

Theorem 2. *Let R be a set of reads and $\mathcal{G}(R)$ be an edge-centric graph model. Algorithm 1 outputs in polynomial time all safe strings of $\mathcal{G}(R)$.*

Prior to starting, we apply the Y-to-V algorithm and the standard graph compaction algorithm to compact unitigs [5]. This significantly reduces the number of nodes/edges in the graph without changing the maximal safe strings.

In the full version [38] we describe other implementation details that are crucial in practice. Our implementation is freely available for use at https://github.com/alexandrutomescu/complete-contigs.

5 Experimental Results

We wanted to test the potential of omnitigs as an alternative to unitigs, under the assumptions of Sect. 2. We chose one bacterial genome, *E.coli*, and one larger genome, Human chr10 (circularized). The graph model was the edge-centric de Bruijn graph built on the set of all $(k+1)$-mers of the genome. We used $k = 31$ and $k = 55$ for *E.coli* and chr10, respectively, according to what was used in practice for their assembly.

Table 1. Results for $\mathrm{DB}^k_{\mathrm{ec}}(R)$, where R is the set of all $(k+1)$-mers of the genome.

	E.coli ($k = 31$)				chr10 ($k = 55$)			
	# strings	avg len	E-size	time (s)	# strings	avg len	E-size	time (s)
unitigs	1,743	2,654	33,309	<1	259,845	546	8,344	1
Y-to-V	1,004	4,682	33,632	<1	159,101	878	8,376	2
omnitigs	983	4,832	34,557	<1	158,236	887	8,401	1,046

We wanted to measure the effect of omnitigs on assembly contiguity in terms of (1) increase in contig length, and (2) increase of biological context for elements of interest. To measure the increase in length, we measured the average contig length and the E-size. Since multiple contigs can cover overlapping regions, we found the E-size metric [35] to be more appropriate than the N50 metric. The E-size of a set of substrings of a genome is defined as the average, over all genomic positions i, of the mean length of all substrings spanning position i. This was computed by aligning the contigs to the reference. Table 1 shows that omnitigs exhibit significantly more contiguity than unitigs, with an average contig length that is 62–82 % higher. The little improvement in the E-size (1–4 %) indicates that the increase in average length comes from shorter contigs.

We wanted to also measure the potential of omnitigs to improve downstream biological analysis, relative to unitigs. Longer contigs can provide more flanking context around important genomic elements such as SNPs. One general type of study collects statistics about the relationship of each SNP to other SNPs on the same contig; such a study is necessarily limited by the number of SNPs present on the same contig [39]. We call this number the *block size* of a SNP. To see the effect of omnitigs on such a study, we identified chr10 locations of SNPs in the human population (using dbSNP), and the block size of each SNP in the omnitig vs. the unitig algorithms. Figure 2A shows that omnitigs in many cases provide more SNP context. The number of SNPs whose block size increased was ∼1.7 million (out of ∼5.9 million) and whose block size increased by more than

10 was ∼137 thousand. The average number of SNPs per unitig was 41, with only 26 per unitig. Consistent with the contiguity results of Table 1, the effect is more pronounced on contigs with less SNPs.

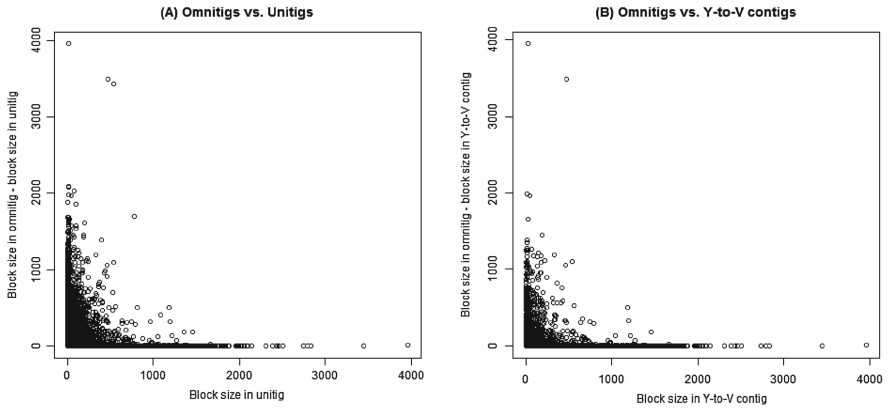


Fig. 2. The increase in SNP block size in omnitigs compared to unitigs (A) and Y-to-V contigs (B). Each point is a SNP, and the x-value is the block size of the unitig (in A) or Y-to-V contig (in B) covering it. The y-value is the increase in the block size, when compared with omnitigs. Note that the y-axis does not represent the block size, but a difference of block sizes.

We also compared omnitigs to Y-to-V contigs. Y-to-V contigs have been proposed in the literature [10,15,21], but, to the best of our knowledge, there has not been a quantitative study comparing their merits against other contig algorithms. Omnitigs also provide more SNP context than Y-to-V contigs, with ∼266 thousand SNPs having an increase in block size (Fig. 2B). Omnitigs are only marginally better than Y-to-V contigs in terms of average contiguity (Table 1). Our results suggest that, though not as beneficial as omnitigs, Y-to-V contigs may nevertheless provide a faster alternative to unitigs than the omnitig algorithm.

Table 1 also shows the wall-clock running times of our algorithms. The experiments were run on a node with two Xeon 2.53 GHz CPUs. We parallelized the omnitig algorithm so that it utilized all 8 available cores. We observe negligible running times for all algorithms on *E.coli*. On chr10, the running time of the omnitig algorithm is significantly longer (by 18 mins) than the unitig or Y-to-V algorithm, though it would still not form a bottleneck in an assembly pipeline. The memory usage did not exceed 1 GB at any point, though we believe it can be significantly reduced with a more careful implementation.

Conclusion: There are two natural directions for future work: practical and theoretical. In the practical direction, the omnitig algorithm should be extended to handle the complexities of real data such as sequencing errors, imperfect coverage, linear genomes, and double-strandedness. This is a non-trivial task which is outside the scope of the current study, but it will be important in facilitating

the application to genome analysis and assembly. In the theoretical direction, we believe that omnitigs exhibit more structure that can be exploited in a faster algorithm for finding all maximal omnitigs. We are also currently studying the graph model when a genomic reconstruction is any collection of circular covering walks (as in metagenomic sequencing of bacteria).

Acknowledgments. We would like to thank Daniel Lokshtanov for initial discussions, Rayan Chikhi for feedback on the manuscript, and Nidia Obscura Acosta for helpful discussions. This work was supported in part by NSF awards DBI-1356529, IIS-1453527, and IIS-1421908 to PM, and by Academy of Finland grant 274977 to AT.

References

1. Bankevich, A., et al.: SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. J. Comp. Biol. **19**(5), 455–477 (2012)
2. Boetzer, M., et al.: Scaffolding pre-assembled contigs using SSPACE. Bioinformatics **27**(4), 578–579 (2011)
3. Boetzer, M., Pirovano, W.: Toward almost closed genomes with gapfiller. Genome Biol. **13**(6), 1–9 (2012)
4. Bresler, G., et al.: Optimal assembly for high throughput shotgun sequencing. BMC Bioinform. **14**(Suppl 5), S18 (2013)
5. Chikhi, R., Limasset, A., Jackman, S., Simpson, J.T., Medvedev, P.: On the representation of de Bruijn graphs. In: Sharan, R. (ed.) RECOMB 2014. LNCS, vol. 8394, pp. 35–55. Springer, Heidelberg (2014)
6. Chikhi, R., Rizk, G.: Space-efficient and exact de Bruijn graph representation based on a bloom filter. In: Raphael, B., Tang, J. (eds.) WABI 2012. LNCS, vol. 7534, pp. 236–248. Springer, Heidelberg (2012)
7. Guénoche, A.: Can we recover a sequence, just knowing all its subsequences of given length? Comput. Appl. Biosci. **8**(6), 569–574 (1992)
8. Haussler, D., et al.: Genome 10 K: a proposal to obtain whole-genome sequence for 10,000 vertebrate species. J. Hered. **100**(6), 659–674 (2008)
9. Idury, R.M., Waterman, M.S.: A new algorithm for DNA sequence assembly. J. Comp. Biol. **2**(2), 291–306 (1995)
10. Jackson, B.G.: Parallel methods for short read assembly. Ph.D. thesis, Iowa State University (2009)
11. Kapun, E., Tsarev, F.: De Bruijn superwalk with multiplicities problem is NP-hard. BMC Bioinform. **14**(Suppl 5), S7 (2013)
12. Kapun, E., Tsarev, F.: On NP-hardness of the paired de Bruijn sound cycle problem. In: Darling, A., Stoye, J. (eds.) WABI 2013. LNCS, vol. 8126, pp. 59–69. Springer, Heidelberg (2013)
13. Kececioglu, J.D., Myers, E.W.: Combinatiorial algorithms for DNA sequence assembly. Algorithmica **13**(1/2), 7–51 (1995)
14. Kececioglu, J.D.: Exact and approximation algorithms for DNA sequence reconstruction. Ph.D. thesis, University of Arizona, Tucson, AZ, USA (1992)
15. Kingsford, C., et al.: Assembly complexity of prokaryotic genomes using short reads. BMC Bioinform. **11**(1), 21 (2010)
16. Lam, K., et al.: Near-optimal assembly for shotgun sequencing with noisy reads. BMC Bioinform. **15**(S–9), S4 (2014)

17. Lander, E.S., Waterman, M.S.: Genomic mapping by fingerprinting random clones: a mathematical analysis. Genomics **2**(3), 231–239 (1988)
18. Luo, R., et al.: SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler. GigaScience **1**(1), 18 (2012)
19. Lysov, I., et al.: Determination of the nucleotide sequence of DNA using hybridization with oligonucleotides. a new method. Dokl Akad Nauk SSSR **303**(6), 1508–1511 (1988)
20. Medvedev, P., Brudno, M.: Maximum likelihood genome assembly. J. Comp. Biol. **16**(8), 1101–1116 (2009)
21. Medvedev, P., Georgiou, K., Myers, G., Brudno, M.: Computability of models for sequence assembly. In: Giancarlo, R., Hannenhalli, S. (eds.) WABI 2007. LNCS (LNBI), vol. 4645, pp. 289–301. Springer, Heidelberg (2007)
22. Medvedev, P., et al.: Paired de Bruijn graphs: a novel approach for incorporating mate pair information into genome assemblers. J. Comp. Biol. **18**(11), 1625–1634 (2011)
23. Miller, J.R., et al.: Assembly algorithms for next-generation sequencing data. Genomics **95**(6), 315–327 (2010)
24. Motahari, A.S., et al.: Information theory of DNA shotgun sequencing. IEEE Trans. Inf. Theory **59**(10), 6273–6289 (2013)
25. Myers, E.W.: The fragment assembly string graph. In: ECCB/JBI, p. 85 (2005)
26. Nagarajan, N., Pop, M.: Parametric complexity of sequence assembly: theory and applications to next generation sequencing. J. Comp. Biol. **16**(7), 897–908 (2009)
27. Nagarajan, N., Pop, M.: Sequence assembly demystified. Nat. Rev. Genet. **14**(3), 157–167 (2013)
28. Narzisi, G., Mishra, B., Schatz, M.C.: On algorithmic complexity of biomolecular sequence assembly problem. In: Dediu, A.-H., Martín-Vide, C., Truthe, B. (eds.) AlCoB 2014. LNCS, vol. 8542, pp. 183–195. Springer, Heidelberg (2014)
29. Peltola, H., et al.: Algorithms for some string matching problems arising in molecular genetics. In: IFIP Congress, 59–64 (1983)
30. Pevzner, P.A.: L-Tuple DNA sequencing: computer analysis. J. Biomol. Struct. Dyn. **7**(1), 63–73 (1989)
31. Pevzner, P.A., et al.: An Eulerian path approach to DNA fragment assembly. Proc. Natl. Acad. Sci. **98**(17), 9748–9753 (2001)
32. Rubinov, A.R., Gelfand, M.S.: Reconstruction of a string from substring precedence data. J. Comp. Biol. **2**(2), 371–381 (1995)
33. Sahlin, K., et al.: BESST-efficient scaffolding of large fragmented assemblies. BMC Bioinform. **15**(1), 281 (2014)
34. Salmela, L., Sahlin, K., Mäkinen, V., Tomescu, A.I.: Gap filling as exact path length problem. In: Przytycka, T.M. (ed.) RECOMB 2015. LNCS, vol. 9029, pp. 281–292. Springer, Heidelberg (2015)
35. Salzberg, S.L., et al.: GAGE: a critical evaluation of genome assemblies and assembly algorithms. Genome Res. **22**, 557–567 (2012)
36. Simpson, J.T., Durbin, R.: Efficient construction of an assembly string graph using the FM-index. Bioinformatics **26**(12), i367–i373 (2010)
37. Simpson, J.T., Durbin, R.: Efficient de novo assembly of large genomes using compressed data structures. Genome Res. **22**, 549–556 (2012)
38. Tomescu, A.I., Medvedev, P.: Safe and complete contig assembly via omnitigs (2016). http://arxiv.org/abs/1601.02932
39. Uricaru, R., et al.: Reference-free detection of isolated SNPs. Nucleic Acids Res. **43**(2), e11 (2015)

40. Vyahhi, N., Pyshkin, A., Pham, S., Pevzner, P.A.: From de Bruijn graphs to rectangle graphs for genome assembly. In: Raphael, B., Tang, J. (eds.) WABI 2012. LNCS, vol. 7534, pp. 249–261. Springer, Heidelberg (2012)
41. Zerbino, D.R., Birney, E.: Velvet: algorithms for de novo short read assembly using de Bruijn graphs. Genome Res. **18**(5), 821–829 (2008)

Long Single-Molecule Reads Can Resolve the Complexity of the Influenza Virus Composed of Rare, Closely Related Mutant Variants

Alexander Artyomenko[1(✉)], Nicholas C. Wu[2], Serghei Mangul[3], Eleazar Eskin[3], Ren Sun[4], and Alex Zelikovsky[1]

[1] Computer Science Department, Georgia State University, Atlanta, GA 30303, USA
{aartyomenko,alexz}@cs.gsu.edu
[2] Department of Integrative Structural and Computational Biology, The Scripps Research Institute, La Jolla, CA 92037, USA
nicwu@scripps.edu
[3] Computer Science Department, University of California, Los Angeles, Los Angeles, CA 90095, USA
smangul@ucla.edu, eeskin@cs.ucla.edu
[4] Molecular and Medical Pharmacology, University of California, Los Angeles, Los Angeles, CA 90095, USA
rsun@mednet.ucla.edu

Abstract. As a result of a high rate of mutations and recombination events, an RNA-virus exists as a heterogeneous "swarm" of mutant variants. The long read length offered by single-molecule sequencing technologies allows each mutant variant to be sequenced in a single pass. However, high error rate limits the ability to reconstruct heterogeneous viral population composed of rare, related mutant variants. In this paper, we present 2SNV, a method able to tolerate the high error-rate of the single-molecule protocol and reconstruct mutant variants. 2SNV uses linkage between single nucleotide variations to efficiently distinguish them from read errors. To benchmark the sensitivity of 2SNV, we performed a single-molecule sequencing experiment on a sample containing a titrated level of known viral mutant variants. Our method is able to accurately reconstruct clone with frequency of 0.2 % and distinguish clones that differed in only two nucleotides distantly located on the genome. 2SNV outperforms existing methods for full-length viral mutant reconstruction. The open source implementation of 2SNV is freely available for download at http://alan.cs.gsu.edu/NGS/?q=content/2snv.

Keywords: SMRT reads · RNA viral variants · Single nucleotide variation

A. Artyomenko, N.C. Wu and S. Mangul—Equal contributor.

Electronic supplementary material The online version of this chapter (doi:10.1007/978-3-319-31957-5_12) contains supplementary material, which is available to authorized users.

M. Singh (Ed.): RECOMB 2016, LNBI 9649, pp. 164–175, 2016.
DOI: 10.1007/978-3-319-31957-5_12

1 Introduction

Majority of the emerging and re-emerging diseases (influenza, hantaviruses, Ebola virus, and Nipah virus), which represent a global threat to the public health, are caused by RNA viruses [28]. RNA viruses can be featured by their robust adaptability and evolvability due to their high mutation rates and rapid replication cycles [6,15]. This enables a within-host RNA virus population to organize as a complex and dynamic mutant swarm of many highly similar viral genomes. This mutant spectrum, also known as quasispecies [9], is continuously maintained and regenerated during viral infection [7,19]. Deep sequencing has provided a new lens to monitor individual viral variants accelerating the understanding of escape and resistance mechanisms [3,26], in addition to providing insights about the viral evolutionary landscape and the genomic interactions [22,30,39].

Short reads offered by commonly used fragmentation-based protocols are well suited to detect discrete genome components, such as the frequency of each single-nucleotide polymorphism. However, high similarity of the individual viral genomes imposes a huge challenge to assemble discrete components into a population of full-length viral genomes. In particular, mutations are often located on the distances unreachable by the short reads. Therefore even hybrid technologies based on error correction of PacBio reads with Illumina reads were not applied to sequencing of viral variants. Indeed, short reads cannot tell the allele – the same short read is equally well mapped to a variant with the major allele and a variant with the minor allele.

Single Molecule Real Time (SMRT) sequencing is a parallelized single molecule DNA sequencing method. PacBio SMRT sequencing reads are much longer than sequencing reads provided by Illumina, however, its throughput is much lower and the error rate is significantly higher. The read length offered by a single-molecule sequencing protocol [8] is comparable to the genome size of most RNA viruses. It allows each genome variant to be sequenced in a single pass, providing an accurate phasing of the distant mutations. The main drawbacks of the long single-molecule technologies are the high error rate and comparatively low throughput, limiting ability of those technologies to study the heterogeneous viral populations. Thus, a complete profiling of all viral genomes within a mutant spectrum is not yet possible.

Recently, this problem has been addressed using various computational and statistical approaches implemented in Quasirecomb [36], PredictHaplo [32], HaploClique [35], VGA [24], and *k*GEM [33]. These methods perform reasonably well on short reads with high coverage and low error rate, but our experimental validation shows far from satisfactory performance on the sequencing data provided by single-molecule technologies. Also a workflow for reconstruction of closely related variants from raw reads generated during SMRT sequencing was proposed in [4]. Note that a recent method for haplotyping using Pacbio reads proposed in [34] is only applicable for diploid organisms and is not suitable for viral haplotyping with numerous variants.

In this paper, we present **two S**ingle **N**ucleotide **V**ariants (2SNV), a comprehensive method for the accurate reconstruction of the heterogeneous viral population from the long single-molecule reads. The 2SNV method hierarchically clusters together reads containing pairs of correlated (i.e., linked) SNVs until no cluster has correlated SNVs left and outputs consensus of each cluster. It allows to reduce error rate and differentiate true biological variants from sequencing artifacts, thus providing increased accuracy to study diversity and composition of the viral spectrum. To benchmark the sensitivity of 2SNV, we performed a single-molecule sequencing experiment on a sample containing a titrated level of known viral mutant variants. We were able to reconstruct a haplotype with a frequency of 0.2 % and distinguish clones that differed in only two nucleotides. We also showed that 2SNV outperformed existing haplotype reconstruction tools. With a high sensitivity and accuracy, 2SNV is anticipated to facilitate not only viral quasispecies reconstruction, but also other biological questions that require detection of rare haplotypes such as genetic diversity in cancer cell population, and monitoring B-cell and T-cell receptor repertoire.

2 Methods

Any method for reconstruction of viral variants from single-molecule reads should overcome low volume and high error rate of sequencing data combined with very high similarity and very low frequency of viral variants. This challenge is equivalent to extraction of an extremely weak signal from very noisy background with signal-to-noise ratio approaching zero. However impossible this task may seem, a satisfactory solution can be based on distinguishing randomness of the noise from systematic signal repetition. Previously, linkage between SNVs was used for distinguishing sequencing errors from SNVs [23], however, to the best of our knowledge, it was never applied for haplotyping.

Since all reads are from the same RNA region of very similar sequences, they can be reliably aligned to each other. In general, the errors in different positions are independent from each other and the further these positions are from each other the less likely any dependency can be caused by systematic errors. Therefore, even slightly more than expected co-occurrence of two rare alleles in non-adjacent positions may serve as a trustful signature of one or more rare variants having the both rare alleles. Such single nucleotide variations (SNVs) are called linked.

The proposed 2SNV method recursively clusters reads containing pairs of linked SNVs until no pair of SNVs exhibits statistically significant linkage in any cluster. Then each cluster should contain just a single viral variant which can be simply reconstructed as the consensus of all reads in the cluster.

In the remainder of the section we derive statistical conditions of SNV linkage and then give detailed description of the 2SNV method which identifies rare variants based SNV pairs satisfying these conditions.

2.1 Linkage of SNV Pairs

In this section we analyze statistical significance of the linkage between a pair of SNVs which allows to distinguish reads emitted by a rare variant from background errors.

We assume that errors are random and a rare variant has at least 2 mismatches with other variants. Let us consider an arbitrary pair of two distinct positions $I, J \in \{1, \ldots, L\}, I \neq J$, where L be the length of the amplicon (see Fig. 1b). Let I_1 and J_1 be the alleles of the most frequent 2-haplotype $(I_1 J_1)$. Note that $(I_1 J_1)$ should be a 2-haplotype from at least one true viral variant assuming that the error rates in the I-th and J-th positions are small and independent.

Let $I_2 \neq I_1$ and $J_2 \neq J_1$ be the alleles of another 2-haplotype. Let E_{kl}, $k, l \in \{1, 2\}$, be the expected number of reads with 2-haplotypes $(I_k J_l)$. The following theorem can be used to decide if the haplotype $I_2 \neq I_1$ exists.

Theorem 1. *Assume that the sequencing error is random, independent and does not exceed 50 %. If no viral variant with the haplotype $(I_2 J_2)$ exists, then the expected value of E_{22} is at most*

$$E_{22} \leq \frac{E_{21} \cdot E_{12}}{E_{11}} \tag{1}$$

The inequality (1) becomes an equality if at least one of 2-haplotypes $(I_1 J_2)$ or $(I_2 J_1)$ also does not exist.

Proof. Let ε_I^{kl} and ε_J^{kl}, $k, l \in \{1, 2\}$, be the probabilities to observe the allele l instead of the true allele k in the positions I and J, respectively. We are not going to estimate the parameters ϵ_I^{kl}. The model only assumes that these parameters are random, independent, and do not exceed 50 %.

Let T_{kl}, $k, l \in \{1, 2\}$, be the true count of 2-haplotypes $(I_k J_l)$. Then error randomness and independence imply that

$$E_{kl} = \sum_{m,n=1,2} \varepsilon_I^{mk} \varepsilon_J^{nl} T_{mn}$$

In order to prove (1), it is sufficient to show that $E_{11} \cdot E_{22} \leq E_{12} \cdot E_{21}$ assuming that $T_{22} = 0$. Indeed,

$$\begin{aligned} E_{11} \cdot E_{22} &= \sum_{m,n=1,2} \varepsilon_I^{m1} \varepsilon_J^{n1} T_{mn} \cdot \sum_{m,n=1,2} \varepsilon_I^{m2} \varepsilon_J^{n2} T_{mn} \\ &= \varepsilon_I^{11} \varepsilon_J^{11} \varepsilon_I^{12} \varepsilon_J^{12} T_{11}^2 + \varepsilon_I^{21} \varepsilon_J^{11} \varepsilon_I^{22} \varepsilon_J^{12} T_{21}^2 + \varepsilon_I^{11} \varepsilon_J^{21} \varepsilon_I^{12} \varepsilon_J^{22} T_{12}^2 \\ &\quad + (\varepsilon_I^{11} \varepsilon_J^{11} \varepsilon_I^{12} \varepsilon_J^{22} + \varepsilon_I^{11} \varepsilon_J^{21} \varepsilon_I^{12} \varepsilon_J^{12}) T_{11} T_{12} \\ &\quad + (\varepsilon_I^{11} \varepsilon_J^{11} \varepsilon_I^{22} \varepsilon_J^{12} + \varepsilon_I^{21} \varepsilon_J^{11} \varepsilon_I^{12} \varepsilon_J^{12}) T_{11} T_{21} \\ &\quad + (\varepsilon_I^{11} \varepsilon_J^{21} \varepsilon_I^{22} \varepsilon_J^{12} + \varepsilon_I^{21} \varepsilon_J^{11} \varepsilon_I^{12} \varepsilon_J^{22}) T_{12} T_{21} \end{aligned}$$

$$
\begin{aligned}
E_{12} \cdot E_{21} &= \sum_{m,n=1,2} \varepsilon_I^{m1}\varepsilon_J^{n2} T_{mn} \cdot \sum_{m,n=1,2} \varepsilon_I^{m2}\varepsilon_J^{n1} T_{mn} \\
&= \varepsilon_I^{11}\varepsilon_J^{12}\varepsilon_I^{12}\varepsilon_J^{11} T_{11}^2 + \varepsilon_I^{21}\varepsilon_J^{12}\varepsilon_I^{22}\varepsilon_J^{11} T_{21}^2 + \varepsilon_I^{11}\varepsilon_J^{22}\varepsilon_I^{12}\varepsilon_J^{21} T_{12}^2 \\
&\quad + (\varepsilon_I^{11}\varepsilon_J^{12}\varepsilon_I^{12}\varepsilon_J^{21} + \varepsilon_I^{11}\varepsilon_J^{22}\varepsilon_I^{12}\varepsilon_J^{11}) T_{11}T_{12} \\
&\quad + (\varepsilon_I^{11}\varepsilon_J^{12}\varepsilon_I^{22}\varepsilon_J^{11} + \varepsilon_I^{21}\varepsilon_J^{12}\varepsilon_I^{12}\varepsilon_J^{11}) T_{11}T_{21} \\
&\quad + (\varepsilon_I^{11}\varepsilon_J^{22}\varepsilon_I^{22}\varepsilon_J^{11} + \varepsilon_I^{21}\varepsilon_J^{12}\varepsilon_I^{12}\varepsilon_J^{21}) T_{12}T_{21}
\end{aligned}
$$

Note that only coefficients for $T_{12}T_{21}$ are different for these products. Therefore, if either $T_{12} = 0$ or $T_{21} = 0$, then $E_{11} \cdot E_{22} = E_{12} \cdot E_{21}$. Otherwise, let all three 2-haplotypes (I_1J_1), (I_1J_2), and (I_2J_1) exist. Then

$$
\begin{aligned}
&E_{12}E_{21} - E_{11}E_{22} \\
&= (\varepsilon_I^{11}\varepsilon_J^{22}\varepsilon_I^{22}\varepsilon_J^{11} + \varepsilon_I^{21}\varepsilon_J^{12}\varepsilon_I^{12}\varepsilon_J^{21} - \varepsilon_I^{11}\varepsilon_J^{21}\varepsilon_I^{22}\varepsilon_J^{12} - \varepsilon_I^{21}\varepsilon_J^{11}\varepsilon_I^{12}\varepsilon_J^{22}) T_{12}T_{21} \\
&= \left(1 - \frac{\varepsilon_I^{12}}{\varepsilon_I^{11}}\frac{\varepsilon_I^{21}}{\varepsilon_I^{22}}\right)\left(1 - \frac{\varepsilon_J^{12}}{\varepsilon_J^{11}}\frac{\varepsilon_J^{21}}{\varepsilon_J^{22}}\right)\varepsilon_I^{11}\varepsilon_J^{22}\varepsilon_I^{22}\varepsilon_J^{11} T_{12}T_{21} > 0
\end{aligned}
$$

The last inequality holds since observing the true allele is more probable than observing the erroneous allele and, therefore, $\varepsilon_I^{kl} < \varepsilon_I^{kk}$ and $\varepsilon_J^{kl} < \varepsilon_J^{kk}$, $k, l \in \{1, 2\}$. QED

Note that Theorem 1 does not require linkage disequilibrium of haplotypes - the lack of linkage is explained by errors. The 2SNV method uses Theorem 1 to decide if the alleles I_2 and J_2 are linked as follows. Let O_{kl}, $k, l \in \{1, 2\}$, be the observed number of reads with 2-haplotypes (I_kJ_l). Let n be the total number of reads covering the both positions I and J, then

$$
p = \frac{O_{21} \cdot O_{12}}{O_{11} \cdot n} \tag{2}
$$

is the largest probability of observing the 2-haplotype (I_2J_2) among these n reads. The probability to observe at least O_{22} reads in the (n, p) binomial distribution equals

$$
Pr(X \geq O_{22}) = 1 - \sum_{i=1}^{O_{22}-1} \binom{n}{i} p^i (1-p)^{n-i} \tag{3}
$$

Since we are looking for a pair of SNVs among $\binom{L}{2}$ possible pairs, we also adjust to multiple testing using Bonferroni correction requiring

$$
1 - \sum_{i=1}^{O_{22}-1} \binom{n}{i} p^i (1-p)^{n-i} \leq \frac{\mathcal{P}}{\binom{L}{2}} \tag{4}
$$

where p is defined in (2) and $\mathcal{P}$ is the user-defined P-value, by default $\mathcal{P} = 0.01$.

Finally, when the cluster is too small, the statistical test (4) may be not stringent enough to weed out spurious linkages. Therefore, we require the number

of reads O_{22} to be at least an empirically defined value (by default equal 30), in order to decide whether there is an additional haplotype producing these reads.

Note that the binomial model used in (4) may not be stringent enough to compensate for reducing PPV caused by overdispersion especially for higher coverage. In future releases of our tool we plan to take in account additional variance modeling unknown experimental data processes contributing to variance, e.g., replacing the binomial distribution with the beta-binomial distribution.

2.2 2SNV Method for Viral Variant Reconstruction

The input to 2SNV consists of a set of aligned PacBio reads (see Fig. 1(a)). Alignment required to be in a form of multiple sequence alignment (MSA). The MSA algorithms are too slow to handle PacBio datasets, so instead, we use pairwise alignment by BWA [20] and b2w from Shorah [41] to transform pairwise alignment to MSA format.

The main novel step of the 2SNV algorithm identifies a pair of linked SNVs (see Fig. 1(b)). with higher than expected portion of reads containing the 2-haplotype with the both minor alleles according to (2–4).

The 2SNV method maintains a partition of all reads into clusters. Each cluster is assumed to consist of the reads emitted by the single variant coinciding with the cluster consensus (see Fig. 1(c)). Until no pair of SNVs in the cluster C is linked, we recursively partition C into two clusters C_1 and C_2. C_1 consists of reads with the linked pair of SNVs C_2 consists of the remaining reads of C.

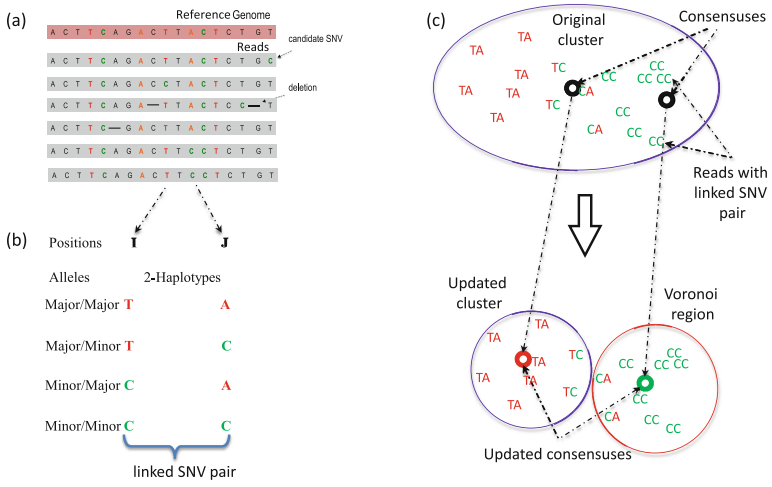


Fig. 1. Overview of the 2SNV method: (a) multiple sequence alignment of reads from the same amplicon; (b) identification of a linked SNV pair in positions I and J; (c) recursive cluster splitting: (i) finding consensus of reads with the linked SNV pair, (ii) finding Voronoi region of this consensus, (iii) update the original cluster and the consensuses for the two new clusters.

Algorithm 1. 2SNV Algorithm

procedure 1: constructing the consensus haplotype for all reads:
Initialize the set of all clusters with a single cluster with all reads $\mathcal{C} \leftarrow \{R\}$
For each position i find allele of highest frequency a_i
$Consensus(C) \leftarrow (a_1, \ldots, a_L)$
procedure 2: partitioning reads into simple clusters
while not all clusters are simple **do**
for each non-simple cluster $C \in \mathcal{C}$ **do**
if no pair SNVs is linked according to (2-4) **then**
Regard C as a simple cluster
else
Find a pair of linked SNVs I_2 and J_2 minimizing (3)
Find the set C_1 of all reads with the 2-haplotype $(I_2 J_2)$
Find the consensus $c_1 \leftarrow Consensus(C_1)$
$C_1 \leftarrow Voronoi(c_1)$
$C_2 \leftarrow C \setminus C_1$, $c_2 \leftarrow Consensus(C_2)$
$\mathcal{C} \leftarrow \mathcal{C} \cup \{C_1\} \cup \{C_2\} \setminus \{C\}$
procedure 3: estimating frequencies of the consensuses of simple clusters
Run kGEM algorithm for the set of haplotypes $\{Consensus(C), C \in \mathcal{C}\}$.

We further modify C_1 and C_2 by replacing them with the Voronoi regions of their consensuses, where *the Voronoi region* of the consensus c_1 of C_1 consists of reads that are closer to c_1 than to the consensus of C_2. Finally, kGEM finds maximum likelihood estimates of frequencies of haplotypes represented by cluster consensuses using expectation-maximization algorithm [33].

Algorithm 1 describes the formal pseudocode of the 2SNV algorithm.

3 Results

We were using 3 datasets: PacBio reads from a single IAV clone and 10 IAV clones, and simulated PacBio reads from 20 HCV clones.

Error-prone PCR was performed on the influenza A virus (A/WSN/33) PB2 segment using GeneMorph II Random Mutagenesis Kits (Agilent Technologies, Westlake Village, CA) according to manufacturer's instruction. The 2 kb region was amplified from the IAV viral population and subjected to PacBio RS II sequencing using 2 SMRT cells with P4-C2. The average read length was 1973 bp and ranges from 200 bp to 5 kb. Some reads are much longer than the amplified region due to long insertions which are sequencing errors. Raw sequencing data have been submitted to the NIH Short Read Archive (SRA) under accession number: BioProject PRJNA284802. The nucleotide sequences of the 10 clones are freely available at http://alan.cs.gsu.edu/NGS/?q=content/2snv.

The Dataset with a Single IAV Clone. There total number of reads were 11,907 and the average Hamming distance between the true haplotype and reads is 14.4 %.

The Dataset with 10 IAV Clones. 10 independent clones, ranging from 1 to 13 mutations from the original single were selected. These 10 clones were mixed at a geometric ratio with two-fold difference in occurrence frequency for consecutive clones starting with the maximum frequency of 50 % and the minimum frequency of 0.1 %. The pairwise edit distance between clones are given in the heat-map on Fig. 2 in Supplement. In total, there were 33,558 reads generated from 10 clones.

The Simulated Dataset with 20 HCV Clones. 21K simulated PacBio reads were generated from 1739-bp long fragment from the E1E2 region of 20 HCV sequences [38] using simulator pbsim [29]. The reads were simulated with mean accuracy 98 % and minimum accuracy 95 % reflecting advancements in PacBio technology. We have generated reads 10 times for two distributions of the clone frequencies – uniform (all frequencies are 5 %) and skewed (a single clone has 90.5 % and every other clone has frequency 0.5 %).

3.1 Reconstruction of Viral Variants

2SNV was compared with 2 tools originally tuned to handle HIV variants (PredictHaplo [32] and Quasirecomb [36]) and *k*GEM [33] tuned for a short HCV amplicon. We could not compare with HaploClique [35] since it is no longer maintained by the authors. A workflow [4] is not currently available and we were not able to run it on our data. Also the experimental data in [4] are also not fully available and we were not able to run 2SNV on these data.

For the dataset with a single IAV clone 2SNV, *k*GEM, and PredictHaplo were able to reconstruct no more a single variant which perfectly matches the original clone. Quasirecomb reported multiple variants none fully matching the original clone.

For the dataset with 10 IAV clones, 2SNV reported 10 haplotypes: the 9 most frequent haplotypes exactly matching 9 most frequent clones and the least frequent haplotype (1 %) not matching any clone. The correlation between the estimated and true frequencies of the 9 correctly reconstructed haplotypes is 99.4 %. PredictHaplo was able to reconstruct only 6 true variants missing 4 variants with total frequency of 8 % while not having any false positives. In order to reliably compare the reconstruction rate of two methods, we have applied them to 40 sub-samples of the original data (each subsample consists of 33558 reads randomly selected with repetition from the original data). The results are presented on Fig. 4 and Table 1 in Supplement. *k*GEM was able to reconstruct only 2 most frequent clones and Quasirecomb failed to reconstruct even a single clone.

In order to estimate how accuracy of reconstruction methods depends on the coverage, we have randomly sub-sampled N reads ($N = 500$, 1000, 2000, 4000, 8000, 16000) from the original 33558 reads and run 2SNV and PredictHaplo. The results are shown on Fig. 2 in Supplement. For each coverage and each clone (except Clone5), 2SNV more accurately estimates the frequency. Clone6 and Clone8 for all sub-samples, Clone4 for $N \leq 8000$ and Clone 3 for $N \leq 1000$

are missed by PredictHaplo but reconstructed by 2SNV. Clone6 which is only two mutations away from the more frequent Clone5 was successfully reconstructed for $N \geq 4000$ while PredictHaplo was never able to reconstruct Clone6. Note that since these 2 SNVs between Clone5 and Clone6 are far apart, only long reads can reconstruct this rare variant. From the last plot one can see that the false positive rate for PredictHaplo is also higher than for 2SNV, e.g. 2SNV does not report false positives for $N \leq 8000$. The averages of all runs are given in Table 2 in Supplement.

For the simulated dataset with 20 HCV variants, we have compared 2SNV only with PredictHaplo. For the uniform frequency distribution the average sensitivity and PPV for 2SNV are 85 % and 100 %, respectively, while for PredictHaplo the corresponding values are 72 % and 53 %, respectively. For the skewed frequency distribution, the average sensitivity and PPV for 2SNV are 99 % and 69 %, respectively, while for PredictHaplo the corresponding values are 36 % and 46 %, respectively.

Runtime. The runtime of 2SNV is linear with respect to the number of reads, however implementation is $O(nlogn)$ due to parallelization (see Fig. 5 in Supplement) and quadratic with respect to the length of the amplicon region. For all experiments we used the same PC (Intel(R) Xeon(R) CPU X5550 2.67 GHz x2 8 cores per CPU, DIMM DDR3 1333 MHz RAM 4Gb x12) with operating system CentOS 6.4.

4 Discussion

Haplotype phasing represents one of the biggest challenges in next-generation sequencing due to the short read length. The recent development of single-molecule sequencing platform produces reads that are sufficiently long to span the entire gene or small viral genome. It not only benefits the assembly of genomic regions with tandem repeat [5,18,37], but also offers the opportunity to examine the genetic linkage between mutations. In fact, it is shown that the long read in single-molecule sequencing aids haplotype phasing in diploid genome [31], and in polyploid genome [1]. Nonetheless, the sequencing error rate of single-molecule sequencing platform is extremely high (≈14 % as estimated by this study), which hampers its ability to reconstruct rare haplotypes. This drawback prohibits single-molecule sequencing platform from applications in which a high sensitivity of haplotypes are needed, such as quasispecies reconstruction. In this study, we have developed 2SNV, which allows quasispecies reconstruction using single-molecule sequencing despite the high sequencing error rate. The high sensitivity of 2SNV permits the detection of extremely rare haplotypes and distinguish between closely related haplotypes. Based on titrated levels of known haplotypes, we demonstrates that 2SNV is able to detect a haplotype that has a frequency as low as 0.2 %. This sensitivity is comparable to many deep sequencing-based point mutation detection methods [10,11,13,21]. In addition, 2SNV successfully distinguishes between Clone5 and Clone6 in this study, which are only two nucleotides away from each other. It highlights the sensitivity of

2SNV to distinguish closely related haplotypes. Our results also show that the sensitivity is coverage-dependent, implying that the sensitivity of 2SNV may further improve when sequencing depth increases. Therefore, the constant increase of sequencing throughput offered by single-molecule sequencing technology provides the unprecedented resolution promising to increase number of discovered rare haplotypes.

The ability to accurately determine the genomic composition of the viral populations and identify closely related viral genomes makes our tool applicable for dissecting evolutionary trajectories and examining mutation interactions in RNA viruses. Evolutionary trajectories and mutation interactions have been shown to play an important role in viral evolution, such as drug resistance [2,3,26,39], immune escape [12], and cross-species adaptation [14,16]. An unbiased and accurate understanding of the genomic composition of the RNA viruses opens a new avenue to study the underlying mechanism of adaptation, persistence and virulence factors of the pathogen, which are yet to be comprehended.

While viral quasispecies reconstruction is used as a proof-of-concept in this study, the application of 2SNV can be extended to detect haplotype variants in any sample with high genetic heterogeneity and diversity, such as B-cell and T-cell receptor repertoire, cancer cell populations, and metagenomes. It is shown that monitoring B-cell and T-cell receptor repertoire helps investigate virus-host interaction dynamics [17,27,40,42,43]. Furthermore, examining the genetic composition of the cancer cell populations in high sensitivity can facilitate diagnosis and treatment [25]. Therefore, we anticipate that 2SNV will benefit different subfields of biomedical research in the genomic era. We also propose that 2SNV can be applied to increase the resolution of metagenomics profiling from species level to strain level. In summary, 2SNV is a widely applicable tool as single-molecule sequencing technology being popularized.

Supplement. The Supplement to this paper containing Figs. 1–6 and Tables 1 and 2 is available here: http://alan.cs.gsu.edu/NGS/?q=content/2snv_supplement

Acknowledgments. We would like to thank H. Hao for performing the PacBio sequencing at Johns Hopkins Deep Sequencing & Microarray Core Facility. A.A. was supported by GSU Molecular Basis of Disease Fellowship. S.M. and E.E were supported by National Science Foundation grants 0513612, 0731455, 0729049, 0916676, 1065276, 1302448 and 1320589, and National Institutes of Health grants K25-HL080079, U01-DA024417, P01- HL30568, P01-HL28481, R01-GM083198, R01-MH101782 and R01-ES022282. S.M. was supported in part by Institute for Quantitative & Computational Biosciences Fellowship, UCLA.

References

1. Aguiar, D., Istrail, S.: Haplotype assembly in polyploid genomes and identical by descent shared tracts. Bioinformatics **29**(13), i352–i360 (2013)
2. Beerenwinkel, N., et al.: Diversity and complexity of HIV-1 drug resistance: a bioinformatics approach to predicting phenotype from genotype. Proc. Natl. Acad. Sci. **99**(12), 8271–8276 (2002)

3. Bushman, F.D., et al.: Massively parallel pyrosequencing in HIV research. Aids **22**(12), 1411–1415 (2008)
4. Dilernia, D.A., et al.: Multiplexed highly-accurate DNA sequencing of closely-related HIV-1 variants using continuous long reads from single molecule, real-time sequencing. Nucleic Acids Res. **43**(20), e129 (2015)
5. Doi, K., et al.: Rapid detection of expanded short tandem repeats in personal genomics using hybrid sequencing. Bioinformatics **30**(6), 815–822 (2014)
6. Domingo, E.: Mutation rates and rapid evolution of RNA viruses. In: Morse, S.S. (ed.) The Evolutionary Biology of Viruses, pp. 161–184. Raven Press, New York (1994)
7. Domingo, E., Holland, J.: RNA virus mutations and fitness for survival. Annu. Rev. Microbiol. **51**(1), 151–178 (1997)
8. Eid, J., et al.: Real-time dna sequencing from single polymerase molecules. Science **323**(5910), 133–138 (2009)
9. Eigen, M.: Selforganization of matter and the evolution of biological macromolecules. Naturwissenschaften **58**(10), 465–523 (1971)
10. Flaherty, P., et al.: Ultrasensitive detection of rare mutations using next-generation targeted resequencing. Nucleic Acids Res. **40**(1), e2 (2012)
11. Forshew, T., et al.: Noninvasive identification and monitoring of cancer mutations by targeted deep sequencing of plasma DNA. Sci. Transl. Med. **4**(136), 136ra68 (2012)
12. Goepfert, P.A., et al.: Transmission of HIV-1 Gag immune escape mutations is associated with reduced viral load in linked recipients. J. Exp. Med. **205**(5), 1009–1017 (2008)
13. Harismendy, O., et al.: Detection of low prevalence somatic mutations in solid tumors with ultra-deep targeted sequencing. Genome Biol. **12**(12), R124 (2011)
14. Herfst, S., et al.: Airborne transmission of influenza A/H5N1 virus between ferrets. Science **336**(6088), 1534–1541 (2012)
15. Holland, J., et al.: Rapid evolution of RNA genomes. Science **215**(4540), 1577–1585 (1982)
16. Imai, M., et al.: Experimental adaptation of an influenza H5 HA confers respiratory droplet transmission to a reassortant H5 HA/H1N1 virus in ferrets. Nature **486**(7403), 420–428 (2012)
17. Klarenbeek, P.L., et al.: Deep sequencing of antiviral T-cell responses to HCMV and EBV in humans reveals a stable repertoire that is maintained for many years. PLoS Pathog **8**(9), e1002889 (2012)
18. Schrago, C.G., Carvalho, A.B.: Long-read single molecule sequencing to resolve tandem gene copies: the Mst77Y region on the drosophila melanogaster Y chromosome. G3 (Bethesda) **5**(6), 1145–1150 (2015)
19. Lauring, A.S., Andino, R.: Quasispecies theory and the behavior of RNA viruses. PLoS Pathog **6**(7), e1001005 (2010)
20. Li, H., Durbin, R.: Fast and accurate short read alignment with burrows-wheeler transform. Bioinformatics **25**(14), 1754–1760 (2009)
21. Li, M., Stoneking, M.: A new approach for detecting low-level mutations in next-generation sequence data. Genome Biol. **13**(5), R34 (2012)
22. Liu, J., et al.: Analysis of low-frequency mutations associated with drug resistance to raltegravir before antiretroviral treatment. Antimicrob. Agents Chemother. **55**(3), 1114–1119 (2011)
23. Macalalad, A.R., et al.: Highly sensitive and specific detection of rare variants in mixed viral populations from massively parallel sequence data. PLoS Comput. Biol. **8**(3), e1002417 (2012)

24. Mangul, S., et al.: Accurate viral population assembly from ultra-deep sequencing data. Bioinformatics **30**(12), i329–i337 (2014)
25. Mardis, E.R., Wilson, R.K.: Cancer genome sequencing: a review. Hum. Mol. Genet. **18**(R2), R163–168 (2009)
26. Margeridon-Thermet, S., et al.: Ultra-deep pyrosequencing of hepatitis B virus quasispecies from nucleoside and nucleotide reverse-transcriptase inhibitor (NRTI)-treated patients and NRTI-naive patients. J. Infect. Dis. **199**(9), 1275–1285 (2009)
27. Miconnet, I.: Probing the T-cell receptor repertoire with deep sequencing. Curr. Opin. HIV AIDS **7**(1), 64–70 (2012)
28. Murphy, F.A., Kingsbury, D.W.: Virus taxonomy. Fields Virol. **2**, 15–57 (1996)
29. Asai, K., Hamada, M.: PBSIM: PacBio reads simulator toward accurate genome assembly. Bioinformatics **29**(1), 119–121 (2013)
30. Palmer, S., et al.: Selection and persistence of non-nucleoside reverse transcriptase inhibitor-resistant HIV-1 in patients starting and stopping non-nucleoside therapy. Aids **20**(5), 701–710 (2006)
31. Pendleton, M., et al.: Assembly and diploid architecture of an individual human genome via single-molecule technologies. Nat. Methods **12**, 780–786 (2015)
32. Beerenwinkel, N., Roth, V.: HIV haplotype inference using a propagating Dirichlet process mixture model. IEEE/ACM Trans. Computat. Biol. Bioinform. (TCBB) **11**(1), 182–191 (2014)
33. Skums, P., et al.: Computational framework for next-generation sequencing of heterogeneous viral populations using combinatorial pooling. Bioinformatics **31**(5), 682–690 (2015)
34. Sharon, D., Snyder, M.P.: Defining a personal, allele-specific, and single-molecule long-read transcriptome. Proc. Natl. Acad. Sci. **111**(27), 9869–9874 (2014)
35. Töpfer, A., Marschall, T., Bull, R.A., Luciani, F., Schönhuth, A., Beerenwinkel, N.: Viral quasispecies assembly via maximal clique enumeration. In: Sharan, R. (ed.) RECOMB 2014. LNCS, vol. 8394, pp. 309–310. Springer, Heidelberg (2014)
36. Töpfer, A., et al.: Probabilistic inference of viral quasispecies subject to recombination. J. Comput. Biol. **20**(2), 113–123 (2013)
37. Ummat, A., Bashir, A.: Resolving complex tandem repeats with long reads. Bioinformatics **30**(24), 3491–3498 (2014)
38. Von Hahn, T., et al.: Hepatitis C virus continuously escapes from neutralizing antibody and T-cell responses during chronic infection in vivo. Gastroenterology **132**(2), 667–678 (2007)
39. Ronaghi, M., Shafer, R.: Characterization of mutation spectra with ultra-deep pyrosequencing: application to HIV-1 drug resistance. Genome Res. **17**(8), 1195–1201 (2007)
40. Wu, X., et al.: Focused evolution of HIV-1 neutralizing antibodies revealed by structures and deep sequencing. Science **333**(6049), 1593–1602 (2011)
41. Eriksson, N., Beerenwinkel, N.: Shorah: estimating the genetic diversity of a mixed sample from next-generation sequencing data. BMC Bioinform. **12**(1), 119 (2011)
42. Zhu, J., et al.: Mining the antibodyome for HIV-1-neutralizing antibodies with next-generation sequencing and phylogenetic pairing of heavy/light chains. Proc. Natl. Acad. Sci. U.S.A. **110**(16), 6470–6475 (2013)
43. Zhu, J., et al.: De novo identification of VRC01 class HIV-1-neutralizing antibodies by next-generation sequencing of B-cell transcripts. Proc. Natl. Acad. Sci. U.S.A. **110**(43), E4088–4097 (2013)

Structural Variation Detection with Read Pair Information—An Improved Null-Hypothesis Reduces Bias

Kristoffer Sahlin[1(✉)], Mattias Frånberg[2,3], and Lars Arvestad[3,4]

[1] KTH Royal Institute of Technology, Science for Life Laboratory, School of Computer Science and Communication, Stockholm, Sweden
ksahlin@kth.se

[2] Atherosclerosis Research Unit, Department of Medicine, Solna, Karolinska Institutet, Stockholm, Sweden
mattias.franberg@ki.se

[3] Department of Numerical Analysis and Computer Science, Stockholm University, Stockholm, Sweden
arve@nada.su.se

[4] Swedish e-Science Research Centre (SeRC), Stockholm, Sweden

Abstract. Reads from paired-end and mate-pair libraries are often utilized to find structural variation in genomes, and one common approach is to use their fragment length for detection. After aligning read-pairs to the reference, read-pair distances are analyzed for statistically significant deviations. However, previously proposed methods are based on a simplified model of observed fragment lengths that does not agree with data. We show how this model limits statistical analysis of identifying variants and propose a new model, by adapting a model we have previously introduced for contig scaffolding, which agrees with data. From this model we derive an improved null hypothesis that, when applied in the variant caller CLEVER, reduces the number of false positives and corrects a bias that contributes to more deletion calls than insertion calls. A reference implementation is freely available at https://github.com/ksahlin/GetDistr.

1 Introduction

Genomic structural variation, for example insertion and deletion of DNA, are common in the human population and have been linked to various diseases and conditions. The basic question scientists and clinicians want to answer is: given a DNA sample from a donor and a suitable reference genome, what structural variants does the donor have in comparison to the reference? Methods for identifying structural variants are continuously worked on, in terms of both experimental protocols and bioinformatic analysis. Short-read technologies are, despite their weaknesses, the primary data source because of the superior throughput/cost ratio. It is today important to improve accuracy of predictions and in particular to reduce the false-positive rate while retaining sensitivity. To that end, we have

M. Singh (Ed.): RECOMB 2016, LNBI 9649, pp. 176–188, 2016.
DOI: 10.1007/978-3-319-31957-5_13

worked on improving the statistical analysis of paired reads, using paired-end (PE) or mate-pair (MP) libraries, for evaluating the significance of a detected insertion or deletion.

While aligned reads are important for identifying short variants and substitutions, larger variants and variants in repetitive regions where alignment is difficult are easier detected by paired reads spanning over the region. In PE and MP protocols, reads are from the ends of DNA *fragments* from the donor. PE libraries have short-range fragment lengths (up to 100 s bp), MP libraries are long range (1000 s bp), and they each have their own strengths and limitations. PE libraries often has superior coverage and narrow fragment length distribution while long range MP libraries can span larger insertions and, at similar read coverage, provide higher span coverage (the number of MP pairs separated by a random position) than PE libraries, which in theory can make up for the increased variation in individual fragment lengths by increasing statistical power from more observations.

Numerous structural variation algorithms using read pairs, and their fragment length, to detect variants have been proposed. Many tools use only *discordant* read pairs for downstream calling of variants, i.e., read pairs that align at a distance smaller than $\mu - k\sigma$ or larger than $\mu + k\sigma$ base pairs from each other, where μ and σ are the mean and standard deviation of the fragment length distribution and $k \in \mathbb{R}$ [2,5,11,12,20]. This restriction may reduce the computational demand, but it sacrifices sensitivity [17] by removing observations.

There are also tools with a statistical model/approach that utilizes all read pairs. CLEVER [17] finds insertions and deletions based on statistically significant deviation of the mean fragment length of all reads[1] over a position from μ. This method finds more and smaller variants compared to methods that use only discordant reads [17]. [9] models the number of discordant and concordant read pairs (classified by a cutoff) over a region as following a binomial distribution and finds inversions and deletions based on statistically significant accumulation of discordant read pairs over regions. However, any binary classification cutoff causes loss of information [8], thus statistical power, as they do not consider how much above or below the cutoff a fragment length is[2]. Another approach is non-parametric testing of the distribution over a region, *e.g.*, using the Kolmogorov-Smirnov test [15], but as [17] noted, this is computationally expensive. [10] presented a model to find the most likely common deletion length from several donor genomes with different fragment length distributions by maximizing the likelihood of observed fragment lengths given a deletion size and each of the distributions.

[1] With some modifications to account for heterozygous variants. Only reads that have enough overlap and similar fragment lengths are grouped together.

[2] Under a normal distribution, 100 continuous observations are statistically equivalent to 158 binary observations for the best possible "cut point", which is the mean. The loss of information becomes worse the further away the cut point is from the mean, *e.g.*, $\mu \pm k\sigma$, as k increases. In practice $k \in [3, 6]$ in variant detection tools.

These methods however assumes that the probability of a fragment length being observed over a position/region follows the probability distribution of the full library fragment length distribution, which is not true [22]. Longer fragment lengths span more positions than shorter fragment lengths, so over any position in the genome there will be a bias towards read-pairs further apart than μ. This observation bias of fragment sizes has been investigated earlier in an assembly context, estimating the gap size between contigs [18,21,22]. The approaches given in [18,21] are more general by using the exact (empirical) distribution over the fragment length, which also makes them computationally demanding. GapEst [22] assumes a normal fragment length distribution and derives an analytic expression for the likelihood of a gap size that scales very well, which opens up for other applications where this type of problem needs to be calculated for a large number of instances, *e.g.*, structural variation detection. There is no previous work known to the authors on incorporating this model, or a similar one, to structural variation and investigating how it affects the balance between detecting deletions and insertions.

1.1 Contribution

We use the statistical model given in [22] and present it in the context of structural variation detection. The model provides a probability distribution for the fragment sizes we observe over a position (*e.g.,* a potential *breakpoint*) or region. Given this distribution we derive a null-hypothesis distribution to detect variants. We show that the corrected null-hypothesis agrees with both simulated and biological data, while a commonly used null-hypothesis does not. We implement the null-hypothesis in the state-of-the-art fragment-length based variant caller CLEVER [17]. Although CLEVER uses constraints and assumptions that do not agree with our model, we show that the detection of insertions and deletions becomes more balanced and that the number of false positive calls decreases. This is a promising first result as we could only apply a part of our theory in CLEVER without a significant restructuring of the code. We also believe that this work is a step towards creating a statistical rigorous approach for read pair fragment lengths where we can detect indels to a much higher resolution than cutoff based ones.

In some places, we have to refer to an Appendix, which can be found in a bioRxiv (http://dx.doi.org/10.1101/036707) version of this paper.

2 Methods

We will review a model used to determine contig distances in scaffolding [22] and use it in the context of structural variation detection. Notation and assumptions are presented in Sect. 2.1. In Sect. 2.2 we present the probability distribution in a structural variation detection context. Section 2.4 discuss a commonly used null-hypothesis used for detecting variants with fragment length and derives an improved null-hypothesis using our model.

2.1 Notation and Assumption

We refer to our model as the *Observed Fragment Length* (OFL) model. This model carries no new concepts and makes the same assumptions as the Lander-Waterman model [14], but adds a variable and some constants. We only state it here for convenience of referencing to a model when we derive probabilities and a null-hypothesis. Read pairs are sampled independently and uniformly from the donor genome. Let G denote the length of the reference genome. Alignment of read pairs to the reference genome yields our observations: distance o between reads in a read pair, read length r, and number of allowed "inner"[3] softclipped bases s [16] in an alignment, see Fig. 1a. Read-pair distances x come from a library fragment length distribution $f(x)$ (either given or estimated from alignments). We denote the mean and standard deviation of this distribution as μ and σ. Finally, a parameter δ models the number of missing or added base pairs in the reference, compared to the donor sequence. That is, if the donor sequence contains an insertion, δ is negative and we say that the donor sequence has δ added bases. Similarly, if the donor sequence contains a deletion, δ is positive and we say that the donor sequence has δ deleted bases. For a given read-pair with fragment length x, let $w_{G,p}(x)$ denote the probability that it spans over position p on a genome of size G. As we do not model that any two positions have different probabilities to be spanned over (reads are drawn uniformly), w will not depend on p and we omit it and refer to $w_G(x)$ from now on.

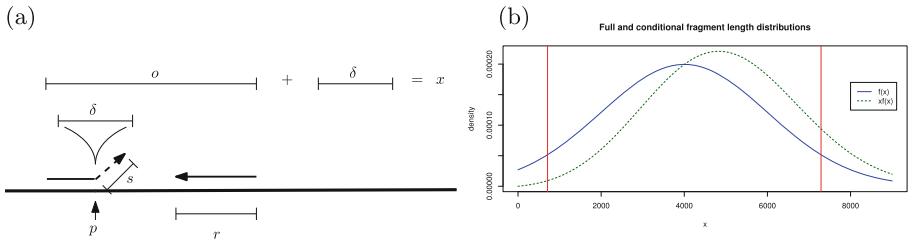


Fig. 1. (a) Constants and variables in the OFL model. The figure illustrates the scenario of an insertion in the donor genome of length δ at position p. Two reads (marked by arrows) of length r are at distance o from each other, with the left read partially aligned leaving s positions unaligned (softclipped). (b) Illustration of a full fragment size distribution $f(x)$ from $N(4000, 2000)$ (blue line), from which H_0 is derived. The green dashed line shows the observed fragment distribution over any variant free position for fragments coming from $f(x)$ for the simplified case when $r, s, \delta = 0$ (*i.e.*, this is exactly the function $xf(x)$). Red lines indicate the $\mu \pm 2\sigma$ quantiles of $f(x)$. It is less likely to observe a smaller fragment size over any given position in the genome (see density of green distribution at red lines), as opposed to identical significance under $f(x)$.

[3] We call the side of the read that is closest to its mate "inner".

2.2 Probability Function over Observed Fragment Lengths

The distribution and probabilities derived in this section closely matches those given in [22] with the minor addition of the constant s. We restate the expressions in a structural variation context for clarity.

No Variant. First, we assume that donor and reference are identical, therefore $\delta = 0$ at any position. Given the OFL model, the probability that we observe a read pair with fragment size x over a position p on a genome of length G is

$$P(x|\delta = 0) = \frac{w_G(x)f(x)}{\sum_x w_G(x)f(x)} = \frac{\frac{(x-2(r-s))}{G}f(x)}{\sum_x \frac{(x-2(r-s))}{G}f(x)}. \quad (1)$$

Here $f(x)$ is the probability to draw a fragment of length x from the full library and $w_G(x)$ the probability that it spans over position p. The denominator is a normalization constant to make P a probability. It is assumed that $x \geq 2(r-s)$. For example, if the read length is 100 and maximum allowed softclipped bases of an aligned read is 30 a read pair with fragment length 300 will have $300 - 2(100 - 70) = 160$ possible placements where it spans position p. For simplicity, we omit the special case when p is near the end of a chromosome.

Modeling Variant at a Position. Let δ be the unknown variant size. In this case we cannot observe the true fragment length x of read pairs. What we observe is instead $o = x - \delta$ (see Fig. 1a). A modification of $w(x)$ is needed as fragment sizes is now required to span δ base pairs and have sufficiently many base pairs on each side to be mapped $(2(r-s))$. We have

$$w(x, \delta) = \frac{1}{G}\max\{x - \delta - 2(r-s) + 1, 0\}.$$

The 0 in the max function keeps the function weight to 0 in case we have no possible placing of a paired read over a variant. We can simplify this function to be expressed in o, as $o = x - \delta$, and write $w(o) = G^{-1}\max\{o - 2(r-s) + 1, 0\}$.

We see that the function w is constant for any given observation and can therefore be interpreted as a "weight" function, hence the notation w.

2.3 Probability of Variant Size δ

We can express the probability of δ given observations as $P(\delta|o)$. Lacking prior information about δ, we model it with the uniform distribution[4]. Using Bayes theorem, we get

$$P(\delta|o) = \frac{P(o|\delta)P(\delta)}{P(o)} \propto P(o|\delta)P(\delta) \propto P(o|\delta)$$

[4] A more informative prior could improve results, *e.g.*, by fitting to the expected frequency and length of variants, studied in [4,6]. By tailoring the prior we could essentially obtain any specificity and sensitivity for a given indel size. We believe that is promising future work.

where $P(o)$ and $P(\delta)$ are constant by the assumption of a uniform distribution. We now have

$$P(o|\delta) = \frac{w(o|\delta)f(\delta + o|\delta)}{\sum_t w(t-\delta)f(t)} \tag{2}$$

where the denominator is the sum of all possible fragment sizes that can be observed given δ and f. We can now find the most likely δ using maximum likelihood estimation (MLE) over (2). The time complexity for the MLE is $O(n+\log t)$[5] if $f \sim N$ (with t continuous), where n is the number of observations [22]. Note that we implicitly get $P(x|\delta)$ since $P(x|\delta) = P(o+\delta|\delta) = P(o|\delta)$.

2.4 Null-Hypothesis and Statistical Testing

Let $Y \doteq O|\delta = 0$, that is, the random variable over observed fragment lengths given $\delta = 0$. Let $\bar{y} \doteq \frac{\sum_{i=1}^{n} y_i}{n}$ be the sample mean of observed fragment lengths. Considering $\bar{y}$ a random variable over experiments, it is commonly assumed that $\bar{y} \sim N(\mu, \sigma/\sqrt{n})$, *i.e.*, the distribution of the sample mean of $f(x)$ under the central limit theorem (CLT), and this distribution is used as null-hypothesis [5,17]. We call this null-hypothesis H_0. Furthermore, the variant size δ is estimated from observed fragment lengths o as $\hat{\delta} = \mu - \bar{o}$ [5,9,10,12,17,20]. At first glance this formula seems reasonable since we take the expected fragment size and subtract the mean of the observations, but it has strong limitations. One is that $\hat{\delta}$ in this case has an upper bound of $\mu - 2(r-s)$ since $o \geq 2(r-s)$. This equation implies that we can never span over a sequence longer than $\mu - 2(r-s)$. We use Eq. 2 to derive the correct mean and standard deviation of Y given the OFL model, denoted μ_p and σ_p respectively. The derivation of μ_p is similar to derivation of observed fragment size linking two contigs given in [22], and the derivation of σ_p is a special case of the derivation of the variance of observed fragment size linking two contigs given in [23]. See proof in Appendix[6].

Theorem 1. *Given the OFL model, $f \sim N(\mu, \sigma)$, and $\delta = 0$, we have $\mu_p \approx \mu + \frac{\sigma^2}{\mu - (2(r-s)+1)}$ and $\sigma_p \approx \sigma\sqrt{1 - \sigma^2\left(\mu - (2(r-s)+1)\right)^{-2}}$.*

The null-hypothesis is that there is no variant, thus $\delta = 0$. Under CLT, as n increases, we therefore have $\bar{y} \sim N(\mu_p, \sigma_p/\sqrt{n})$. Notice that we can calculate μ_p and σ_p without the assumption $f \sim N(\mu, \sigma)$ by using an empirical estimate of $f(x)$ from aligned read pairs. Nevertheless, the closed expression formulas in Theorem 1 illustrates a basic feature of the model — larger variance increases the discrepancy between μ and μ_p. It is also robust to non-normality, as we will see in Sect. 3.2. In case we have enough observations to motivate the Z-test, we perform a simple Z-test and obtain a p-value based on a two sided test (both deletions and insertions are tested for) using the z-statistic

$$z = \frac{\bar{y} - \mu_p}{\sigma_p/\sqrt{n}}. \tag{3}$$

[5] n to obtain sample mean $\bar{o}$, and $\log t$ to search the convex ML curve.
[6] http://dx.doi.org/10.1101/023929, Sect. 5.5.

We refer to the null-hypothesis test using (3) as H_0'. Thus, we have derived a different distribution under the null-hypothesis which we advocate should be used instead of H_0. In case we have few observations (more often over insertions), approximation with the Z-test is poor. To get an exact test we would need to derive the distribution of $\sum_{i=1}^{n} Y_i$, for n observations y_i $i \in [1, n]$. This could improve power to detect insertions, but we refrain from studying this in the present paper.

3 Results

We discuss why modeling bias contributes to making deletion calls more frequent than insertions calls in Sect. 3.1. In Sect. 3.2 we show that our corrected hypothesis agrees with biological data, and in Sect. 3.3 that how indel detection is affected in CLEVER when our null-hypothesis is inserted.

3.1 Bias Between Detection of Deletions and Insertions

As donor fragments need to span over insertions ($\delta > 0$), and this probability is $w(x, \delta) = \frac{1}{G} \max\{x - \delta - 2(r - s) + 1, 0\}$ according to the OFL model, it is less likely that such fragments will be observed, as δ grows. We will therefore have a lower sample size over insertions in general. This naturally gives less power to detect an insertion compared to a deletion of similar size. However, methods using H_0 have less power than necessary. Firstly, as $\mu_p > \mu$, this gives too many significant upper quantile p-values (deletions) and too few significant lower quantile p-values (insertions). The difference in significance of observing a fragment of size $\mu + 2\sigma$ compared to observing a fragment of size $\mu - 2\sigma$ under H_0, compared to the when observed under H_0' is seen in Fig. 1b. Secondly, the positive skew of the OFL distribution (Fig. 1b) makes a Z-test approximation less powerful compared to an exact test, especially for small sample sizes — as is more likely for insertions.

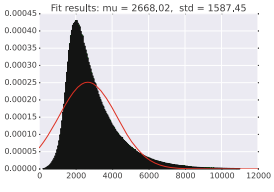
3.2 Evaluating the Accuracy of H_0'

We evaluated the accuracy of our null-hypothesis on three mate pair libraries. We used a mate pair library from *Rhodobacter sphaeroides* from [21] denoted *rhodo*, a mate pair library from *Plasmodium falciparum* used in [13] denoted *plasm*, and mate-pair data from a human individual in the CEPH 1463 family-trio[7]. For the human dataset we aligned the reads to the complete human genome, but limited analysis to chromosome 13. We call this dataset *hs13*. Table 1 shows information about the datasets. Recall from Sect. 2.4 that μ_p and σ_p are the true mean and standard deviation of fragment lengths over a position that does not contain a variant. Let $\hat{\mu}_p^c$ and $\hat{\sigma}_p^c$ refer to the estimated quantity of μ_p and σ_p from the closed formulas in Theorem 1. Similarly, let $\hat{\mu}_p^e$ and $\hat{\sigma}_p^e$ be the estimates of

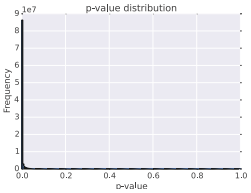
[7] http://www.ebi.ac.uk/ena/data/view/ERR262996.

Table 1. Library information. Reads were aligned with BWA-MEM [16] version 0.7.12 with default parameters. Physical coverage is c, for all reads, and c (pp), for restricted proper pairs, i.e., read pairs that have both mates mapped in correct orientation and within a distance that depends on a statistical filtering of outliers based on the library distribution. The filtering bounds were roughly 10000, 6000, and 14000 bp for rhodo, plasm, and hs13 respectively. μ and σ are the mean and standard deviation of the full fragment length distribution. True mean insert-size and standard deviation over a position on the genome, μ_p and σ_p (calculated as the average over all positions in the genome) and predictions with closed formula, $\hat{\mu}_p^c$ and $\hat{\sigma}_p^c$, and exact calculation, $\hat{\mu}_p^e$ and $\hat{\sigma}_p^e$.

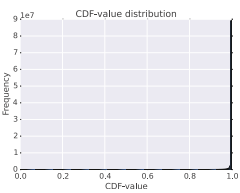
Organism	r	c	c (pp)	μ	σ	μ_p	σ_p	$\hat{\mu}_p^e$	$\hat{\sigma}_p^e$	$\hat{\mu}_p^c$	$\hat{\sigma}_p^c$
rhodo	101	43	34.5	2640	1390	3480	1534	3446	1526	3434	1143
plasm	75	4.9	4.2	2955	524	3056	511	3056	517	3053	515
hs13	80	11.1	9.0	2947	1454	3688	1780	3719	1806	3705	1241



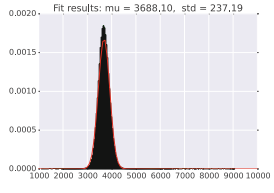
(a) $f(x)$



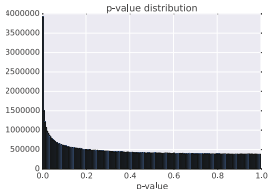
(b) p-values of H_0 based on μ, σ (naive) in equation 3.



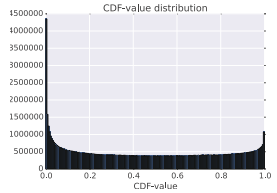
(c) CDF-values $P(X < x)$ of $f(x)$



(d) Distribution of $\mu_p, \forall p \in [1, G]$.



(e) p-values of H_0' based on $\hat{\mu}_p^e, \hat{\sigma}_p^e$ in equation 3



(f) CDF-values $P(X < x)$ of $w(x)f(x)$

Fig. 2. (a) The fragment length distribution $f(x)$ for the hs13 dataset and the red line is a best fit of a truncated normal distribution. $f(x)$ deviates significantly from a normal distribution. Although the mean of $f(x)$ is 2947, the average observed fragment length over position p (μ_p) over all positions on hs13 shows that most values occur between 3000–4500 bp with the average around 3688 bp (d) — as approximately predicted from $\hat{\mu}_p^e$ and $\hat{\mu}_p^c$. Figures (b) and (c) shows the p-value distribution and CDF values from using H_0 (*i.e.*, using μ and σ). Figures (e) and (f) shows the p-value distribution and CDF values from using H_0' (*i.e.*, using $\hat{\mu}_p^e$ and $\hat{\sigma}_p^e$).

μ_p and σ_p by using an empirical distribution of $f(x)$ (estimated from a sample) and summing up the probabilities in Eq. 2 with $\delta = 0$. Estimates and observed values are shown in Table 1. It is our assumption that an overwhelming majority of positions are variant free[8]. Thus, we expect a model that fits data should give a uniformly distributed p-value distribution. Our observations are summarized below.

Predicting μ_p. μ_p is estimated very well by both $\hat{\mu}_p^e$ and $\hat{\mu}_p^c$; compare Fig. 2d for hs13, and Appendix Figs. 3b and d for rhodo and plasm respectively with the estimated values in Table 1. Hence, testing $\bar{o} = \hat{\mu}_p^e$ (or $\hat{\mu}_p^c$) as in H_0' introduce symmetrical cumulative distribution function (CDF) values, Fig. 2f, compared to CDF based on testing $\bar{o} = \hat{\mu}$ where all values are distributed around 1.0 — suggesting significant deletions, see Fig. 2c.

Predicting σ_p. The closed formula predictions of σ_p works best if $f(x)$ is normal (plasm). For rhodo and hs13, $\hat{\sigma}_p^e$ and $\hat{\sigma}_p^c$ differs significantly and $\hat{\sigma}_p^e$ should be used, compare σ_p with $\hat{\sigma}_p^e$ and $\hat{\sigma}_p^c$ in Table 1.

p-values. The p-value distribution (ideally uniform) greatly improves with H_0' (Fig. 2e) compared to p-values obtained with H_0 (Fig. 2b). Abnormalities in the p-values are most likely explained by: alignment artifacts (some regions are more difficult aligning to), fragment length bias [1,19], coverage bias from GC-content, and in some cases, real variants, see evaluation of hs13 dataset in Sect. 5.7 of the Appendix. Similar p-value distributions are obtained on rhodo and plasm genome (data not shown) — that should not contain any variants — indicating that most of the enrichment of low p-values on hs13 is explained by any of the former three causes.

Table 2. Insertions and deletions called with CLEVER using H_0 and H_0'. Column δ contains the size of 50 insertions and deletions, simulated on the reference genomes by either deleting or inserting a δ bp sequence on the reference. A "0" indicates that the original biological dataset was used.

Dataset	δ	H_0				H_0'			
		TP (del/ins)		FP (del/ins)		TP (del/ins)		FP (del/ins)	
plasm	0	0	(0/0)	**20**	(6/14)	0	(0/0)	27	(6/21)
	2000	89	(50/39)	**22**	(8/22)	89	(50/39)	38	(8/30)
rhodo	0	0	(0/0)	78	(78/0)	0	(0/0)	**18**	(14/4)
	2000	49	(49/0)	54	(54/0)	**57**	(45/12)	**13**	(9/4)
sim-N(500,75)	75	**94**	(94/0)	0	(0/0)	78	(62/16)	0	(0/0)
	100	110	(100/10)	0	(0/0)	**188**	(100/88)	0	(0/0)
		ETP (hits)		TC (del/ins)		ETP (hits)		TC (del/ins)	
hs13	0	**9** (31)		1740	(1740/0)	3 (4)		**8**	(8/0)

[8] Even small variants $\delta \ll \sigma$ will not affect the model much.

3.3 Implementing the Corrected Null-Hypothesis in CLEVER

In this section we illustrate as a proof-of-concept how the corrected hypothesis H_0' (with $\hat{\mu}_p^e$ and $\hat{\sigma}_p^e$) balances the ratio between detected insertions and deletions. We applied H_0' in CLEVER (v 1.1). *However, we want to emphasize that we did not tailor the statistical tests as needed to fit the assumptions made by their particular method. This limits the performance improvement.* To further improve results with CLEVER, we would need to (1) implement exact tests for few observations — giving more power to detect insertions, (2) use the OFL-model for CLEVER's discovery of positions to study, (3) based on our model adjust CLEVER's methods to handle, *e.g.*, heterozygous variants and controlling the false discovery rate. This would require additional modeling and significant restructuring of the code and we do not consider it here. Our aim here is only to illustrate how the simple adjustment of inserting H_0' instead of H_0 in CLEVER has a significant impact on the output. We investigated how the replacement of H_0' instead of H_0 changed variant calls from CLEVER on hs13, rhodo and plasm as well with ideal condition simulated data denoted *sim-*$N(\cdot,\cdot)$ (full simulated results in Appendix Sect. 5.6). For simulated variants, similarly to [17], a prediction is classified as a true positive (TP) if the breakpoint prediction is not further than one mean insert size (*i.e.*, at most $\mu - 2r$) away from the true breakpoint. Otherwise it is classified as a false positive (FP). All variant calls on rhodo and plasm that are not from simulated variants are assumed to be false positives.

Because hs13 likely harbors true variants, we used annotated variants from dbVar [7], together with manual inspection in BamView [3], to assess if *hits* are true or false positives. For a deletion call in CLEVER with start and end coordinates p_s, p_e and a deletion in dbVar with coordinates q_s, q_e, we let $max\_del = \max(p_e - p_s, q_e - q_s)$ and $overlap = \min\{0, \min(p_e, q_e) - \max(p_s, q_s)\}$. We let $hit\_value = overlap/max\_del$ and a call is a *hit* if $hit\_value > T$, where $0 < T < 1$ is a threshold. Because dbVar contains a large amount of annotated variants from several individuals and CLEVER produces many calls under H_0, roughly 173, 106 and 40 hits are expected by chance with $T = 0.25, 0.5, 0.75$ (estimation in Appendix Sect. 5.3), which is similar numbers to the observed hits from CLEVER: 226, 109 and 31 respectively under $T = 0.25, 0.5, 0.75$. We therefore further manually evaluated the hits produced with $T = 0.75$ by looking for coverage drop and accumulation of softclips near each breakpoint. This gave us Estimated True Positives (ETP) as a rough measure of the TP rate for hs13. Therefore, we report ETP and Total Calls (TC) for hs13 in Table 2, contrary to simple TP and FP for the other data sets where we have the ground truth.

Improvements. From Table 2 and Fig. 5 (Appendix) we see that CLEVER with H_0 detects significantly more deletions than insertions of the same sizes. Using H_0', reduces this bias to some extent by increasing the detection of insertions across all data sets. CLEVER also returns significantly fewer false positive deletion calls with H_0', see rhodo Table 2 and sim-$N(300,\cdot)$. Even though H_0 have more sensitivity in calling deletions on hs13, the signal disappears in the

overwhelming amount of total calls, compare ETP and TC for H_0 and H_0' in Table 2.

Deterioration. A consequence of using H_0' is fewer deletion calls, which unfortunately also removes some true positive deletion calls (Table 2 and Appendix Fig. 5). It also increases the FP insertion calls on plasm (Table 2). We believe that calling variants with the plasm library carries additional difficulties due to its GC-poor genome sequence, such as positional fragment length bias [1,19].

Additional evidence that most calls with H_0 on hs13 are FPs are found by comparing statistics on CLEVER's deletion calls (Appendix Fig. 6) and numbers reported in recent extensive studies [4,6]. For example, [4] provide frequency distributions for both previously discovered and new deletions on single genomes. Roughly 250 deletions have lengths over 1000 bp (inspection of plot). The simplifying assumption that large-indel distribution is uniform over chromosomes gives around 8 expected deletions[9] in size range 1000 bp. This approximate number, and the fact that almost all calls were removed when using H_0' corroborates, that the vast majority (in the order of $> 99\,\%$) of calls with H_0 are FPs — likely a consequence of using $H_0 : \mu_p = 2410$ compared to the true value $\mu_p = 3719$.

4 Conclusions

We stated a probability distribution of observed fragment length over a position or region and derived a new null-hypothesis for detecting variants with fragment length, which is sound and agrees with biological data. Applied in CLEVER, our null-hypothesis detects more insertions and reduces false positive deletion calls. Results could be further improved by deriving an exact distribution instead of a Z-test and updating CLEVER's edge-creating conditions to agree with our model. The presented model, distribution, and null-hypothesis are general and could be used together with other information sources such as split reads, softclipped alignments, and read-depth information.

References

1. Benjamini, Y., Speed, T.P.: Summarizing and correcting the GC content bias in high-throughput sequencing. Nucleic Acids Res. **40**(10), e72–e72 (2012)
2. Bickhart, D., Hutchison, J., Xu, L., Schnabel, R., Taylor, J., Reecy, J., Schroeder, S., Van Tassell, C., Sonstegard, T., Liu, G.: RAPTR-SV: a hybrid method for the detection of structural variants. Bioinformatics **31**(13), 2084–2090 (2015)
3. Carver, T., Böhme, U., Otto, T.D., Parkhill, J., Berriman, M.: BamView: viewing mapped read alignment data in the context of the reference sequence. Bioinformatics **26**(5), 676–677 (2010)

[9] Estimated as $250 \cdot \frac{(114-16)\text{Mbp}}{3\ \text{Gbp}} = 8$, including compensation for the 16M N's at the start of the reference sequence for chr 13.

4. Chaisson, M.J.P., Huddleston, J., Dennis, M.Y., Sudmant, P.H., Malig, M., Hormozdiari, F., Antonacci, F., Surti, U., Sandstrom, R., Boitano, M., Landolin, J.M., Stamatoyannopoulos, J.A., Hunkapiller, M.W., Korlach, J., Eichler, E.E.: Resolving the complexity of the human genome using single-molecule sequencing. Nature **517**(7536), 608–611 (2015)
5. Chen, K., Wallis, J.W., McLellan, M.D., Larson, D.E., Kalicki, J.M., Pohl, C.S., McGrath, S., Wendl, M., Zhang, Q., Locke, D.P., Shi, X., Fulton, R.S., Ley, T., Wilson, R., Ding, L., Mardis, E.R.: BreakDancer: an algorithm for high-resolution mapping of genomic structural variation. Nat. Meth **6**(9), 677–681 (2009)
6. Chen, W., Zhang, L.: The pattern of DNA cleavage intensity around indels. Sci. Rep. **5**, 8333 (2015)
7. Church, D.M., Lappalainen, I., Sneddon, T.P., Hinton, J., Maquire, M., Lopez, J., Garner, J., Paschall, J., DiCuccio, M., Yaschenko, E., Scherer, S.W., Feuk, L., Flicek, P.: Public data archives for genomic structural variation. Nat. Genet. **42**(10), 813–814 (2010)
8. Fedorov, V., Mannino, F., Zhang, R.: Consequences of dichotomization. Pharm. Stat. **8**(1), 50–61 (2009)
9. Gillet-Markowska, A., Richard, H., Fischer, G., Lafontaine, I.: Ulysses: accurate detection of low-frequency structural variations in large insert-size sequencing libraries. Bioinformatics **31**(6), 801–808 (2015)
10. Handsaker, R.E., Korn, J.M., Nemesh, J., McCarroll, S.A.: Discovery and genotyping of genome structural polymorphism by sequencing on a population scale. Nat. Genet. **43**(3), 269–276 (2011)
11. Hayes, M., Pyon, Y.S., Li, J.: A model-based clustering method for genomic structural variant prediction and genotyping using paired-end sequencing data. PLoS ONE **7**(12), e52881 (2012)
12. Hormozdiari, F., Alkan, C., Eichler, E.E., Sahinalp, S.C.: Combinatorial algorithms for structural variation detection in high-throughput sequenced genomes. Genome Res. **19**(7), 1270–1278 (2009)
13. Hunt, M., Newbold, C., Berriman, M., Otto, T.: A comprehensive evaluation of assembly scaffolding tools. Genome Biol. **15**(3), R42 (2014)
14. Lander, E.S., Waterman, M.S.: Genomic mapping by fingerprinting random clones: a mathematical analysis. Genomics **2**(3), 231–239 (1988)
15. Lee, S., Hormozdiari, F., Alkan, C., Brudno, M.: MoDIL: detecting small indels from clone-end sequencing with mixtures of distributions. Nat. Methods **6**(7), 473–474 (2009)
16. Li, H., Durbin, R.: Fast and accurate long-read alignment with Burrows-Wheeler transform. Bioinformatics **26**(5), 589–595 (2010)
17. Marschall, T., Costa, I., Canzar, S., Bauer, M., Klau, G., Schliep, A., Schönhuth, A.: CLEVER: Clique-enumerating variant finder. Bioinformatics **28**(22), 2875–2880 (2012)
18. Nurk, S., Bankevich, A., Antipov, D., Gurevich, A.A., Korobeynikov, A., Lapidus, A., Prjibelski, A.D., Pyshkin, A., Sirotkin, A., Sirotkin, Y., Stepanauskas, R., Clingenpeel, S.R., Woyke, T., Mclean, J.S., Lasken, R., Tesler, G., Alekseyev, M.A., Pevzner, P.A.: Assembling single-cell genomes and mini-metagenomes from chimeric MDA products. J. Comput. Biol. **20**(10), 714–737 (2013)
19. Poptsova, M.S., Il'icheva, I.A., Nechipurenko, D.Y., Panchenko, L.A., Khodikov, M.V., Oparina, N.Y., Polozov, R.V., Nechipurenko, Y.D., Grokhovsky, S.L.: Non-random DNA fragmentation in next-generation sequencing. Sci. Rep. **4**, 4532 (2014)

20. Quinlan, A.R., Clark, R.A., Sokolova, S., Leibowitz, M.L., Zhang, Y., Hurles, M.E., Mell, J.C., Hall, I.M.: Genome-wide mapping and assembly of structural variant breakpoints in the mouse genome. Genome Res. **20**(5), 623–635 (2010)
21. Ribeiro, F.J., Przybylski, D., Yin, S., Sharpe, T., Gnerre, S., Abouelleil, A., Berlin, A.M., Montmayeur, A., Shea, T.P., Walker, B.J., Young, S.K., Russ, C., Nusbaum, C., MacCallum, I., Jaffe, D.B.: Finished bacterial genomes from shotgun sequence data. Genome Res. **22**(11), 2270–2277 (2012)
22. Sahlin, K., Street, N., Lundeberg, J., Arvestad, L.: Improved gap size estimation for scaffolding algorithms. Bioinformatics **28**(17), 2215–2222 (2012)
23. Sahlin, K., Vezzi, F., Nystedt, B., Lundeberg, J., Arvestad, L.: BESST – efficient scaffolding of large fragmented assemblies. BMC Bioinf. **15**(1), 281 (2014)

On Computing Breakpoint Distances for Genomes with Duplicate Genes

Mingfu Shao(✉) and Bernard M.E. Moret(✉)

Laboratory for Computational Biology and Bioinformatics,
EPFL, Lausanne, Switzerland
{mingfu.shao,bernard.moret}@epfl.ch

Abstract. A fundamental problem in comparative genomics is to compute the distance between two genomes in terms of its higher-level organization (given by genes or syntenic blocks). For two genomes without duplicate genes, we can easily define (and almost always efficiently compute) a variety of distance measures, but the problem is NP-hard under most models when genomes contain duplicate genes. To tackle duplicate genes, three formulations (exemplar, maximum matching, and any matching) have been proposed, all of which aim to build a matching between homologous genes so as to minimize some distance measure. Of the many distance measures, the breakpoint distance (the number of non-conserved adjacencies) was the first one to be studied and remains of significant interest because of its simplicity and model-free property.

The three breakpoint distance problems corresponding to the three formulations have been widely studied. Although we provided last year a solution for the exemplar problem that runs very fast on full genomes, computing optimal solutions for the other two problems has remained challenging. In this paper, we describe very fast, exact algorithms for these two problems. Our algorithms rely on a compact integer-linear program that we further simplify by developing an algorithm to remove variables, based on new results on the structure of adjacencies and matchings. Through extensive experiments using both simulations and biological datasets, we show that our algorithms run very fast (in seconds) on mammalian genomes and scale well beyond. We also apply these algorithms (as well as the classic orthology tool MSOAR) to create orthology assignment, then compare their quality in terms of both accuracy and coverage. We find that our algorithm for the "any matching" formulation significantly outperforms other methods in terms of accuracy while achieving nearly maximum coverage.

Keywords: Breakpoint distance · Exemplar · Intermediate · Maximum matching · Gene family · ILP · Orthology assignment

1 Introduction

The combinatorics and algorithmics of genomic rearrangements have been the subject of much research in comparative genomics since the problem was formulated in the 1990s (see, e.g., [1]). Perhaps the most fundamental problem is

M. Singh (Ed.): RECOMB 2016, LNBI 9649, pp. 189–203, 2016.
DOI: 10.1007/978-3-319-31957-5_14

the computation of some distance measure between two genomes. When the two genomes being compared have no duplicate genes (or synteny blocks, since the basic unit of description need not be restricted to genes), we have linear-time algorithms for most of these distance problems, such as the breakpoint distance, the inversion distance [2], and the DCJ distance [3,4].

However, gene duplications are widespread events and have long been recognized as a major driving force of evolution [5,6]. To define the distance in the presence of duplicate genes, three formulations have been proposed, all based on building a matching between duplicate genes and discarding copies not in the matching. This matching leads to treating each matched pair as a new gene family and thus removes duplicates, reducing the distance problem to its simplest version. In all formulations, the goal is to return that matching (from among all those obeying stated constraints) which minimizes the chosen distance. The first formulation is due to Sankoff [7], who proposed the *exemplar* model (E-model): select exactly one matched pair in each gene family. Several years later, Blin *et al.* [8] proposed the *maximum matching* model (M-model): use as many matched pairs as possible—until all genes in the genome with the smaller number of copies in each gene family are matched. Finally, Angibaud *et al.* [9] proposed what we shall term the *intermediate* model (I-model): the matching must contain at least one matched pair per gene family—so that the focus is clearly on minimizing the distance, not on meeting constraints on the matching. Figure 1 illustrates these three formulations. Unfortunately, for almost all distance measures, the corresponding distance problems under the three formulations are NP-hard [10].

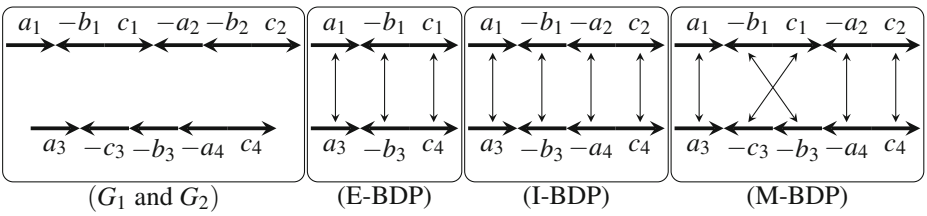


Fig. 1. Examples for E-BDP, I-BDP and M-BDP. Optimal matchings are shown within each model.

In this paper, we focus on the distance problems in the presence of duplicate genes under the breakpoint distance measure. We refer to the three corresponding problems under the three formulations as E-BDP (short for exemplar breakpoint distance problem), M-BDP, and I-BDP, respectively. Many algorithms have been proposed for these problems. For E-BDP, Sankoff gave a first branch-and-bound algorithm in his original paper [7]. Nguyen *et al.* gave a much faster divide-and-conquer approach [11], while Angibaud *et al.* [9] gave an exact algorithm by formulating it as an integer linear program (ILP). In [12], we gave an exact solution through combining ILP and novel preprocessing algorithms and

constraint-generating algorithms, which runs very fast (in a few seconds) and scales up to the largest genomes analyzed on both simulations and biological datasets. For M-BDP, Blin *et al.* [8] had defined (but not tested) a branch-and-bound algorithm, while Swenson *et al.* [13] gave an approximation algorithm. In [14], Angibaud *et al.* gave exact ILP formulations as well as several heuristics for both M-BDP and I-BDP—the heuristics being needed as the ILP formulations did not scale well. The ILP formulations in [14] were tested on 12 bacterial genomes, with numbers of genes varying from a few hundreds to three thousand; among the 66 pairwise comparisons, their ILP for I-BDP took 3 min on 63 of them, but could not terminate on the other 3 within a few hours. It thus remained a challenge to design fast and exact algorithms for I-BDP (and the simpler M-BDP) that could scale easily to mammalian genome sizes and beyond.

From the perspective of biologist, neither the E-model nor the M-model is satisfactory. In real gene families, we expect to find multiple orthologs, not just a single pair, so the E-model is an oversimplification and provides only a small amount of information, which in turn could lead to distortions in the pairwise distances. The M-model suffers from an even more fundamental problem: forcing all genes in the genome with the smaller number of copies to be matched effectively amounts to stating that every one of these genes has an ortholog in the other genome, something that clearly need not always be true. In terms of evolutionary events, this requirement also gives a much larger weight to duplications and losses of genes than to genomic rearrangements, thus implying that duplication and loss events are much less likely than rearrangement events. In other words, the M-model loses the model-free characteristic of the breakpoint distance. In contrast, the I-model retains the model-free property of the breakpoint distance while making full use of the information present in the gene families.

In this paper, we describe fast and exact algorithms for M-BDP and I-BDP. These algorithms employ ILP formulations and use a novel algorithm to reduce the number of key variables in the ILP, based on new results we prove about matchings and adjacencies. We evaluate these algorithms on simulated genomes and on several mammalian genomes. Our results show that our algorithms easily scale beyond the size of mammalian genomes: in all of our testing, almost all instances took a few seconds and none more that 70 s. They also demonstrate that our new algorithm to reduce the number of variables is a crucial contributor to this speed, especially for the more challenging I-BDP. Thus, with our previous (and equally fast) algorithm for E-BDP [12], we now have fast and exact algorithms to compute the breakpoint distance under all three formulations.

We also apply our algorithms to infer orthologs among five mammalian genomes and compare the quality of these assignments both among the three formulations and with the classic orthology tool MSOAR [15–17]. (We use MSOAR as it is one of the few orthology tools based on genome rearrangements and matching and because it has been extensively tested by its authors against other orthology tools.) The results demonstrate that the I-model substantially outperforms all other methods in terms of accuracy while giving very high coverage—close to the M-model (which defines the maximum possible number).

2 Problem Statement

We model each genome as a set of chromosomes and model each chromosome as a linear or circular list of genes. Each gene is represented by a signed (+ or −) symbol, where the sign indicates the transcriptional direction of this gene. Given a chromosome, we can reverse the list of symbols and switch all the signs, which will result in the same chromosome; for instance, $g_1 g_2 \cdots g_{n-1} g_n$ and $-g_n - g_{n-1} \cdots - g_2 - g_1$ represent the same linear chromosome. Homologous genes among the given genomes are grouped into *gene families*. In this paper, we assume that the given genomes have the same set of gene families, denoted by $\mathcal{F}$. For a gene family $f \in \mathcal{F}$ and a genome G, we use $F(G, f)$ to denote the set of genes in G that come from f. We say a gene family $f \in \mathcal{F}$ is a *singleton* in G if $|F(G, f)| = 1$; otherwise we say f is a *multi-gene family*.

Two consecutive genes g and h on the same chromosome, with g ahead of h along the chromosome, form an *adjacency*, written as gh. Given two genomes G_1 and G_2, we say two adjacencies $g_1h_1 \in G_1$ and $g_2h_2 \in G_2$ form a *pair of shared adjacencies* or a *PSA*, written as $\langle g_1h_1, g_2h_2 \rangle$, if g_1 and g_2 (and also h_1 and h_2) have the same sign and come from the same gene family, or g_1 and h_2 (and also h_1 and g_2) have opposite signs and come from the same gene family. If two given genomes G_1 and G_2 contain only singletons (have at most one gene each per gene family), then, for each adjacency in G_1 (resp. G_2), there exists at most one adjacency in G_2 (resp. G_1) shared with it.

Given two genomes G_1 and G_2 that may contain multi-gene families, we define a *matching* between them as a one-to-one correspondence between a subset of genes in G_1 and a subset of genes in G_2, such that each element of the matching is a pair of homologous genes. We denote by $\mathcal{M}$ the set of all possible matchings between G_1 and G_2. For a matching $M \in \mathcal{M}$ and a gene family $f \in \mathcal{F}$, we use $M(f)$ to denote the set of gene pairs in M that come from f. We say a gene is *covered* by M if it appears in some pair in M. Given $M \in \mathcal{M}$, we can modify G_1 and G_2 as follows: we first remove all genes that are not covered by M, then set up a new distinct gene family for each pair of genes in M. Clearly, these two new genomes contain only singletons; we denote by $\mathcal{S}(M)$ the set of PSAs between such two new genomes induced by M.

Given two genomes, the *breakpoint distance problem* is to compute a matching (satisfying some requirements depending on certain models) such that the number of shared adjacencies between the new genomes induced by this matching is maximized. Define the following sets of matchings:

$$\mathcal{M}_e = \{M \in \mathcal{M} \colon |M(f)| = 1, \forall f \in \mathcal{F}\};$$
$$\mathcal{M}_i = \{M \in \mathcal{M} \colon |M(f)| \geq 1, \forall f \in \mathcal{F}\};$$
$$\mathcal{M}_m = \{M \in \mathcal{M} \colon |M(f)| = \min\{|F(G_1, f)|, |F(G_2, f)|\}, \forall f \in \mathcal{F}\}.$$

Then the three problems corresponding to the three formulations can be written as:

$$\max_{M \in \mathcal{M}_e} |\mathcal{S}(M)| \quad \text{(E-BDP)}$$

$$\max_{M \in \mathcal{M}_i} |\mathcal{S}(M)| \quad \text{(I-BDP)}$$

$$\max_{M \in \mathcal{M}_m} |\mathcal{S}(M)| \quad \text{(M-BDP)}$$

Notice that our formulations are essentially based on *conserved adjacencies*, not breakpoints. For the E-model and the M-model, the two are equivalent; for the I-model, they are different (although clearly similar). This is due to the fact that the number of breakpoints equals the total number of adjacencies minus the number of conserved adjacencies, and that for the E-model and the M-model, the number of adjacencies is a fixed value, while for the I-model it is variant. Using conserved adjacencies rather than breakpoints has a significant advantage when the size of the matching is not fixed, as in the I-model the measure rewards both increased coverage and increased structural similarity.

3 Algorithms

We described a fast and exact algorithm for E-BDP in previous work [12]. In this section, we describe fast and exact algorithms for I-BDP and M-BDP. Both algorithms use the same framework, consisting of an integer linear program (ILP), described in Sect. 3.1, and an algorithm to reduce the number of variables in the ILP, described in Sect. 3.2. In each of them, we first describe the formulation or algorithm for I-BDP, then we state them for M-BDP, mainly focusing on clarifying their differences.

3.1 ILP Formulations

We first generalize the definition of adjacency. For two genes g and h on the same chromosome (g is ahead of h), we use $[g, h]$ to represent the genes from g to h along the chromosome (including g and h), and use (g, h) to represent the genes between g and h (excluding g and h). We say $[g, h]$ forms a *potential adjacency* for I-BDP, if we can remove all genes in (g, h) such that at least one gene remains in each gene family (as required by I-BDP). We say two potential adjacencies $[g_1, h_1] \in G_1$ and $[g_2, h_2] \in G_2$ form a *pair of shared potential adjacencies* (PSPA), written as $\langle [g_1, h_1], [g_2, h_2] \rangle$, if g_1 and g_2 (and also h_1 and h_2) have the same sign and come from the same gene family (case 1), or g_1 and h_2 (and also h_1 and g_2) have opposite signs and come from the same gene family (case 2). Without loss of generality, in the following we always assume that a PSPA belongs to case 1—all corresponding results for PSPAs belonging to case 2 can be derived directly in a symmetric way. We denote by $\mathcal{P}_i$ the set of all

PSPAs between G_1 and G_2 for I-BDP. Given a matching $M \in \mathcal{M}_i$, we say a PSPA $p = \langle [g_1, h_1], [g_2, h_2] \rangle \in \mathcal{P}_i$ *survives w.r.t.* M if we have $\langle g_1 h_1, g_2 h_2 \rangle \in \mathcal{S}(M)$.

Let $M_i^* \in \mathcal{M}_i$ be an optimal matching for I-BDP. Our ILP formulation to compute M_i^* has three types of variables. First, for every gene g in the two given genomes, we use one binary variable x_g to indicate whether g is covered by M_i^*. Second, for each pair of homologous genes $g_1 \in G_1$ and $g_2 \in G_2$, we use one binary variable y_{g_1,g_2} to indicate whether pair $\langle g_1, g_2 \rangle$ is in M_i^*. Third, for every PSPA $p \in \mathcal{P}_i$, we use one binary variable z_p to indicate whether p survives *w.r.t.* M_i^*.

Our ILP to compute M_i^* has three types of constraints. First, we require that for each gene family in each genome, at least one gene is covered by M_i^*:

$$\begin{aligned} \sum_{g \in F(G_1, f)} x_g \geq 1, \quad \forall f \in \mathcal{F}; \\ \sum_{g \in F(G_2, f)} x_g \geq 1, \quad \forall f \in \mathcal{F}. \end{aligned} \tag{1}$$

Second, we use the following constraints to guarantee that gene g is covered by M_i^* if and only if there exists a pair in M_i^* that includes g:

$$\begin{aligned} \sum_{g_2 \in F(G_2, f)} y_{g_1,g_2} = x_{g_1}, \quad \forall g_1 \in F(G_1, f), \forall f \in \mathcal{F}; \\ \sum_{g_1 \in F(G_1, f)} y_{g_1,g_2} = x_{g_2}, \quad \forall g_2 \in F(G_2, f), \forall f \in \mathcal{F}. \end{aligned}$$

Third, for each PSPA $p = \langle [g_1, h_1], [g_2, h_2] \rangle \in \mathcal{P}_i$, we require that, if p survives *w.r.t.* M_i^*, then we must have $\langle g_1, g_2 \rangle \in M_i^*$, $\langle h_1, h_2 \rangle \in M_i^*$, and that all genes in $(g_1, h_1) \cup (g_2, h_2)$ cannot be covered by M_i^*:

$$\begin{aligned} y_{g_1,g_2}, y_{h_1,h_2} \geq z_p, \quad \forall p = \langle [g_1, h_1], [g_2, h_2] \rangle \in \mathcal{P}_i; \\ 1 - z_p \geq x_g, \quad \forall g \in (g_1, h_1) \cup (g_2, h_2). \end{aligned} \tag{2}$$

The objective of the ILP is to maximize the sum of the variables for PSPAs:

$$\max \sum_{p \in \mathcal{P}_i} z_p.$$

A few modifications to this ILP for I-BDP provide an ILP for M-BDP. We say that $[g, h]$ is a potential adjacency for M-BDP if at least $\min\{|F(G_1, f)|, |F(G_2, f)|\}$ genes remain in gene family $f \in \mathcal{F}$ after genes in (g, h) are removed. Let $\mathcal{P}_m$ be the set of PSPAs for M-BDP and let $M_m^* \in \mathcal{P}_m$ be one optimal matching. We can construct the ILP to compute M_m^* from the above one to compute M_i^* by replacing M_i^* with M_m^*, $\mathcal{P}_i$ with $\mathcal{P}_m$, and constraints (1) with the following ones:

$$\begin{aligned} \sum_{g \in F(G_1, f)} x_g = \min\{|F(G_1, f)|, |F(G_2, f)|\}, \quad \forall f \in \mathcal{F}; \\ \sum_{g \in F(G_2, f)} x_g = \min\{|F(G_1, f)|, |F(G_2, f)|\}, \quad \forall f \in \mathcal{F}. \end{aligned}$$

Our ILP formulations use the same ideas we used for the exact algorithm to solve E-BDP in [12]. They are similar to those proposed in [14], but have fewer variables. In fact, the variables in our formulations are a subset of those in the formulations of [14]: the authors use two additional binary variables, c_{g_1,h_1} and c_{g_2,h_2}, for each PSPA $\langle [g_1,h_1],[g_2,h_2]\rangle$ to indicate whether g_1h_1 and g_2h_2 form adjacencies in the resulting genomes. In our formulations, the functions of these variables are carried out by constraints (2). ILP solvers typically do better with fewer variables and do not suffer (and often benefit) from the addition of extra constraints.

3.2 Building a Sufficient Subset

The number of binary variables corresponding to the PSPAs is the most important factor affecting the efficiency of our ILP formulations. We now describe an algorithm to reduce the number of these variables while preserving the optimality of the ILP. Formally, we say $\mathcal{P} \subset \mathcal{P}_i$ is *sufficient* if there exists an optimal matching $M_i^* \in \mathcal{M}_i$ such that for any PSA $\langle g_1h_1, g_2h_2\rangle \in \mathcal{S}(M_i^*)$, its corresponding PSPA $\langle [g_1,h_1],[g_2,h_2]\rangle$ is in $\mathcal{P}$. Clearly, if we replace $\mathcal{P}_i$ by a sufficient subset $\mathcal{P}$ in the ILP formulation described in Sect. 3.1, an optimal matching can be still obtained. In the following we first prove two conditions to reduce a single PSPA, then use these results to devise an iterative algorithm to compute a sufficient subset of reduced cardinality.

Intuitively, the first lemma states that we can remove a PSPA without breaking sufficiency if it can be split into two PSPAs.

Lemma 1. *Let $\mathcal{P} \subset \mathcal{P}_i$ be a sufficient subset. Let $p = \langle [g_1,h_1],[g_2,h_2]\rangle \in \mathcal{P}$. If there exists gene $x_1 \in (g_1,h_1)$ and gene $x_2 \in (g_2,h_2)$ that have the same sign and come from the same gene family, then we have that $\mathcal{P}\backslash\{p\}$ is sufficient.*

Proof. We prove this lemma by contradiction. Suppose that $\mathcal{P}\backslash\{p\}$ is not sufficient. Let M_i^* be any optimal matching. Since $\mathcal{P}$ is sufficient but $\mathcal{P}\backslash\{p\}$ is not, p survives *w.r.t.* M_i^*, i.e., we have $\langle g_1h_1, g_2h_2\rangle \in \mathcal{S}(M_i^*)$. Let $M_i' = M_i^* \cup \{\langle x_1,x_2\rangle\}$. Clearly, we have that $\mathcal{S}(M_i') = \mathcal{S}(M_i^*)\backslash\{\langle g_1h_1, g_2h_2\rangle\} \cup \{\langle g_1x_1, g_2x_2\rangle, \langle x_1h_1, x_2h_2\rangle\}$. Thus we have that $|\mathcal{S}(M_i')| = |\mathcal{S}(M_i^*)| + 1$, contradicting with the assumption that M_i^* is optimal. □

We now give another condition to reduce the size of a sufficient subset. Intuitively, the following lemma states that we can remove a PSPA without breaking sufficiency if there exists a gene inside that can play the same role as one of its four boundary genes.

Lemma 2. *Let $\mathcal{P} \subset \mathcal{P}_i$ be a sufficient subset. Set $p = \langle [g_1,h_1],[g_2,h_2]\rangle \in \mathcal{P}$ and $X = \{\langle x_1,x_2\rangle : \langle [h_1,x_1],[h_2,x_2]\rangle \in \mathcal{P}\}$. If there exists $h \in (g_1,h_1)$ satisfying $\langle [g_1,h],[g_2,h_2]\rangle \in \mathcal{P}$, such that for all $\langle x_1,x_2\rangle \in X$, we have $\langle [h,x_1],[h_2,x_2]\rangle \in \mathcal{P}$, then $\mathcal{P}\backslash\{p\}$ is sufficient.*

Proof. Let M_i^* be an optimal matching. If p does not survive *w.r.t.* M_i^*, then we can conclude that $\mathcal{P}\backslash\{p\}$ is sufficient. Now consider the case where p survives

w.r.t. M_i^*, i.e., $\langle g_1h_1, g_2h_2\rangle \in \mathcal{S}(M_i^*)$. Set $M_i' = M_i^* \setminus \{\langle h_1, h_2\rangle\} \cup \{\langle h, h_2\rangle\}$. Consider the difference between $\mathcal{S}(M_i^*)$ and $\mathcal{S}(M_i')$. Clearly, at most two PSAs in $\mathcal{S}(M_i^*)$ can be affected by the removal of $\langle h_1, h_2\rangle$. One of them is $\langle g_1h_1, g_2h_2\rangle$, while we have $\langle g_1h, g_2h_2\rangle \in \mathcal{S}(M_i')$ in return. If we further have $\langle h_1x_1, h_2x_2\rangle \in \mathcal{S}(M_i^*)$ for some $\langle x_1, x_2\rangle$, we must also have $\langle hx_1, h_2x_2\rangle \in \mathcal{S}(M_i')$ in return. Thus, we have $|\mathcal{S}(M_i')| = |\mathcal{S}(M_i^*)|$, which implies that M_i' is also optimal. On the other hand, we can show that all PSAs in $\mathcal{S}(M_i')$ have their corresponding PSPAs in $\mathcal{P}\backslash\{p\}$. In fact, p does not survive *w.r.t.* M_i', and we have $\langle [g_1, h], [g_2, h_2]\rangle \in \mathcal{P}$ and $\langle [h, x_1], [h_2, x_2]\rangle \in \mathcal{P}$; finally, the validity of all other PSAs in $\mathcal{S}(M_i')$ is guaranteed by the sufficiency of $\mathcal{P}$. Thus, in addition to the optimality of M_i', we have that $\mathcal{P}\backslash\{p\}$ is sufficient. □

Given a PSPA $\langle [g_1, h_1], [g_2, h_2]\rangle$, Lemma 2 only proves the condition related to h_1. We can derive the other three conditions for g_1, g_2 and h_2 analogously. If a PSPA passes any of these four conditions, it can be removed while keeping sufficiency.

Based on the above two lemmas, our algorithm to build a sufficient subset (BSS) of reduced cardinality proceeds as follows. The algorithm initializes the current set of PSPAs $\mathcal{P}$ as $\mathcal{P}_i$, and then it iteratively removes redundant PSPAs in $\mathcal{P}$. In each iteration, the algorithm examines each PSPA $p \in \mathcal{P}$ using the above two lemmas: if p passes either of the tests, then we update $\mathcal{P}$ as $\mathcal{P}\backslash\{p\}$ and starts a new iteration. The algorithm terminates if no PSPA passes any tests in an iteration.

We give two examples to illustrate the effect of the BSS algorithm. First, suppose that two genomes contain a pair of nontrivial ($n > 1$) shared segments $\langle g^1g^2\cdots g^n, h^1h^2\cdots h^n\rangle$, i.e., g^k and h^k have the same sign and come from the same gene family, for $1 \le k \le n$. Then there are $n \cdot (n-1)/2$ PSPAs generated from this shared segment, namely, $\langle [g^i, g^j], [h^i, h^j]\rangle$ for all $1 \le i < j \le n$. After running BSS on these PSPAs, we can get a sufficient subset with only $(n-1)$ PSPAs, namely, $\langle [g^k, g^{k+1}], [h^k, h^{k+1}]\rangle$, $1 \le k \le n-1$. The removal of PSPAs is due here to Lemma 1—Lemma 1 is very effective when genomes contain duplicate segments. Secondly, suppose that G_1 contains a segment $a_1b_1c_1$ and G_2 contains a segment $a_2b_2b_3\cdots b_nc_2$. Here $2 \cdot (n-1)$ PSPAs can be generated from these two segments, namely, $\langle [a_1, b_1], [a_2, b_k]\rangle$ and $\langle [b_1, c_1], [b_k, c_2]\rangle$, $2 \le k \le n$. After applying BSS on these PSPAs, we can see that a sufficient subset with at most 4 PSPAs is returned, namely, $\{\langle [a_1, b_1], [a_2, b_2]\rangle, \langle [b_1, c_1], [b_n, c_2]\rangle\} \cup \{\langle [a_1, b_1], [a_2, b_k]\rangle, \langle [b_1, c_1], [b_k, c_2]\rangle\}$ for some k, $2 \le k \le n$, where the value of k depends on the order in which these PSPAs are tested in BSS. The removal of PSPAs here is due to Lemma 2—Lemma 2 is very effective when genomes contain locally duplicate genes.

Since M-BDP requires keeping the maximum number of pairs, we have that a PSPA $\langle [g_1, h_1], [g_2, h_2]\rangle \in \mathcal{P}_m$ cannot have a pair of genes $x_1 \in (g_1, h_1)$ and $x_2 \in (g_2, h_2)$ within the same gene family. In other words, the PSPAs that are tested for removal in Lemma 1 do not appear in $\mathcal{P}_m$ by definition. In fact, it is clear that $\mathcal{P}_m$ is a (usually very small) subset of $\mathcal{P}_i$, making M-BDP significantly easier to solve than I-BDP (as is also apparent from the results of Sects. 4 and 5).

Lemma 2 holds unchanged for M-BDP. Thus the BSS algorithm for M-BDP uses only Lemma 2.

4 Simulation Results

We refer to the full algorithms (ILP plus BSS) for I-BDP and M-BDP as I-A1 and M-A1, respectively, and refer to the baseline algorithms (just the ILP) for I-BDP and M-BDP as I-A0 and M-A0, respectively. We use simulated data to evaluate these four algorithms in order to illustrate both the performance of the full algorithms and the effectiveness of the BSS reduction algorithm. We do not explicitly compare with the algorithms proposed in [14] for I-BDP and M-BDP, because their performance is similar to and bounded by that of I-A0 and M-A0 respectively, as discussed in Sect. 3.1.

We simulate a pair of genomes as follows. We start from a genome with only one linear chromosome consisting of N singletons. We then perform S_1 segmental duplications to make some multi-gene families, which then forms the ancestor genome. A segmental duplication randomly chooses a segment of length L and inserts its copy to another random position. The two extant genomes then speciate independently from this ancestor genome. After that on each branch it happens randomly mixed I inversions and S_2 segmental duplications. An inversion randomly chooses two positions in the genome and then reverses the segment in between. We make sure that the expected number of genes per gene family in each extant genome is 2 (the average copy number of each gene family in human, mouse and rat genomes, are 1.46, 1.55 and 1.28, respectively; thus in terms of duplicated genes, our simulated genomes are more complicated than typical mammalian genomes). Therefore, we have that $(S_1 + S_2) \cdot L = N$. Let $r_1 = S_1/(S_1 + S_2)$ be the percentage of segmental duplications before speciation for each genome. Let $r_2 = I/(I + S_2)$ be the percentage of inversions after speciation. We can calculate that $S_1 = r_1 \cdot N/L$, $S_2 = (1 - r_1) \cdot N/L$, and $I = r_2 \cdot (1-r_1) \cdot N/(L \cdot (1-r_2))$. Thus, a simulation configuration is determined by parameters (N, L, r_1, r_2). For each parameter combination, we randomly simulate 10 independent instances and run the four algorithms (I-A1, I-A0, M-A1 and M-A0) to calculate the average running time over these 10 instances. Since ILP might take very long time, we give a time limit of 30 min for each instance—an algorithm will be terminated when it exceeds such time limit.

In the following experiments, all our ILP instances are solved with GUROBI [18]. We first test parameters $(N = 10000, L, r_1 = 0, r_2)$, where $r_2 \in \{0.0, 0.1, \cdots, 0.9\}$ and $L \in \{1, 5, 10\}$. In this setting, all segmental duplications appear after speciation ($r_1 = 0$), and the expected number of genes in each extant genome is 20000. The results on these parameters are shown in Table 1. First, we can observe that as L increases, all algorithms take more time, indicating that the simulated instances become harder with larger L. This is because longer shared segments create drastically more PSPAs (in order of $O(n^2)$, where n is the length of this shared segments). Second, all the four algorithms take less time as r_2 increases. This is because larger r_2 means more inversions after

Table 1. Comparison of the four algorithms on parameters $(N = 10000, L, r_1 = 0, r_2)$. For each combination, if all 10 instances finish in 30 min, the average running time (in seconds) is recorded; otherwise, the number of finished instances is recorded in parentheses. All four programs were run on an 8-core (2.1GHz) PC with 16GB memory.

r_2	$L = 1$				$L = 5$				$L = 10$			
	I-A1	I-A0	M-A1	M-A0	I-A1	I-A0	M-A1	M-A0	I-A1	I-A0	M-A1	M-A0
0.0	1	2	0	0	25	(0)	8	8	55	(0)	10	12
0.1	1	1	0	0	21	(0)	7	9	43	(0)	10	13
0.2	0	0	0	0	14	(0)	7	7	33	(0)	9	11
0.3	0	0	0	0	11	(0)	5	4	26	(0)	8	8
0.4	0	0	0	0	6	554	3	3	22	(0)	7	8
0.5	0	0	0	0	4	123	2	2	15	(0)	6	5
0.6	0	0	0	0	2	38	1	1	10	(0)	4	4
0.7	0	0	0	0	1	3	1	0	7	430	3	2
0.8	0	0	0	0	0	0	0	0	4	70	1	1
0.9	0	0	0	0	0	0	0	0	0	0	0	0

speciation, which can destroy existing PSPAs. Third, we can see I-A1 can finish in less than 1 min for all parameters, while I-A0 will exceed time limit for large L and small r_2—this proves that the BSS algorithm is very crucial for I-BDP. Fourth, we can see both M-A1 and M-A0 can finish in a very short time for all parameters, indicating that the ILP formulation for M-BDP is already very efficient. This is because the definition of M-BDP determines that the number of PSPAs is reasonably small, as we analyzed theoretically in Sect. 3.2. For the same reason, we can also observe that M-BDP is easier to solve than I-BDP.

We then test parameters $(N = 20000, L, r_1 = 0.5, r_2)$, where $r_2 \in \{0.0, 0.1, \cdots, 0.9\}$ and $L \in \{1, 5, 10\}$. In this setting, the expected number of genes in the ancestor genome is 30000, and the expected number of genes in each extant genome is 40000. The results are shown in Table 2. Again, we can observe that the instances become harder to solve as L increases and r_2 decreases. Also observe that M-A1 and M-A0 takes similar (and very small) amount of time, since the ILP formulation for M-BDP is already very efficient. We emphasize that I-A0 can only finish within the time limit for small L and large r_2, while I-A1 can get the optimal solution for all parameters very fast, showing that the BSS algorithm plays a key role in producing a fast algorithm for I-BDP. Notice that in these simulations, the size of extant genomes exceeds that of typical mammalian genomes and our full algorithms (I-A1 and M-A1) return optimal solutions in a very short time.

Table 2. Comparison of the four algorithms on parameters ($N = 20000, L, r_1 = 0.5, r_2$). The setup is the same as in Table 1.

r_2	$L = 1$				$L = 5$				$L = 10$			
	I-A1	I-A0	M-A1	M-A0	I-A1	I-A0	M-A1	M-A0	I-A1	I-A0	M-A1	M-A0
0.0	6	(0)	2	2	48	(0)	22	25	104	(0)	44	36
0.1	8	(0)	2	2	45	(0)	26	22	93	(0)	39	39
0.2	5	521	1	1	42	(0)	22	24	89	(0)	38	37
0.3	6	200	1	1	37	(0)	20	19	83	(0)	36	32
0.4	8	28	1	1	31	(0)	18	17	65	(0)	31	29
0.5	4	3	2	1	29	(0)	16	17	57	(0)	35	31
0.6	2	1	1	1	25	(0)	13	14	52	(0)	29	26
0.7	1	1	1	0	18	(0)	11	15	42	(0)	27	22
0.8	1	1	1	0	8	336	5	7	27	(0)	18	15
0.9	1	1	2	0	2	3	1	1	11	437	8	5

5 Biological Results

We study five well-annotated genomes, human (*H.s.*), gorilla (*G.g.*), orangutan (*P.a.*), mouse (*M.m.*), and rat (*R.n.*). For each species, we collect all the protein-coding genes and download their positions on the chromosomes and their Ensembl gene family names from Ensembl (http://www.ensembl.org). Genes are grouped into the same gene family if they have the same Ensembl gene family name. For each species, we merge each group of tandemly arrayed genes into a single gene by keeping only the first gene and discarding the following ones in the group.

We perform pairwise comparisons among these five species. For each pair of species, we compare the running time for the four algorithms (I-A1, I-A0, M-A1 and M-A0). We also run another two algorithms for comparison, namely, the exact algorithm to solve E-BDP described in [12] (referred to as E-A1), and MSOAR [16], which uses heuristics to compute a matching such that the inversion distance induced by this matching is minimized. The results are shown in Table 3. Note that I-A0 cannot finish within the time limit for any pair, while I-A1 can finish in a very short time (less than 30 s except one taking 72 s) for all pairs, once again indicating that the BSS algorithm is indispensable for solving I-BDP. The BSS algorithm improves the solution for M-BDP as well, allowing M-A1 to finish within 2 s for all pairs. (MSOAR takes quite long time for these pairs, ranging from half an hour to a few hours.) Thus we now have exact and very fast algorithms (E-A1, I-A1 and M-A1) for all three formulations.

All of these algorithms give a matching between the homologous genes for each pair of species, a matching that defines a subset of orthologs under a parsimonious evolutionary assumption. We thus apply our full algorithms (E-A1, I-A1 and M-A1) together with MSOAR to infer orthologs among these 5 species.

Table 3. The running time (in seconds) of the algorithms on comparing five genomes. I-A0 cannot finish in 30 min for any species pair.

Species pairs	I-A1	I-A0	M-A1	M-A0	E-A1	MSOAR
G.g.&H.s.	8	N/A	1	1	2	1770
G.g.&M.m.	20	N/A	1	59	3	5298
G.g.&P.a.	16	N/A	1	1	2	3191
G.g.&R.n.	22	N/A	1	4	3	12555
H.s.&M.m.	23	N/A	1	2	2	3660
H.s.&P.a.	9	N/A	1	2	2	1585
H.s.&R.n.	22	N/A	1	6	2	7328
M.m.&P.a.	18	N/A	1	1	2	4287
M.m.&R.n.	72	N/A	2	66	2	5627
R.n. &P.a.	18	N/A	1	2	3	6009

The quality of the inferred orthologs is evaluated using two measures, the coverage (the number of orthologous pairs identified) and the accuracy. To compute the accuracy of a matching, we use the gene symbols (HGNC symbols for primate genes, MGI symbols for mouse genes, and RGD symbols for rat genes, downloaded from Ensembl): those gene pairs that have the same gene symbol form the set of *true orthology pairs*. We say a pair in a matching is *trivial* if it consists of two singletons. Notice that by definition all trivial pairs must appear in the matchings returned by these four algorithms. We thus exclude trivial pairs for comparison. Among the non-trivial pairs in a matching, we say a pair is *assessable* if at least one gene in this pair is covered by some true orthology pair. Then the *accuracy* of a matching is defined as the ratio between the number of non-trivial true orthology pairs in this matching and the number of (non-trivial) assessable pairs in this matching.

The quality of the orthologs inferred by these four algorithms is shown in Table 4. First, we can observe that the coverage of I-A1 (by definition, it is between E-A1 and M-A1) is much closer to M-A1 (much higher than E-A1). Second, notice that the accuracy of I-A1 significantly outperforms the other three algorithms—it is 1.08 % higher than E-A1, 0.89 % higher than M-A1, and 1 % higher than MSOAR, on average over the 10 pairs. Third, the quality of MSOAR is very close to M-A1, in terms of both the coverage and accuracy (the accuracy of M-A1 is 0.11 % higher than that of MSOAR on average). Thus, we believe that I-A1 is an excellent choice for inferring orthologs, which outperforms in terms of both coverage and accuracy.

We do not evaluate the accuracy of these algorithms on simulation data. The reason is that orthology assignment requires a biologically credible model for generating simulation data—a model that need combine duplications, losses, rearrangements, and sequence mutations and indels. Such a model that takes into account all these evolutionary events in a biologically reasonable way is

Table 4. The performance of the algorithms on inferring orthologs among the five genomes.

Species pairs	Trivial	Non-trivial				Accuracy (%)			
		E-A1	I-A1	M-A1	MSOAR	E-A1	I-A1	M-A1	MSOAR
G.g. &H.s.	8331	3448	7830	8131	8051	97.51	98.18	97.56	97.60
G.g. &M.m.	7304	3478	7572	8025	7858	97.33	98.29	97.61	97.48
G.g. &P.a.	7737	3399	7466	7893	7720	97.26	98.10	97.37	97.17
G.g. &R.n.	6915	3787	7826	8610	8317	94.65	96.12	94.56	94.43
H.s. &M.m.	7932	3250	7546	7834	7722	97.89	98.71	98.16	98.20
H.s. &P.a.	8091	3223	7355	7585	7501	98.27	98.73	98.03	98.05
H.s. &R.n.	7436	3638	7808	8196	8072	94.64	96.10	94.94	94.93
M.m. &P.a.	7311	3276	7208	7537	7408	97.91	98.61	97.95	97.81
M.m. &R.n.	7953	3706	8376	8837	8671	94.82	96.52	95.63	95.20
R.n. &P.a.	6911	3610	7466	7987	7789	94.90	96.63	95.27	95.08

currently not available. Besides, we have high-quality fully assembled genomes with curated annotations to assess these methods, which leaves little necessity to perform comparison on simulation data.

6 Conclusion and Discussion

We have described exact and very fast algorithms for the three breakpoint distance problems, by formulating them as integer linear programs and designing additional algorithms to improve their efficiency through proving key properties of these problems. Using extensive experiments on both simulations and biological datasets, we have demonstrated that these algorithms scale beyond the size of mammalian genomes, and also achieve very high accuracy when applied to infer orthologs. We conclude that among those orthology assignment tools, I-A1 is the most suitable choice, since it gives highest accuracy and nearly highest coverage.

We also help understanding the structures of these problems through proving several properties. As we have already illustrated, these properties are crucial in designing efficient algorithms for the corresponding problems. Notice that some properties are common among the three problems, while some others only hold for one or two of them. In Sect. 3.2, we give some theoretical analysis between I-BDP and M-BDP through showing that Lemma 1 only applies for I-MDP, while Lemma 2 holds for both of them. Now we further state the relationship between E-BDP and I-BDP/M-BDP. We can easily prove that both lemmas in this paper also apply to E-BDP. On the other side, in [12], we proved a lemma for E-BDP stating that if one pair in a PSA is in an optimal matching, then the other pair can be also fixed optimally. However, this property does not hold for I-BDP and M-BDP. To see that, consider the example of $G_1 = a_1b_1c_1b_2d_1$

and $G_2 = a_2b_3d_2b_4c_2$. Notice that $\langle a_1, a_2\rangle$ is a singleton pair and thus must be in any optimal matching. If we apply this property, then $\langle b_1, b_3\rangle$ is also in some optimal matching. However, we can easily verify that for both I-BDP and M-BDP, $\langle b_1, b_3\rangle$ is not in any optimal matching.

Acknowledgements. We thank Daniel Dörr for helpful discussions.

References

1. Fertin, G., Labarre, A., Rusu, I., Tannier, E., Vialette, S.: Combinatorics of Genome Rearrangements. MIT Press, Cambridge (2009)
2. Bader, D.A., Moret, B.M.E., Yan, M.: A fast linear-time algorithm for inversion distance with an experimental comparison. J. Comput. Biol. **8**(5), 483–491 (2001)
3. Yancopoulos, S., Attie, O., Friedberg, R.: Efficient sorting of genomic permutations by translocation, inversion and block interchange. Bioinformatics **21**(16), 3340–3346 (2005)
4. Bergeron, A., Mixtacki, J., Stoye, J.: A unifying view of genome rearrangements. In: Bücher, P., Moret, B.M.E. (eds.) WABI 2006. LNCS (LNBI), vol. 4175, pp. 163–173. Springer, Heidelberg (2006)
5. Bailey, J.A., Eichler, E.E.: Primate segmental duplications: crucibles of evolution, diversity and disease. Nat. Rev. Genet. **7**(7), 552–564 (2006)
6. Lynch, M.: The Origins of Genome Architecture. Sinauer, Sunderland (2007)
7. Sankoff, D.: Genome rearrangement with gene families. Bioinformatics **15**(11), 909–917 (1999)
8. Blin, G., Chauve, C., Fertin, G.: The breakpoint distance for signed sequences. In: Proceedings of the 1st Conference on Algorithms and Computational Methods for Biochemical and Evolutionary Networks (CompBioNets 2004), vol. 3, pp. 3–16 (2004)
9. Angibaud, S., Fertin, G., Rusu, I., Thévenin, A., Vialette, S.: A pseudo-boolean programming approach for computing the breakpoint distance between two genomes with duplicate genes. In: Tesler, G., Durand, D. (eds.) RECMOB-CG 2007. LNCS (LNBI), vol. 4751, pp. 16–29. Springer, Heidelberg (2007)
10. Blin, G., Chauve, C., Fertin, G., Rizzi, R., Vialette, S.: Comparing genomes with duplications: a computational complexity point of view. ACM/IEEE Trans. Comput. Bio. Bioinf. **14**, 523–534 (2007)
11. Nguyen, C.T., Tay, Y.C., Zhang, L.: Divide-and-conquer approach for the exemplar breakpoint distance. Bioinformatics **21**(10), 2171–2176 (2005)
12. Shao, M., Moret, B.M.E.: A fast and exact algorithm for the exemplar breakpoint distance. In: Przytycka, T.M. (ed.) RECOMB 2015. LNCS, vol. 9029, pp. 309–322. Springer, Heidelberg (2015)
13. Swenson, K.M., Marron, M., Earnest-DeYoung, J.V., Moret, B.M.E.: Approximating the true evolutionary distance between genomes. In: Proceedings of the 7th SIAM Workshop on Algorithm Engineering and Experiments (ALENEX 2005), pp. 121–129. SIAM Press (2005)
14. Angibaud, S., Fertin, G., Rusu, I., Thévenin, A., Vialette, S.: Efficient tools for computing the number of breakpoints and the number of adjacencies between two genomes with duplicate genes. J. Comput. Biol. **15**(8), 1093–1115 (2008)

15. Chen, X., Zheng, J., Fu, Z., Nan, P., Zhong, Y., Lonardi, S., Jiang, T.: Assignment of orthologous genes via genome rearrangement. ACM/IEEE Trans. Comput. Bio. Bioinf. **2**(4), 302–315 (2005)
16. Fu, Z., Chen, X., Vacic, V., Nan, P., Zhong, Y., Jiang, T.: MSOAR: a high-throughput ortholog assignment system based on genome rearrangement. J. Comput. Biol. **14**(9), 1160–1175 (2007)
17. Shi, G., Zhang, L., Jiang, T.: MSOAR 2.0: incorporating tandem duplications into ortholog assignment based on genome rearrangement. BMC Bioinform. **11**(1), 10 (2010)
18. Gurobi Optimization Inc.: Gurobi optimizer reference manual (2013)

New Genome Similarity Measures Based on Conserved Gene Adjacencies

Luis Antonio B. Kowada[1], Daniel Doerr[2], Simone Dantas[1], and Jens Stoye[1,3(✉)]

[1] Universidade Federal Fluminense, Niterói, Brazil
[2] École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
[3] Faculty of Technology, Center for Biotechnology, Bielefeld University, Bielefeld, Germany
jens.stoye@uni-bielefeld.de

Abstract. Many important questions in molecular biology, evolution and biomedicine can be addressed by comparative genomics approaches. One of the basic tasks when comparing genomes is the definition of measures of similarity (or dissimilarity) between two genomes, for example to elucidate the phylogenetic relationships between species.

The power of different genome comparison methods varies with the underlying formal model of a genome. The simplest models impose the strong restriction that each genome under study must contain the same genes, each in exactly one copy. More realistic models allow several copies of a gene in a genome. One speaks of gene families, and comparative genomics methods that allow this kind of input are called *gene family-based*. The most powerful – but also most complex – models avoid this preprocessing of the input data and instead integrate the family assignment within the comparative analysis. Such methods are called *gene family-free*.

In this paper, we study an intermediate approach between family-based and family-free genomic similarity measures. The model, called *gene connections*, is on the one hand more flexible than the family-based model, on the other hand the resulting data structure is less complex than in the family-free approach. This intermediate status allows us to achieve results comparable to those for family-free methods, but at running times similar to those for the family-based approach.

Within the gene connection model, we define three variants of genomic similarity measures that have different expression power. We give polynomial-time algorithms for two of them, while we show NP-hardness of the third, most powerful one. We also generalize the measures and algorithms to make them more robust against recent local disruptions in gene order. Our theoretical findings are supported by experimental results, proving the applicability and performance of our newly defined similarity measures.

1 Introduction

Many important questions in molecular biology, evolution and biomedicine can be addressed by comparative genomics approaches. One of the basic tasks in

M. Singh (Ed.): RECOMB 2016, LNBI 9649, pp. 204–224, 2016.
DOI: 10.1007/978-3-319-31957-5_15

this area is the definition of measures of similarity between two genomes. Direct applications of such measures are the computation of phylogenetic trees or the reconstruction of ancestral genomes, but also more indirect tasks like the prediction of orthologous gene pairs (derived from the same ancestor gene through speciation) or the transfer of gene function across species profit immensely from accurate genome comparison methods.

Indeed, over the past forty-or-so years, many methods have been proposed to quantify the similarity of single genes, mostly based on pairwise or multiple sequence alignments. However, in many situations similarity measures based on whole genomes are more meaningful than gene-based measures, because they give a more representative picture and are more robust against side effects such as horizontal gene transfer. Therefore, in this paper we develop and analyze methods for whole genome comparison, based on the physical structure (gene order) of the genomes.

The most simple picture of a genome is one where in a set of genomes under study orthologous genes have been identified beforehand, and only groups of orthologous genes (also known as *gene families*) are considered that have exactly one member in each genome. In this model, a variety of genomic similarity (or distance) measures have been studied and are relatively easy to compute [1–4]. However, the singleton gene family is a great oversimplification compared to what we find in nature. Therefore, more general models have been devised where several genes from the same family can exist in one genome. The computation of genomic similarities in these cases is generally much more difficult, though. In fact, many problem variants are NP-hard [5–9].

Another biological inaccuracy arises from the fact that a gene family assignment is not always without dispute, because orthology is usually not known but just predicted, and most prediction methods require some arbitrary threshold, deciding when two genes belong to the same family and when not. Therefore *gene family-free* measures have recently been proposed, based on pairwise similarities between genes [10–13]. While the resulting similarity measures are very promising, their computation is usually not easier than for the family-based models and therefore NP-hard as well [10,13].

In this paper, we study an intermediate approach between family-based and family-free genomic similarity measures, *gene connections*. It requires some preprocessing of the genes contained in the genomes under study, but in a less stringent way than in the family-based approach. On the other hand, the resulting data structure is less complex than in the family-free approach, where arbitrary (real-valued) similarities between genes are considered. This intermediate status allows us to achieve results comparable to those for family-free methods, but at time complexities similar to those for the family-based approach.

The paper is structured as follows. We first define three new genome similarity measures based on conserved gene adjacencies (Sect. 2), followed by some pointers to related literature (Sect. 3). Each of the three following sections is then devoted to one of the similarity measures. We show that the first problem can be computed in polynomial time, but is biologically quite simplistic. The second

one, while avoiding some of the weaknesses of the first, is NP-hard to compute and can therefore not be applied for genomes of realistic size. The third measure, finally, provides a compromise between biological relevance and computational complexity. In Sect. 7 we compare the results obtained with our similarity measures experimentally, using a large data set of plant (rosid) genomes. The last section concludes the paper.

The implemented algorithms used in this work as well as the studied dataset are available for download from http://bibiserv.cebitec.uni-bielefeld.de/newdist.

2 Basic Definitions

An *alphabet* is a finite set of *characters*. A *string* over an alphabet $\mathcal{A}$ is a sequence of characters from $\mathcal{A}$. Given a string S, $S[i]$ refers to the ith character of S and $|S|$ is the *length* of S, i.e., the number of characters in S. In a *signed string* S, each character is labeled with a sign, denoted $sgn_S(i)$ for the character at index position i. A sign is either positive $(+)$ or negative $(-)$. In comparative genomics, for example, the signs may indicate the orientations of genes on their genomic sequences, which themselves are represented as strings. Therefore in this paper we use the term *gene* as a synonym for "signed character" and the term *genome* as a synonym for "signed string".

Definition 1 (gene connection graph). *Given two genomes S and T, a* gene connection graph *$G(S,T)$ of S and T is a bipartite graph with one vertex for each gene of S and one vertex for each gene of T. An edge between two vertices, one from S and one from T, indicates that there is some* connection *between the two genes represented by these vertices.*

The term *connection* in the above definition is not very specific. Depending on the data set and context, connections may be defined based on gene homology, sequence similarity, functional relatedness, or any other similarity measure between genes.

For ease of notation, we let $S[i]$ denote both the ith gene of genome S, as well as the vertex of G representing this gene. Similar for $T[j]$. The set of edges of a graph G is denoted by $E(G)$. The size of a graph G is the number of its edges, $|G| = |E(G)|$. Further, we define a *connection function* t that returns for an index position i of S the list $t(i)$ of index positions in T that are connected to $S[i]$ by an edge in $G(S,T)$. That is, $t(i) = [j \mid (i,j) \in E(G(S,T))$ for $1 \leq j \leq |T|]$. The function $s(j)$ for an index position of T is defined analogously.

A pair of adjacent index positions (i,i') with $i' = i+1$ in a string is called an *adjacency*. Note that this definition of adjacency only considers direct neighborhood of genes ($i' = i+1$), while all our following uses of this term refer to an extended definition given by Zhu *et al.* [14], who introduced *generalized gene adjacencies* as follows:

Definition 2 (adjacency). *Given an integer $\theta \geq 1$, a pair of index positions (i,i') with $i' \leq i+\theta$ in a string is a (θ-)* adjacency.

In other words, two genes of the same genome form a θ-adjacency if the number of genes between them is less than θ. In the following we will frequently differentiate between *simple adjacencies* ($\theta = 1$) and *generalized adjacencies* ($\theta \geq 1$).

As mentioned in the Introduction, in this paper we are interested in defining measures of similarity to compare pairs of genomes. A simple approach is based on their number of *conserved adjacencies*. Although below we will study different variants of similarities, they all use the following basic notion of conserved adjacency:

Definition 3 (conserved adjacency). *Given two genomes S and T and a gene connection graph $G(S,T)$, a pair of adjacencies (i,i') in S and (j,j') in T is called a* conserved adjacency*, denoted $(i,i'||j,j')$, if one of the following two holds:*

(a) $(i,j) \in E(G(S,T))$, $(i',j') \in E(G(S,T))$, $sgn_S(i) = sgn_T(j)$ *and* $sgn_S(i') = sgn_T(j')$*; or*
(b) $(i,j') \in E(G(S,T))$, $(i',j) \in E(G(S,T))$, $sgn_S(i) \neq sgn_T(j')$ *and* $sgn_S(i') \neq sgn_T(j)$.

For an illustration of these definitions, see Fig. 1.

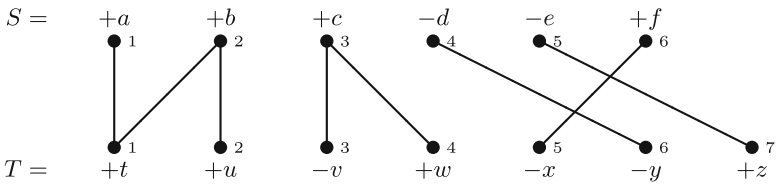


Fig. 1. Gene connection graph of two genomes $S = (+a, +b, +c, -d, -e, +f)$ (top row) and $T = (+t, +u, -v, +w, -x, -y, +z)$ (bottom row). Conserved 2-adjacencies are $(1,2||1,2)$, $(2,3||2,4)$, $(3,4||4,6)$ and $(5,6||5,7)$. Note that $(2,3||1,3)$, $(2,3||2,3)$, $(4,5||6,7)$ and $(4,6||5,6)$ are no conserved 2-adjacencies because the signs do not match the definition.

We further denote two conserved adjacencies as *conflicting* if their intervals in either genome are overlapping:

Definition 4 (conflicting conserved adjacencies). *Two conserved adjacencies $(i,i'||j,j')$ and $(k,k'||l,l')$ are* conflicting *if (1) $(i,i'||j,j') \neq (k,k'||l,l')$ and (2) $[i,i'-1] \cap [k,k'-1] \neq \emptyset$ or $[j,j'-1] \cap [l,l'-1] \neq \emptyset$.*

Subsequently a set of conserved adjacencies is denoted as *non-conflicting* if the above-defined property does not hold between any two of its members.

In the example of Fig. 1, $(3,4||4,6)$ and $(5,6||5,7)$ are the only conflicting conserved adjacencies. All other pairs are non-conflicting.

The different similarity measures that we consider in this work are expressed by the following three problem statements:

Problem 1 (total adjacency model). Given two genomes S and T and a gene connection graph $G(S,T)$, count the number of pairs of index positions (i,i') in S and (j,j') in T that form a conserved adjacency. In other words, compute

$$adj(S,T) = |\{(i,i'||j,j') \mid 1 \leq i < i' \leq |S| \text{ and } 1 \leq j < j' \leq |T|\}|.$$

Because a gene connection graph $G(S,T)$ is not limited to one-to-one connections between genes of genomes S and T, solutions to Problem 1 may biologically not be very plausible. Therefore we define a second measure, motivated by the one used in [10,11], which asks for one-to-one correspondences between genes of S and T in its solutions:

Problem 2 (gene matching model). Given two genomes S and T, a gene connection graph $G(S,T)$ and a real-valued parameter $\alpha \in [0,1]$, find a bipartite matching M in $G(S,T)$ such that the induced sequences S^M and T^M maximize the measure

$$\mathcal{F}_\alpha(M) = \alpha \cdot adj(S^M, T^M) + (1-\alpha) \cdot edg(M),$$

where $edg(M) = |M|$ is the size of matching M. (The induced sequences S^M and T^M are the subsequences of S and T, respectively, that contain those characters incident to edges of M.)

As we will see later in this paper, solving Problem 2 is NP-hard even for simple adjacencies. Therefore we define a third, intermediate measure, which is more efficient to compute in practice, while producing one-to-one correspondences between gene extremities. It is defined as the size of the largest subset of non-conflicting conserved adjacencies found in a pair of genomes:

Problem 3 (adjacency matching model). Given two genomes S and T and a gene connection graph $G(S,T)$, let C be the set of conserved adjacencies between S and T. Compute the size $|C^\star|$ of a maximum cardinality set of non-conflicting conserved adjacencies $C^\star \subseteq C$.

3 Related Work

As mentioned above, the *gene connection graph* input format that we propose here is an intermediate between gene families and the family-free model. Indeed, we do not require the gene connection graph to be transitive, which is the main difference to the *gene family graph*, where vertices are assigned to genes and edges are drawn between genes from different genomes whenever they belong to the same family, thus forming bipartite cliques. (This graph has not been introduced under this name in the literature, but is implicitly mentioned already in [15] and later more explicitly in [10].) On the other end, the *gene similarity graph* [11] is a weighted version of the gene connection graph, increasing the expression power by its ability to represent different strengths of gene connections.

The only previous use of such an intermediate model in comparative genomics that we are aware of is in the form of *indeterminate strings* in [12].

Definition 5 (indeterminate string, signed indeterminate string). *Given an alphabet $\mathcal{A}$, a string S over the power set $\mathcal{P}(\mathcal{A})\backslash\{\emptyset\}$ is called an* indeterminate string *over $\mathcal{A}$. In other words, for $1 \leq i \leq n$, $\emptyset \neq S[i] \subseteq \mathcal{A}$. In a* signed indeterminate string *S, any index position i has a sign $sgn_S(i)$, which therefore is the same for all characters at that position.*

Given two genomes S and T and a gene connection graph $G(S,T)$, it is easy to create a pair of signed indeterminate strings S' and T' over an alphabet $\mathcal{A}'$ that contain the same set of conserved adjacencies as S and T: For any edge $e = (S[i], T[j])$ of $G(S,T)$, create one symbol $e' \in \mathcal{A}'$ and let $e' \in S'[i]$ and $e' \in T'[j]$. The signs are just transferred from S and T to S' and T', respectively: $sgn_{S'}[i] = sgn_S[i]$ for all i, $1 \leq i \leq |S|$, and $sgn_{T'}[j] = sgn_T[j]$ for all j, $1 \leq j \leq |T|$.

Conversely, given two indeterminate strings S' and T', we can easily create sequences S and T and the corresponding gene connection graph with the same set of conserved adjacencies. In order to do this, let $\mathcal{A} = \{1, 2, \ldots, |S'|, 1', 2', \ldots, |T'|'\}$, set $S = sgn_{S'[1]}1, \ldots, sgn_{S'[|S'|]}|S'|$, $T = sgn_{T'[1]}1', \ldots, sgn_{T'[|T'|]}|T'|'$, and create in $G(S,T)$ an edge $e = (S[i], T[j])$ whenever $S'[i] \cap T'[j] \neq \emptyset$.

Clearly, all the information about conserved adjacencies between these two representations is identical, while sometimes the graph representation and sometimes the representation as signed indeterminate string is more concise.

Indeterminate strings in [12] were used to identify regions of common gene content (*gene clusters*) in two genomes, which is important in functional genomics. Here our focus is on conserved adjacencies (which can be seen as small clusters of just two genes) for defining whole-genome similarities. Similar measures are known for singleton gene families as the *breakpoint distance* [16,17], have been extended to gene families in [5,7,15] and were defined for the family-free model in [10].

4 An Optimal Solution for Problem 1

In order to solve Problem 1, we construct a list L of edges of $G(S,T)$ using connection function $t(i)$ for $1 \leq i \leq |S|$. In doing so, we assume that the elements of $t(i)$, $1 \leq i \leq |S|$, are sorted in increasing order. If this is not given as input, it can always be achieved by applying counting sort to all lists $t(i)$ in overall $O(|S| + |T| + |G(S,T)|)$ time, which is proportional to the input size.

We present with Algorithm 1 a solution to Problem 1 for simple adjacencies and subsequently extend this approach for the generalized case. Our algorithm is a simple, linear time procedure which uses three pointers e, e', e'' into list L. These pointers simultaneously traverse L while reporting any pair of adjacent parallel edges (e, e') or crossing edges (e, e'').

Correctness. Given a pair $(i,j) \in L$, there are overall four cases for the signs of index i in S and index j in T, each with two sub-cases for the signs of index $i+1$ in S and index $j+1$ or index $j-1$ in T, listed in the following.

Algorithm 1

Input: genomes S and T, gene connection graph $G(S,T)$

1: Create a list L of all edges $(i,j) \in E(G(S,T))$ ordered by primary index i and secondary index j
2: Let $e' = (i',j')$ and $e'' = (i'',j'')$ point to the second element of L
3: **for each** element $e = (i,j)$ of L in sorted order **do**
4: **if** $sgn_S(i) = sgn_T(j)$ **then**
5: **while** $i' < i+1$ **or** $(i' = i+1$ **and** $j' < j+1)$ **do**
6: advance $e' = (i',j')$ by one step in L
7: **end while**
8: **if** $(i',j') = (i+1,j+1)$ **and** $sgn_S(i') = sgn_T(j')$ **then**
9: report the conserved adjacency $(i,i'||j,j')$
10: **end if**
11: **else**
12: **while** $i'' < i+1$ **or** $(i'' = i+1$ **and** $j'' < j-1)$ **do**
13: advance $e'' = (i'',j'')$ by one step in L
14: **end while**
15: **if** $(i'',j'') = (i+1,j-1)$ **and** $sgn_S(i'') \neq sgn_T(j'')$ **then**
16: report the conserved adjacency $(i,i''||j'',j)$
17: **end if**
18: **end if**
19: **end for**

(1) If $sgn_S(i) = +$ and $sgn_T(j) = +$, then we have a conserved adjacency $(i,i+1||j,j+1)$ if and only if $(i+1,j+1) \in L$ and either $sgn_S(i+1) = +$ and $sgn_T(j+1) = +$ or $sgn_S(i+1) = -$ and $sgn_T(j+1) = -$.
(2) If $sgn_S(i) = +$ and $sgn_T(j) = -$, then we have a conserved adjacency $(i,i+1||j-1,j)$ if and only if $(i+1,j-1) \in L$ and either $sgn_S(i+1) = +$ and $sgn_T(j-1) = -$ or $sgn_S(i+1) = -$ and $sgn_T(j-1) = +$.
(3) If $sgn_S(i) = -$ and $sgn_T(j) = +$, then we have a conserved adjacency $(i,i+1||j-1,j)$ if and only if $(i+1,j-1) \in L$ and either $sgn_S(i+1) = -$ and $sgn_T(j-1) = +$ or $sgn_S(i+1) = +$ and $sgn_T(j-1) = -$.
(4) If $sgn_S(i) = -$ and $sgn_T(j) = -$, then we have a conserved adjacency $(i,i+1||j,j+1)$ if and only if $(i+1,j+1) \in L$ and either $sgn_S(i+1) = -$ and $sgn_T(j+1) = -$ or $sgn_S(i+1) = +$ and $sgn_T(j+1) = +$.

Clearly, cases 1 and 4 and cases 2 and 3 can be summarized to the two cases given in Algorithm 1.

Runtime Analysis. The list L has length $|G(S,T)|$ and can be constructed and sorted in linear time $O(|S| + |T| + |G(S,T)|)$, as discussed above. Each of the three edge pointers e, e' and e'' traverses L once from the beginning to the end, so that the **for** loop in lines 3–19 takes $O(|L|)$ time. Therefore the overall running time is $O(|S| + |T| + |G(S,T)|)$.

Space Analysis. The algorithm needs space only for the two input strings S and T, the list L and some constant-space variables. Therefore the space usage is of order $O(|S| + |T| + |G(S,T)|)$.

Extension to Generalized Adjacencies. Algorithm 1' solves Problem 1 for generalized adjacencies. Following the same strategy as Algorithm 1, the extension requires next to the main pointer e additional 2θ pointers into list L that are denoted e'_t and e''_t, $1 \leq t \leq \theta$. While it traverses through each element (i, j) in the list using pointer e, each pointer e'_t, $1 \leq t \leq \theta$, is subsequently increased to point to the smallest element larger than or equal to $(i + t, j + 1)$ in L. A copy $\hat{e}$ of pointer e'_t is then used to find candidates $(i + t, j + 1), \ldots, (i + t, j + \theta)$. Likewise, pointers e''_t, $1 \leq t \leq \theta$, are incremented to the smallest element larger than or equal to $(i + t, j - \theta)$, whereupon copy $\hat{e}$ of e''_t is used to find candidates $(i + t, j - \theta), \ldots, (i + t, j - 1)$.

All pointers e, e'_t, and e''_t, $1 \leq t \leq \theta$ are continuously increased, thus each traversing L once. Any instance of pointer $\hat{e}$ visits at most θ elements in each iteration, thus leading to an overall running time of $O(\theta^2|G(S,T)|)$. The running time is asymptotically optimal in the sense of worst case analysis, since there can be just as many θ-adjacencies in graph $G(S,T)$. Algorithm 1' requires $O(\theta + |S| + |T| + \theta^2|G(S,T)|)$ space.

5 Complexity of Problem 2

While one may hope that the intermediate status of the gene connection graph between the gene family graph and the gene similarity graph allows more efficient algorithms than for the more complex gene similarity graph, this is not the case for the gene matching model.

Only for $\alpha = 0$, we have $\mathcal{F}_\alpha(M) = edg(M) = |M|$ and therefore Problem 2 reduces to computing a maximum bipartite matching, which is possible in polynomial time [18]. However, this case is not very interesting because it completely ignores conserved adjacencies and just compares the gene content of the two genomes. All interesting cases are more difficult to solve, as the following theorem shows:[1]

Theorem 1 *Problem 2 is NP-hard for* $0 < \alpha \leq 1$.

Proof. We will focus on simple adjacencies ($\theta = 1$), as this is sufficient to prove Theorem 1. Inspired by the proof of Bryant [5] for the family-based case, we provide a P-reduction from Vertex Cover: Given a graph $\mathcal{G} = (V, E)$ and an integer λ, does there exist a subset $V' \subseteq V$ such that $|V'| = \lambda$ and each edge in E is adjacent to at least one vertex in V'?

Our reduction transforms an instance of Vertex Cover into an instance of the decision version of Problem 2: Given strings S and T, a gene connection graph $G(S,T)$, a real value α, $0 < \alpha \leq 1$, and a real value $F \geq 0$, does there exist a bipartite matching M in $G(S,T)$ such that $\mathcal{F}_\alpha(M) \geq F$?

Let $\mathcal{G} = (V, E)$ and λ be an instance of Vertex Cover with $V = \{v_1, v_2, \ldots, v_n\}$ and $E = \{e_1, e_2, \ldots, e_m\}$. Then we construct an alphabet $\mathcal{A}$

[1] A weaker result, namely the NP-hardness of Problem 2 for values of α between 0 and 1/3, can be found in [19].

Algorithm 1'

Input: genomes S and T, gene connection graph $G(S,T)$, gap threshold θ

Create a list L of all edges $(i,j) \in E(G(S,T))$ ordered by primary index i and secondary index j
Let $e'_t = (i'_t, j'_t)$ and $e''_t = (i'_t, j'_t)$, $1 \leq t \leq \theta$, point to the second element of L
for each element $e = (i,j)$ of L in sorted order **do**
  if $sgn_S(i) = sgn_T(j)$ **then**
    for each $e'_t = (i'_t, j'_t)$, $1 \leq t \leq \theta$ **do**
      while $i'_t < i + t$ **or** ($i'_t = i + t$ **and** $j'_t < j + 1$) **do**
        advance $e'_t = (i'_t, j'_t)$ by one step in L
      end while
      let $\hat{e} = (\hat{\imath}, \hat{\jmath}) \leftarrow e'_t$
      while $\hat{\imath} = i + t$ **and** $\hat{\jmath} \leq j + \theta$ **do**
        if $sgn_S(\hat{\imath}) = sgn_T(\hat{\jmath})$ **then**
          report the conserved adjacency $(i, \hat{\imath} || j, \hat{\jmath})$
        end if
        advance $\hat{e} = (\hat{\imath}, \hat{\jmath})$ by one step in L
      end while
    end for
  else
    for each $e''_t = (i''_t, j''_t)$, $1 \leq t \leq \theta$ **do**
      while $i''_t < i + t$ **or** ($i''_t = i + t$ **and** $j''_t < j - \theta$) **do**
        advance $e''_t = (i''_t, j''_t)$ by one step in L
      end while
      let $\hat{e} = (\hat{\imath}, \hat{\jmath}) \leftarrow e''_t$
      while $\hat{\imath} = i + t$ **and** $\hat{\jmath} < j - 1$ **do**
        if $sgn_S(\hat{\imath}) \neq sgn_T(\hat{\jmath})$ **then**
          report the conserved adjacency $(i, \hat{\imath} || \hat{\jmath}, j)$
        end if
        advance $\hat{e} = (\hat{\imath}, \hat{\jmath})$ by one step in L
      end while
    end for
  end if
end for

of size $2n + 4m + 2$ given by

$$\mathcal{A} = V \cup \{v'_i \mid v_i \in V\} \cup E \cup \{e'_i \mid e_i \in E\} \cup \{x_i, x'_i \mid 1 \leq i \leq m+1\}.$$

The two genomes S and T are constructed as follows:

$$S = v_1 v'_1 v_2 v'_2 \dots v_n v'_n x_1 x'_1 e_1 e'_1 x_2 x'_2 e_2 e'_2 x_3 x'_3 \dots x_m x'_m e_m e'_m x_{m+1} x'_{m+1}$$

and

$$T = x_{m+1} x'_{m+1} x_m x'_m \dots x_2 x'_2 x_1 x'_1 v_n \mathcal{E}_n v'_n v_{n-1} \mathcal{E}_{n-1} v'_{n-1} \dots v_1 \mathcal{E}_1 v'_1$$

where $\mathcal{E}_i$ is a string of the symbol pairs $e_j e'_j$ for the edges e_j that are adjacent to v_i. The gene connection graph $G(S,T)$ has an edge for each pair of identical

symbols $S[i]$ and $T[j]$. The parameter α may be chosen arbitrarily within the range $0 < \alpha \leq 1$.

First, we show that among the matchings maximizing the value $\mathcal{F}_\alpha$ for this problem, there is always at least one which is a maximal matching. Let M be a non-maximal matching in $G(S,T)$ maximizing $\mathcal{F}_\alpha$ and consider an edge $\ell \notin M$ that may be added to M, forming a new matching $M' = M \cup \{\ell\}$. Clearly, ℓ can dismiss at most two adjacencies of M in M', so $adj(M') \geq adj(M) - 2$. But in our construction, where the symbols of $\mathcal{A}$ (except the e_i and e'_i) are in reverse order in S related to T, and furthermore each e_i and each e'_i is between x_i and x_{i+1} in S, any new edge ℓ added to M can dismiss at most one adjacency: If ℓ is adjacent to a symbol a and the symbol a' is adjacent to another edge $\ell' \in M$ (or vice-versa) then $adj(M') = adj(M) + 1$. Moreover, if two partner edges $\ell, \ell' \notin M$ are added to M and thus $M' = M \cup \{\ell, \ell'\}$, then $adj(M') \geq adj(M)$ and $edg(M') = edg(M) + 2$. Therefore $\mathcal{F}_\alpha(M') > \mathcal{F}_\alpha(M)$ for $\alpha < 1$ and $\mathcal{F}_\alpha(M') \geq \mathcal{F}_\alpha(M)$ for $\alpha = 1$.

Next, we show that there is a vertex cover of size λ for a graph $\mathcal{G}$ if and only if Problem 2 has a solution with $F = \alpha(2m + 1 + (n - \lambda)) + (1 - \alpha)(2n + 4m + 2)$. Note that by construction of S, T and $G(S,T)$, conserved adjacencies in a maximal matching are only possible between pairs of the same symbol of $\mathcal{A}$, i.e. $v_i v'_i$, $e_i e'_i$ or $x_i x'_i$. Therefore we can simplify the notation and represent an adjacency $(i, i' || j, j')$ by the pair of elements in S, $S[i]S[i']$. Clearly, any maximal matching of $G(S,T)$ has $|S| = 2n + 4m + 2$ edges. Moreover, any maximal matching realizes at least the $2m + 1$ conserved adjacencies $e_i e'_i$ and $x_i x'_i$. The other possible adjacencies are the $v_i v'_i$. If there exists a solution with value $F = \alpha(2m + 1 + (n - \lambda)) + (1 - \alpha)|S|$, then there are at least $n - \lambda$ adjacencies involving $v_i v'_i$. These adjacencies are possible if the respective edges of $\mathcal{G}$ are covered by λ vertices. If we do not have a solution with value F, then $\mathcal{G}$ does not have a vertex cover of size λ. □

Solving Problem 2 for simple adjacencies, we make use of a method described in [19], that was originally developed for solving the gene family-free variant of Problem 2. In doing so, it constructs an integer linear program (ILP) similar to program `FFAdj-Int` described in [10]. It includes a preprocessing algorithm that identifies small components in gene similarity graphs which are part of an optimal solution. This approach enables the computation of optimal solutions for small and medium-sized gene similarity graphs. However, as the method is specifically tailored for gene family-free analysis, it does not perform very efficiently on gene connection graphs, as we will see in Sect. 7. We refer to this ILP and its preprocessing step as Algorithm 2.

We further believe it will be difficult to develop a practical algorithm solving Problem 2 for generalized adjacencies.

6 Computing Exact Solutions for Problem 3

We present a polynomial time algorithm solving Problem 3 for simple adjacencies which makes use of the following graph structure:

Definition 6 (conserved adjacencies graph). *Given two genomes S and T and a set $C = \{(i_1, i'_1||j_1, j'_1), \ldots, (i_n, i'_n||j_n, j'_n)\}$ of conserved adjacencies between S and T, the* conserved adjacencies graph $A_C(S,T)$ *is a bipartite graph with one vertex for each gene adjacency (i, i') of S that occurs in C and one vertex for each gene adjacency (j, j') of T that occurs in C. The edges correspond to the conserved adjacencies in C.*

Pseudocode of our algorithm is shown in Algorithm 3. Clearly its running time is dominated by the time to compute a maximum matching in line 3, which in unweighted bipartite graphs with n vertices and m edges is possible in $O(m\sqrt{n})$ time [18]. In our case $n \leq |S| + |T| - 2$ and $m \leq n^2$, therefore Algorithm 3 takes overall $O((|S| + |T|)^{5/2})$ time.

Algorithm 3

Input: genomes S and T, gene connection graph $G(S,T)$
1: Let C be the set of conserved adjacencies reported by Algorithm 1 applied to S, T and $G(S,T)$
2: Construct the conserved adjacencies graph $A = A_C(S,T)$
3: Compute a maximum bipartite matching M on A
4: return $|M|$

Extension to Generalized Adjacencies. Other than for the first two problems, the properties of Problem 3 change drastically when generalized adjacencies are considered. Because a θ-adjacency corresponds to an interval of up to $\theta + 1$ consecutive genes, the intervals of two θ-adjacencies for $\theta \geq 2$ can overlap on more than two genes, or even be contained in one another. The complexity of Problem 3 for $\theta \geq 2$ remains an open question.

Solving Problem 3 for generalized adjacencies, we propose Algorithm 3' that follows the same strategy as its counterpart for simple adjacencies. However, while for the latter it was possible to find a maximum subset of non-conflicting θ-adjacencies using a maximum matching approach, here we propose an ILP, described in Fig. 2. The ILP makes use of two types of binary variables, $\mathbf{a}(i,j)$ for each edge (i,j) in the gene connection graph $G(S,T)$, and $\mathbf{b}(i,i',j,j')$ for each θ-adjacency $(i,i'||j,j')$ in C_θ. We say that a binary variable is *saturated* if it is assigned value 1. While maximizing the number of saturated $\mathbf{b}(.)$ variables (which represents the output of the program), our ILP imposes matching constraints (`C.01`) for the set of edges in selected θ-adjacencies. Further constraints (`C.02`) ensure that for each θ-adjacency $(i,i'||j,j')$ (a) both edges between its corresponding genes are saturated and (b) no saturated edge is incident to a gene in interval $[i+1, i'-1]$ of genome S (i.e. a possibly empty interval corresponding to all genes between i and i') and interval $[j+1, j'-1]$ of genome T, respectively.

Algorithm 3'

Input: genomes S and T, gene connection graph $G(S,T)$, gap threshold θ

1: Let C_θ be the set of conserved adjacencies reported by Algorithm 1' applied to S, T and $G(S,T)$
2: Compute a maximum cardinality set of non-conflicting conserved θ-adjacencies $C^\star_\theta \subseteq C_\theta$ using the ILP given in Fig. 2
3: return $|C^\star_\theta|$

7 Experimental Results

Genomic Dataset. We study genomes of 18 rosid species (see Table 1). Rosids are a prominent subclass of flowering plants to which also many agricultural crops belong. The genomic sequences of the studied species were obtained from *Phytozyme* [20][2], an online resource of the Joint Genome Institute providing

ILP solving Step 2 in Algorithm 3'

Objective:

$$\texttt{maximize} \sum_{(i,i'||j,j'')\in C_\theta} \mathbf{b}(i,i',j,j')$$

Constraints:

(C.01) **for each** $i \leftarrow 1$ **to** $|S|$, $\sum_{j\in t(i)} \mathbf{a}(i,j) \leq 1$

for each $j \leftarrow 1$ **to** $|T|$, $\sum_{i\in s(j)} \mathbf{a}(i,j) \leq 1$

(C.02) **for each** $(i,i'||j,j') \in C_\theta$

if $sgn_S(i) = sgn_S(i')$ **then**
$2\cdot\mathbf{b}(i,i',j,j') - \mathbf{a}(i,j) - \mathbf{a}(i',j') \leq 0$
otherwise
$2\cdot\mathbf{b}(i,i',j,j') - \mathbf{a}(i,j') - \mathbf{a}(i',j) \leq 0$
end if

for each $\hat{\imath} \leftarrow [i+1, i'-1]$ **and each** $\hat{\jmath}$ **in** $t(\hat{\imath})$
$\mathbf{b}(i,i',j,j') + \mathbf{a}(\hat{\imath},\hat{\jmath}) \leq 1$
for each $\hat{\jmath} \leftarrow [j+1, j'-1]$ **and each** $\hat{\imath}$ **in** $s(\hat{\jmath})$
$\mathbf{b}(i,i',j,j') + \mathbf{a}(\hat{\imath},\hat{\jmath}) \leq 1$

end for

Domains:

(D.01) **for each** $(i,j) \in E(G(S,T))$, $\mathbf{a}(i,j) \in \{0,1\}$

(D.02) **for each** $(i,i'||j,j') \in C_\theta$, $\mathbf{b}(i,i',j,j') \in \{0,1\}$

Fig. 2. Integer linear program for finding a maximum subset of non-conflicting conserved adjacencies of a given set C_θ.

[2] The described experiments were performed on data sets of Phytozyme v10.3.

databases and tools for comparative genomics analyses of plant genomes. Most of the studied plant genomes are partially assembled, comprising up to 5,000 scaffolds covering one or more annotated protein coding genes. While the smallest genome in our data set contains roughly 24,500 genes, the largest spans with 56,000 genes more than twice as many. Rosids, just like many other plants, met their evolutionary fate through multiple events of whole genome duplication, followed by periods of fractionation, in which many duplicated genes were lost again.

Construction of Gene Connection and Gene Family Graphs. Next to the genomic sequences and gene annotations, Phytozyme also provides gene family information in form of co-orthologous clusters computed by InParanoid [21]. InParanoid follows a seed-based strategy by identifying pairs of orthologous genes (the "seeds") through reciprocal best BLASTP hits. These are subsequently used to recruit inparalogs, eventually forming groups of co-orthologous genes.

Table 1. The genomic dataset of 18 rosid species used in our experiments.

Species	Version	# genes	# scaffolds	Reference
A. thaliana	TAIR10	27,416	7	[22]
B. rapa	FPSc v1.3	40,492	669	[20]
B. stricta	v1.2	27,416	854	[20]
C. clementina	v1.0	24,533	94	[23]
C. rubella	v1.0	26,521	123	[24]
E. grandis	v1.1	36,376	1,315	[25]
E. salsugineum	v1.0	26,351	61	[26]
F. vesca	v1.1	32,831	8	[27]
G. max	Wm82.a2	56,044	147	[28]
G. raimondii	v2.1	37,505	133	[29]
L. usitatissimum	v1.0	43,471	1,028	[30]
M. truncatula	Mt4.0v1	50,894	1,033	[31]
P. persica	v1.0	27,864	59	[32]
P. trichocarpa	v3.0	41,335	379	[33]
P. vulgaris	v1.0	27,197	91	[34]
R. communis	v0.1	31,221	4,962	[35]
T. cacao	v1.1	29,452	99	[36]
V. vinifera	Genoscope.12X	26,346	33	[37]

We ran BLASTP on all genes of our dataset using an e-value threshold of 10^{-5} and otherwise default parameter settings. We then constructed gene connection graphs for all 153 genome pairs by establishing edges between vertices whose

corresponding genes share reciprocal BLASTP hits. We refer to these graphs as *BLASTP GC graphs*. Similarly, we constructed pairwise gene family graphs using InParanoid's homology assignment, which we refer to as *InParanoid GF graphs*.

Unsurprisingly, the BLASTP GC graphs are much larger in size than the InParanoid GF graphs. We observed average sizes of 150,000 edges for the former, whereas the latter graphs had on average only one fifth of this size. Moreover, only 4 % of edges in InParanoid GF graphs were not contained in their BLASTP GC counterparts. Lacking ground truth of homologies in our dataset, we take a conservative stance by assuming that InParanoid's homology assignment can be considered true, or, in other words, that it contains only a negligible number of false positives. However, we conclude from a previous study [38], in which InParanoid (as well as all other gene family prediction tools in that study) exhibited a poor recall, that the homology assignment may be incomplete. That being said, we regard the edges of BLASTP GC graphs with suspicion. In doing so, we assume many of them leading to false positive homology assignments. We perform subsequent analysis to outline a possible procedure of identifying additional potential homologies that are supported by conservation in gene order in BLASTP GC graphs.

Implementation. All computations were performed on a Linux machine using a single 2.3 GHz CPU. We implemented Algorithms 1, 1', 3, and 3' in Python. For Algorithm 2 we used the implementation of [19]. In Algorithm 3, the maximum cardinality matching was computed using an implementation of Hopcroft and Karp's algorithm [18] provided by the Python-based NetworkX[3] library. The ILPs of Algorithms 2 and 3' were run using CPLEX[4], a solver for various types of linear and quadratic programs.

Runtimes. The runtimes of Algorithms 1 and 3 are shown in Fig. 3 (left). The runtime analysis was repeated 5 times and is visualized by whisker plots. For each of the 153 BLASTP GC graphs in our dataset, the computation was finished in less than 50 CPU seconds. Moreover, our evaluation reveals that the enumeration of the set of conserved adjacencies in our dataset requires often more time than the subsequent computation of the maximum matching for Algorithm 3. The plot on the right side of Fig. 3 shows that the runtimes of Algorithm 1' for $\theta = 2, 3, 4$ increase only moderately for higher values of θ.

Comparing our methods to the gene family-free approach, an implementation of a heuristic method described in [10] failed to return a result for the gene family free variant of Problem 2 on the BLASTP GC graph of *R. communis* and *V. vinifera* within 36 hours of computation. Surprisingly, running Algorithm 2 with $\alpha = 0.1$ just as long, we were able to obtain a suboptimal solution of which CPLEX reported an optimality gap of only 1.89 %. Nevertheless, as a reference for comparison with our various models it would be even more informative to

[3] http://networkx.github.io/.
[4] http://www.ibm.com/software/integration/optimization/cplex-optimizer/.

have optimal solutions of these problems. We leave it as an open problem whether it is possible to improve our ILPs in order to achieve this.

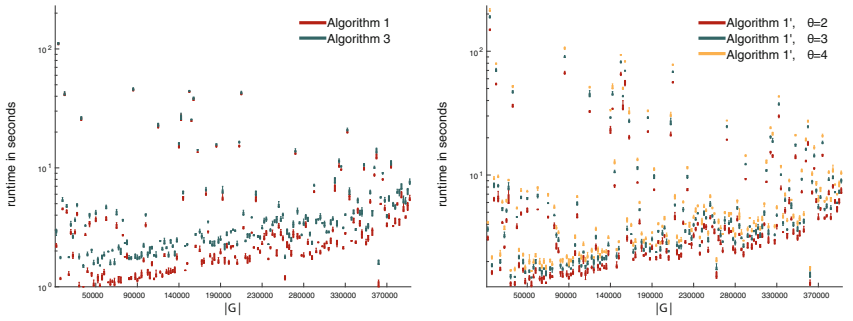


Fig. 3. Left: runtimes of Algorithms 1 and 3 for all 153 BLASTP GC graphs of the studied dataset. Right: runtimes of Algorithm 1' for $\theta = 2, 3, 4$.

Further, we were able to compute exact results for Problem 3 and $\theta = 2$ with Algorithm 3' for all 153 but 19 BLASTP GC graphs and all but 16 InParanoid GC graphs, limiting computation time to two hours per graph instance.

Gene Connection vs. Gene Family Graphs. The overlap between the set of conserved simple adjacencies identified in BLASTP GC graphs and in InParanoid GF graphs is visualized in the left plot of Fig. 4. Overall, 70 % of the conserved adjacencies of the InParanoid GF graphs were also found in the BLASTP GC graphs whereas we find in the latter 90 % more conserved adjacencies than in the former. Investigating the high number of InParanoid adjacencies that are missing in BLASTP GC graphs, we discovered that many generalized adjacencies of the former span genes that are connected (and therefore breaking the surrounding adjacency) in their BLASTP GC counterparts. However, the mean number of connected intervening genes was only 1.4. In fact, the overlap of 2-adjacencies in BLASTP GC graphs with 1-adjacencies of InParanoid GF graphs was at 83 % of all adjacencies in the latter (Fig. 4, right plot).

Lastly, Fig. 5 visualizes the number of non-conflicting conserved adjacencies in BLASTP GC and InParanoid GF graphs computed for $\theta = 1$ using Algorithm 3 (left plot) and computed for $\theta = 2$ using Algorithm 3' (right plot). For the former we observed on average 42 % more non-conflicting conserved adjacencies in BLASTP GC graphs when compared to their InParanoid GF counterparts, whereas for the latter, this number dropped to 32 %. Nevertheless, from $\theta = 1$ to $\theta = 2$ the absolute number of non-conflicting conserved adjacencies increases on average by 27 % for BLASTP GC graphs, respectively by 37 % for InParanoid GF graphs.

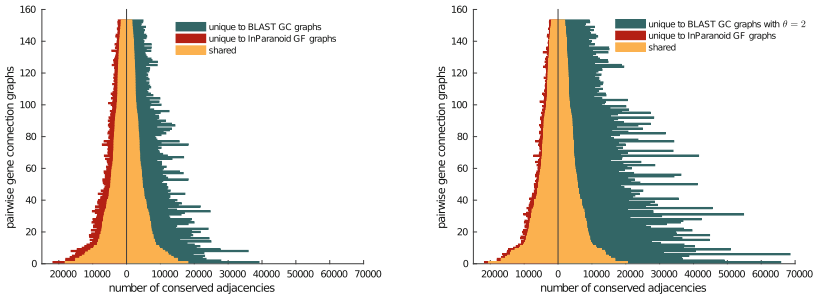


Fig. 4. Overlap of conserved adjacencies between BLASTP GC and InParanoid GF graphs

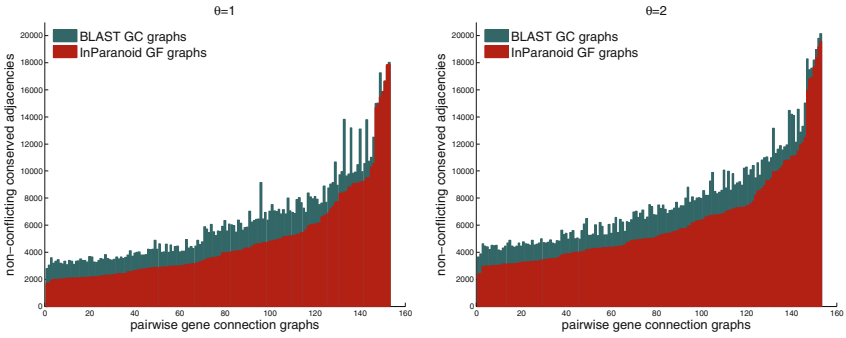


Fig. 5. Numbers of non-conflicting conserved adjacencies in BLASTP GC and InParanoid GF graphs for $\theta = 1$ (left) and $\theta = 2$ (right).

8 Conclusion

We have presented new similarity measures for complete genomes, thereby defining gene connections as an intermediate model of genome similarity representations, between gene families and the gene family-free approach. Our theoretical results with some problem variants being polynomial and others being NP-hard show that we are very close to the hardness border of similarity computations between genomes with unrestricted gene content. On the practical side we could show that the computation of genomic similarities in the gene connection model gives meaningful results and is possible in reasonable time, if the measures and algorithms are designed carefully.

A few questions remain open, though. While Problem 3 is polynomial for $\theta = 1$, the complexity for larger values of θ is unknown. Moreover, it is always difficult to choose optimal values for parameters like the gap threshold θ. It will certainly be worthwhile to examine whether parameter estimation methods for generalized adjacencies as the ones developed in [39] can be adapted to the gene connection model.

Various model extensions can also be envisaged. The adjacency matching model (Problem 3) removes inconsistencies from the output of the total adjacencies model (Problem 1) by computing a maximum matching on it. It could be tested whether other criteria to remove genes from the genomes and thus derive consistent sets of conserved adjacencies yield even better results. Moreover, the resulting reduced genomes with conserved adjacencies could be used to predict orthologies between the involved genes, not only to compute genomic similarities.

An alternative objective function for our problem formulations, instead of counting (generalized) gene adjacencies, could be a variant of the *summed adjacency disruption number* [40] that also allows to distinguish between small and larger distortions in gene order.

Finally, Algorithm 3 can easily be generalized for weighted gene similarities (instead of gene connections). It remains to be evaluated if such a more fine-grained measure in the spirit of a family-free analysis has advantages compared to the unit-cost measures studied in this paper.

Acknowledgements. The research of LABK and SD is partially supported by FAPERJ and CNPq. This work was performed while JS was on sabbatical as Special Visiting Researcher at UFF in Niteri, Brazil, funded by Cincia sem Fronteiras/CAPES.

References

1. Sankoff, D.: Edit distance for genome comparison based on non-local operations. In: Apostolico, A., Galil, Z., Manber, U., Crochemore, M. (eds.) CPM 1992. LNCS, vol. 644, pp. 121–135. Springer, Heidelberg (1992)
2. Hannenhalli, S., Pevzner, P.A.: Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. J. ACM **46**(1), 1–27 (1999)
3. Yancopoulos, S., Attie, O., Friedberg, R.: Efficient sorting of genomic permutations by translocation, inversion and block interchange. Bioinformatics **21**(16), 3340–3346 (2005)
4. Bergeron, A., Mixtacki, J., Stoye, J.: A new linear time algorithm to compute the genomic distance via the double cut and join distance. Theor. Comput. Sci. **410**(51), 5300–5316 (2009)
5. Bryant, D.: The complexity of calculating exemplar distances. In: Sankoff, D., Nadeau, J.H. (eds.) Comparative Genomics. Computational Biology Series, vol. 1, pp. 207–211. Kluwer Academic Publishers, London (2000)
6. Chen, X., Zheng, J., Fu, Z., Nan, P., Zhong, Y., Lonardi, S., Jiang, T.: Assignment of orthologous genes via genome rearrangement. IEEE/ACM Trans. Comput. Biol. Bioinform. **2**(4), 302–315 (2005)
7. Angibaud, S., Fertin, G., Rusu, I., Thevenin, A., Vialette, S.: Efficient tools for computing the number of breakpoints and the number of adjacencies between two genomes with duplicate genes. J. Comput. Biol. **15**(8), 1093–1115 (2008)
8. Bulteau, L., Jiang, M.: Inapproximability of (1,2)-exemplar distance. IEEE/ ACM Trans. Comput. Biol. Bioinform. **10**(6), 1384–1390 (2012)
9. Shao, M., Lin, Y., Moret, B.M.E.: An exact algorithm to compute the double-cut-and-join distance for genomes with duplicate genes. J. Comput. Biol. **22**(5), 425–435 (2015)

10. Doerr, D., Thvenin, A., Stoye, J.: Gene family assignment-free comparative genomics. BMC Bioinform. **13**(Suppl. 19), S3 (2012)
11. Braga, M.D.V., Chauve, C., Doerr, D., Jahn, K., Stoye, J., Thvenin, A., Wittler, R.: The potential of family-free genome comparison. In: Chauve, C., El-Mabrouk, N., Tannier, E. (eds.) Models and Algorithms for Genome Evolution. Computational Biology Series, vol. 19, pp. 287–307. Springer, London (2013)
12. Doerr, D., Stoye, J., Bcker, S., Jahn, K.: Identifying gene clusters by discovering common intervals in indeterminate strings. BMC Bioinform. **15**(Suppl. 6), S2 (2014)
13. Martinez, F.V., Feijo, P., Braga, M.D.V., Stoye, J.: On the family-free DCJ distance and similarity. Algorithms Mol. Biol. **10**, 13 (2015)
14. Zhu, Q., Adam, Z., Choi, V., Sankoff, D.: Generalized gene adjacencies, graph bandwidth, and clusters in yeast evolution. IEEE/ACM Trans. Comput. Biol. Bioinform. **6**(2), 213–220 (2009)
15. Sankoff, D.: Genome rearrangement with gene families. Bioinformatics **15**(11), 909–917 (1999)
16. Blanchette, M., Kunisawa, T., Sankoff, D.: Gene order breakpoint evidence in animal mitochondrial phylogeny. J. Mol. Evol. **49**(2), 193–203 (1999)
17. Tannier, E., Zheng, C., Sankoff, D.: Multichromosomal median and halving problems under different genomic distances. BMC Bioinform. **10**, 120 (2009)
18. Hopcroft, J.E., Karp, R.M.: An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. SIAM J. Comput. **2**(4), 225–231 (1973)
19. Doerr, D.: Gene family-free genome comparison. Ph.D. thesis, Faculty of Technology, Bielefeld University, Germany (2015)
20. Goodstein, D.M., Shu, S., Howson, R., Neupane, R., Hayes, R.D., Fazo, J., Mitros, T., Dirks, W., Hellsten, U., Putnam, N., Rokhsar, D.S.: Phytozome: a comparative platform for green plant genomics. Nucleic Acids Res. **40**(Database issue), D1178–D1186 (2012)
21. Sonnhammer, E.L.L., Östlund, G.: Inparanoid 8: orthology analysis between 273 proteomes, mostly eukaryotic. Nucleic Acids Res. **43**(Database issue), D234–D239 (2015)
22. Lamesch, P., Berardini, T.Z., Li, D., Swarbreck, D., Wilks, C., Sasidharan, R., Muller, R., Dreher, K., Alexander, D.L., Garcia-Hernandez, M., Karthikeyan, A.S., Lee, C.H., Nelson, W.D., Ploetz, L., Singh, S., Wensel, A., Huala, E.: The arabidopsis information resource (tair): improved gene annotation and new tools. Nucleic Acids Res. **40**(Database issue), D1202–D1210 (2011)
23. Wu, G.A., Prochnik, S., Jenkins, J., Salse, J., Hellsten, U., Murat, F., Perrier, X., Ruiz, M., Scalabrin, S., Terol, J., Takita, M.A., Labadie, K., Poulain, J., Couloux, A., Jabbari, K., Cattonaro, F., Del Fabbro, C., Pinosio, S., Zuccolo, A., Chapman, J., Grimwood, J., Tadeo, F.R., Estornell, L.H., Muñoz-Sanz, J.V., Ibanez, V., Herrero-Ortega, A., Aleza, P., Pérez-Pérez, J., Ramón, D., Brunel, D., Luro, F., Chen, C., Farmerie, W.G., Desany, B., Kodira, C., Mohiuddin, M., Harkins, T., Fredrikson, K., Burns, P., Lomsadze, A., Mark, B., Reforgiato, G., Freitas-Astúa, J., Quetier, F., Navarro, L., Roose, M., Wincker, P., Schmutz, J., Morgante, M., Machado, M.A., Talón, M., Jaillon, O., Ollitrault, P., Gmitter, F., Rokhsar, D.: Sequencing of diverse mandarin, pummelo and orange genomes reveals complex history of admixture during citrus domestication. Nat. Biotechnol. **32**(7), 656–662 (2014)

24. Slotte, T., Hazzouri, K.M., Ågren, J.A., Koenig, D., Maumus, F., Guo, Y.-L., Steige, K., Platts, A.E., Escobar, J.S., Newman, L.K., Wang, W., Mandáková, T., Vello, E., Smith, L.M., Henz, S.R., Steffen, J., Takuno, S., Brandvain, Y., Coop, G., Andolfatto, P., Hu, T.T., Blanchette, M., Clark, R.M., Quesneville, H., Nordborg, M., Gaut, B.S., Lysak, M.A., Jenkins, J., Grimwood, J., Chapman, J., Prochnik, S., Shu, S., Rokhsar, D., Schmutz, J., Weigel, D., Wright, S.I.: The Capsella rubella genome and the genomic consequences of rapid mating system evolution. Nat. Genet. **45**(7), 831–835 (2013)
25. Bartholomé, J., Mandrou, E., Mabiala, A., Jenkins, J., Nabihoudine, I., Klopp, C., Schmutz, J., Plomion, C., Gion, J.-M.: High-resolution genetic maps of eucalyptus improve Eucalyptus grandis genome assembly. New Phytol **206**(4), 1283–1296 (2015)
26. Yang, R., Jarvis, D.E., Chen, H., Beilstein, M.A., Grimwood, J., Jenkins, J., Shu, S., Prochnuk, S., Xin, M., Ma, C., Schmutz, J., Wing, R.A., Mitchell-Olds, T., Schumaker, K.S., Wang, X.: The reference genome of the halophytic plant *Eutrema salsugineum*. Front. Plant Sci. **4**, 46 (2013)
27. Shulaev, V., Sargent, D.J., Crowhurst, R.N., Mockler, T.C., Folkerts, O., Delcher, A.L., Jaiswal, P., Mockaitis, K., Liston, A., Mane, S.P., Burns, P., Davis, T.M., Slovin, J.P., Bassil, N., Hellens, R.P., Evans, C., Harkins, T., Kodira, C., Desany, B., Crasta, O.R., Jensen, R.V., Allan, A.C., Michael, T.P., Setubal, J.C., Celton, J.-M., Rees, D.J.G., Williams, K.P., Holt, S.H., Rojas, J.J.R., Chatterjee, M., Liu, B., Silva, H., Meisel, L., Adato, A., Filichkin, S.A., Troggio, M., Viola, R., Ashman, T.-L., Wang, H., Dharmawardhana, P., Elser, J., Raja, R., Priest, H.D., Bryant, D.W., Fox, S.E., Givan, S.A., Wilhelm, L.J., Naithani, S., Christoffels, A., Salama, D.Y., Carter, J., Girona, E.L., Zdepski, A., Wang, W., Kerstetter, R.A., Schwab, W., Korban, S.S., Davik, J., Monfort, A., Denoyes-Rothan, B., Arus, P., Mittler, R., Flinn, B., Aharoni, A., Bennetzen, J.L., Salzberg, S.L., Dickerman, A.W., Velasco, R., Borodovsky, M., Veilleux, R.E., Folta, K.M.: The genome of woodland strawberry (Fragaria vesca). Nat. Genet. **43**(2), 109–116 (2011)
28. Schmutz, J., Cannon, S.B., Schlueter, J., Ma, J., Mitros, T., Nelson, W., Hyten, D.L., Song, Q., Thelen, J.J., Cheng, J., Xu, D., Hellsten, U., May, G.D., Yu, Y., Sakurai, T., Umezawa, T., Bhattacharyya, M.K., Sandhu, D., Valliyodan, B., Lindquist, E., Peto, M., Grant, D., Shu, S., Goodstein, D., Barry, K., Futrell-Griggs, M., Abernathy, B., Du, J., Tian, Z., Zhu, L., Gill, N., Joshi, T., Libault, M., Sethuraman, A., Zhang, X.-C., Shinozaki, K., Nguyen, H.T., Wing, R.A., Cregan, P., Specht, J., Grimwood, J., Rokhsar, D., Stacey, G., Shoemaker, R.C., Jackson, S.A.: Genome sequence of the palaeopolyploid soybean. Nature **463**(7278), 178–183 (2010)
29. Paterson, A.H., Wendel, J.F., Gundlach, H., Guo, H., Jenkins, J., Jin, D., Llewellyn, D., Showmaker, K.C., Shu, S., Udall, J., Yoo, M.-J., Byers, R., Chen, W., Doron-Faigenboim, A., Duke, M.V., Gong, L., Grimwood, J., Grover, C., Grupp, K., Hu, G., Lee, T.-H., Li, J., Lin, L., Liu, T., Marler, B.S., Page, J.T., Roberts, A.W., Romanel, E., Sanders, W.S., Szadkowski, E., Tan, X., Tang, H., Xu, C., Wang, J., Wang, Z., Zhang, D., Zhang, L., Ashrafi, H., Bedon, F., Bowers, J.E., Brubaker, C.L., Chee, P.W., Das, S., Gingle, A.R., Haigler, C.H., Harker, D., Hoffmann, L.V., Hovav, R., Jones, D.C., Lemke, C., Mansoor, S., Rahman, M.U., Rainville, L.N., Rambani, A., Reddy, U.K., Rong, J.-K., Saranga, Y., Scheffler, B.E., Scheffler, J.A., Stelly, D.M., Triplett, B.A., Van Deynze, A., Vaslin, M.F.S., Waghmare, V.N., Walford, S.A., Wright, R.J., Zaki, E.A., Zhang, T., Dennis, E.S., Mayer, K.F.X., Peterson, D.G., Rokhsar, D.S., Wang, X., Schmutz, J.: Repeated polyploidization of gossypium genomes and the evolution of spinnable cotton fibres. Nature **492**(7429), 423–427 (2012)

30. Wang, Z., Hobson, N., Galindo, L., Zhu, S., Shi, D., McDill, J., Yang, L., Hawkins, S., Neutelings, G., Datla, R., Lambert, G., Galbraith, D.W., Grassa, C.J., Geraldes, A., Cronk, Q.C., Cullis, C., Dash, P.K., Kumar, P.A., Cloutier, S., Sharpe, A.G., Wong, G.K.S., Wang, J., Deyholos, M.K.: The genome of flax (Linum usitatissimum) assembled de novo from short shotgun sequence reads. Plant J. **72**(3), 461–473 (2012)
31. Young, N.D., Debellé, F., Oldroyd, G.E.D., Geurts, R., Cannon, S.B., Udvardi, M.K., Benedito, V.A., Mayer, K.F.X., Gouzy, J., Schoof, H., Van de Peer, Y., Proost, S., Cook, D.R., Meyers, B.C., Spannagl, M., Cheung, F., De Mita, S., Krishnakumar, V., Gundlach, H., Zhou, S., Mudge, J., Bharti, A.K., Murray, J.D., Naoumkina, M.A., Rosen, B., Silverstein, K.A.T., Tang, H., Rombauts, S., Zhao, P.X., Zhou, P., Barbe, V., Bardou, P., Bechner, M., Bellec, A., Berger, A., Bergès, H., Bidwell, S., Bisseling, T., Choisne, N., Couloux, A., Denny, R., Deshpande, S., Dai, X., Doyle, J.J., Dudez, A.-M., Farmer, A.D., Fouteau, S., Franken, C., Gibelin, C., Gish, J., Goldstein, S., González, A.J., Green, P.J., Hallab, A., Hartog, M., Hua, A., Humphray, S.J., Jeong, D.-H., Jing, Y., Jöcker, A., Kenton, S.M., Kim, D.-J., Klee, K., Lai, H., Lang, C., Lin, S., Macmil, S.L., Magdelenat, G., Matthews, L., McCorrison, J., Monaghan, E.L., Mun, J.-H., Najar, F.Z., Nicholson, C., Noirot, C., O'Bleness, M., Paule, C.R., Poulain, J., Prion, F., Qin, B., Qu, C., Retzel, E.F., Riddle, C., Sallet, E., Samain, S., Samson, N., Sanders, I., Saurat, O., Scarpelli, C., Schiex, T., Segurens, B., Severin, A.J., Sherrier, D.J., Shi, R., Sims, S., Singer, S.R., Sinharoy, S., Sterck, L., Viollet, A., Wang, B.-B., Wang, K., Wang, M., Wang, X., Warfsmann, J., Weissenbach, J., White, D.D., White, J.D., Wiley, G.B., Wincker, P., Xing, Y., Yang, L., Yao, Z., Ying, F., Zhai, J., Zhou, L., Zuber, A., Dénarié, J., Dixon, R.A., May, G.D., Schwartz, D.C., Rogers, J., Quetier, F., Town, C.D., Roe, B.A.: The medicago genome provides insight into the evolution of rhizobial symbioses. Nature **480**(7378), 520–524 (2011)
32. Verde, I., Abbott, A.G., Scalabrin, S., Jung, S., Shu, S., Marroni, F., Zhebentyayeva, T., Dettori, M.T., Grimwood, J., Cattonaro, F., Zuccolo, A., Rossini, L., Jenkins, J., Vendramin, E., Meisel, L.A., Decroocq, V., Sosinski, B., Prochnik, S., Mitros, T., Policriti, A., Cipriani, G., Dondini, L., Ficklin, S., Goodstein, D.M., Xuan, P., Del Fabbro, C., Aramini, V., Copetti, D., Gonzalez, S., Horner, D.S., Falchi, R., Lucas, S., Mica, E., Maldonado, J., Lazzari, B., Bielenberg, D., Pirona, R., Miculan, M., Barakat, A., Testolin, R., Stella, A., Tartarini, S., Tonutti, P., Arus, P., Orellana, A., Wells, C., Main, D., Vizzotto, G., Silva, H., Salamini, F., Schmutz, J., Morgante, M., Rokhsar, D.S.: The high-quality draft genome of peach (Prunus persica) identifies unique patterns of genetic diversity, domestication and genome evolution. Nat. Genet. **45**(5), 487–494 (2013)
33. Du, Q., Wang, L., Yang, X., Gong, C., Zhang, D.: Populus endo-β-1,4-glucanases gene family: genomic organization, phylogenetic analysis, expression profiles and association mapping. Planta **241**(6), 1417–1434 (2015)
34. Schmutz, J., McClean, P.E., Mamidi, S., Wu, G.A., Cannon, S.B., Grimwood, J., Jenkins, J., Shu, S., Song, Q., Chavarro, C., Torres-Torres, M., Geffroy, V., Moghaddam, S.M., Gao, D., Abernathy, B., Barry, K., Blair, M., Brick, M.A., Chovatia, M., Gepts, P., Goodstein, D.M., Gonzales, M., Hellsten, U., Hyten, D.L., Jia, G., Kelly, J.D., Kudrna, D., Lee, R., Richard, M.M.S., Miklas, P.N., Osorno, J.M., Rodrigues, J., Thareau, V., Urrea, C.A., Wang, M., Yu, Y., Zhang, M., Wing, R.A., Cregan, P.B., Rokhsar, D.S., Jackson, S.A.: A reference genome for common bean and genome-wide analysis of dual domestications. Nat. Genet. **46**(7), 707–713 (2014)

35. Chan, A.P., Crabtree, J., Zhao, Q., Lorenzi, H., Orvis, J., Puiu, D., Melake-Berhan, A., Jones, K.M., Redman, J., Chen, G., Cahoon, E.B., Gedil, M., Stanke, M., Haas, B.J., Wortman, J.R., Fraser-Liggett, C.M., Ravel, J., Rabinowicz, P.D.: Draft genome sequence of the oilseed species Ricinus communis. Nat. Biotechnol. **28**(9), 951–956 (2010)
36. Motamayor, J.C., Mockaitis, K., Schmutz, J., Haiminen, N., Livingstone, D., Cornejo, O., Findley, S.D., Zheng, P., Utro, F., Royaert, S., Saski, C., Jenkins, J., Podicheti, R., Zhao, M., Scheffler, B.E., Stack, J.C., Feltus, F.A., Mustiga, G.M., Amores, F., Phillips, W., Marelli, J.P., May, G.D., Shapiro, H., Ma, J., Bustamante, C.D., Schnell, R.J., Main, D., Gilbert, D., Parida, L., Kuhn, D.N.: The genome sequence of the most widely cultivated cacao type and its use to identify candidate genes regulating pod color. Genome Biol. **14**(6), r53 (2012)
37. Jaillon, O., Aury, J.-M., Noel, B., Policriti, A., Clepet, C., Casagrande, A., Choisne, N., Aubourg, S., Vitulo, N., Jubin, C., Vezzi, A., Legeai, F., Hugueney, P., Dasilva, C., Horner, D., Mica, E., Jublot, D., Poulain, J., Bruyère, C., Billault, A., Segurens, B., Gouyvenoux, M., Ugarte, E., Cattonaro, F., Anthouard, V., Vico, V., Del Fabbro, C., Alaux, M., Di Gaspero, G., Dumas, V., Felice, N., Paillard, S., Juman, I., Moroldo, M., Scalabrin, S., Canaguier, A., Le Clainche, I., Malacrida, G., Durand, E., Pesole, G., Laucou, V., Chatelet, P., Merdinoglu, D., Delledonne, M., Pezzotti, M., Lecharny, A., Scarpelli, C., Artiguenave, F., Pè, M.E., Valle, G., Morgante, M., Caboche, M., Adam-Blondon, A.-F., Weissenbach, J., Quetier, F., Wincker, P.: The grapevine genome sequence suggests ancestral hexaploidization in major angiosperm phyla. Nature **449**(7161), 463–467 (2007)
38. Lechner, M., Hernandez-Rosales, M., Doerr, D., Wieseke, N., Thvenin, A., Stoye, J., Hartmann, R.K., Prohaska, S.J., Stadler, P.F.: Orthology detection combining clustering and synteny for very large datasets. PLoS ONE **9**(8), e10515 (2014)
39. Yang, Z., Sankoff, D.: Natural parameter values for generalized gene adjacencies. J. Comput. Biol. **17**(9), 1113–1128 (2010)
40. Delgado, J., Lynce, I., Manquinho, V.: Computing the summed adjacency disruption number between two genomes with duplicate genes. J. Comput. Biol. **17**(9), 1243–1265 (2010)

Fast Phylogenetic Biodiversity Computations Under a Non-uniform Random Distribution

Constantinos Tsirogiannis(✉) and Brody Sandel

MADALGO and Department of Bioscience, Aarhus University, Aarhus, Denmark
constant@cs.au.dk, brody.sandel@bios.au.dk

Abstract. Computing the phylogenetic diversity of a set of species is an important part of many ecological case studies. More specifically, let $\mathcal{T}$ be a phylogenetic tree, and let R be a subset of its leaves representing the species under study. Specialists in ecology want to evaluate a function $f(\mathcal{T}, R)$ (a *phylogenetic measure*) that quantifies the evolutionary distance between the elements in R. But, in most applications, it is also important to examine how $f(\mathcal{T}, R)$ behaves when R is selected at random. The standard way to do this is to compute the mean and the variance of f among all subsets of leaves in $\mathcal{T}$ that consist of exactly $|R| = r$ elements. For certain measures, there exist algorithms that can compute these statistics, under the condition that all subsets of r leaves are equiprobable. Yet, so far there are no algorithms that can do this exactly when the leaves in $\mathcal{T}$ are weighted with unequal probabilities. As a consequence, for this general setting, specialists try to compute the statistics of phylogenetic measures using methods which are both inexact and very slow.

We present for the first time exact and efficient algorithms for computing the mean and the variance of phylogenetic measures when leaf subsets of fixed size are selected from $\mathcal{T}$ under a non-uniform random distribution. In particular, let $\mathcal{T}$ be a tree that has n nodes and depth d, and let r be a non-negative integer. We show how to compute in $O((d+\log n)n\log n)$ time and $O(n)$ space the mean and the variance for any measure that belongs to a well-defined class. We show that two of the most popular phylogenetic measures belong to this class: the Phylogenetic Diversity (PD) and the Mean Pairwise Distance (MPD). The random distribution that we consider is the Poisson binomial distribution restricted to subsets of fixed size r. More than that, we provide a stronger result; specifically for the PD and the MPD we describe algorithms that compute in a batched manner the mean and variance on $\mathcal{T}$ for *all* possible leaf-subset sizes in $O((d+\log n)n\log n)$ time and $O(n)$ space.

For the PD and MPD, we implemented our algorithms that perform batched computations of the mean and variance. We also developed alternative implementations that compute in $O((d+\log n)n^2)$ time the same output. For both types of implementations, we conducted experiments and measured their performance in practice. Despite the difference in

MADALGO—Center for Massive Data Algorithmics, a Center of the Danish National Research Foundation.

M. Singh (Ed.): RECOMB 2016, LNBI 9649, pp. 225–236, 2016.
DOI: 10.1007/978-3-319-31957-5_16

the theoretical performance, we show that the algorithms that run in $O((d+\log n)n^2)$ time are more efficient in practice, and numerically more stable. We also compared the performance of these algorithms with standard inexact methods that can be used in case studies. We show that our algorithms are outstandingly faster, making it possible to process much larger datasets than before. Our implementations will become publicly available through the R package `PhyloMeasures`.

1 Introduction

One of the most important aspects of biological diversity is phylogenetic diversity. Given a certain set of species, ecologists often want to know whether these species are close or distant phylogenetic relatives [13]. This is relevant in estimating the importance of conserving some of these species, and can provide insight into the ecological mechanisms that form species communities [6].

To measure the phylogenetic diversity between a set of species, biologists use the following process: first, they choose a phylogenetic tree $\mathcal{T}$ where the examined set of species are represented by a subset of leaf nodes R. The next step is to evaluate a function $f(\mathcal{T}, R)$ which measures the distance between the nodes in R. We call such a function a *phylogenetic measure*. Two very popular phylogenetic measures are the Phylogenetic Diversity (PD) and the Mean Pairwise Distance (MPD–see Sect. 2 for their formal definition). Whichever measure f we may use, in most applications it is not enough to compute only the value $f(\mathcal{T}, R)$ for the examined set R; we also have to check if $f(\mathcal{T}, R)$ is significantly different than the value of f on a randomly selected leaf subset in $\mathcal{T}$ that has the same size as R. To measure this, ecologists calculate an index that is defined as follows:

$$\mathrm{FI} = \frac{f(\mathcal{T}, R) - \mu_r(f, \mathcal{T})}{\sqrt{\mathrm{var}_r(f, \mathcal{T})}},$$

where $r = |R|$ is the number of elements in R, value $\mu_r(f, \mathcal{T})$ is the mean value of f among all leaf subsets in $\mathcal{T}$ that consist of r elements, and $\mathrm{var}_r(f, \mathcal{T})$ is the variance of f among these subsets. Therefore, to calculate index FI we need to compute the mean and variance of f over all leaf subsets of size r. Of course, the value of these two statistical moments depends on the probability distribution that we use to select the leaf subsets. There exist several algorithms that efficiently calculate these moments For the special case that leaves are selected with equal probability [8,12].

Yet, in many cases biologists want to incorporate that certain species are more abundant than others. This can be done by assigning to each leaf node v in $\mathcal{T}$ a probability value $p(v)$; this value represents the abundance in nature of the species represented by v. In this setting, the goal is to compute $\mu_r(f, \mathcal{T})$ and $\mathrm{var}_r(f, \mathcal{T})$ so that these reflect the probability values associated with the leaves in $\mathcal{T}$. We call these values the *weighted* moments of f. Computing the weighted moments requires also to define a probability distribution that assigns

to each subset of r leaves a probability of selection, based on the individual leaf probabilities. However, so far there does not exist any approach that shows how to do this exactly and efficiently. Faller *et al.* provide results for the PD measure, yet they consider a more relaxed model where the mean and the variance are computed among leaf subsets of unequal size [3].

In the absence of an exact solution, ecologists calculate the weighted moments of phylogenetic measures using an inexact approach. According to this approach, several leaf subsets (usually around a thousand) are selected at random from $\mathcal{T}$. Then, f is calculated for each of these subsets, and the mean and variance of f are approximated based on these calculated values. If r and the number of leaves in $\mathcal{T}$ are sufficiently large, this approach does not guarantee a good approximation of the weighted moments. Even worse, these methods can be very slow since they require to select and process a large number of samples. Therefore, there is a need for an exact and also efficient method that calculates the weighted moments of a phylogenetic measure.

Our Results. Inspired by the above, we present for the first time exact and efficient algorithms for computing the weighted moments of phylogenetic measures for leaf subsets of fixed size. Let $\mathcal{T}$ be a tree with n nodes and depth d, and let r be a non-negative integer. We show that we can compute in $O((d+\log n)n\log n)$ time and $O(n)$ space the mean and the variance for any measure that belongs to a certain class. We call the measures of this class *edge-decomposable* measures. We show that both PD and MPD belong to this class. For the algorithms that we propose, we calculate the mean and variance of an edge-decomposable measure based on the following distribution; leaf-subsets are conceptually selected using the Poisson binomial distribution (where each leaf v is picked in a Bernoulli trial with probability $p(v)$), and then a resulting subset is accepted only if it consists of exactly r elements. Specifically for the PD and MPD, we yield a stronger result; we present algorithms that compute in a batched manner the weighted moments of these measures for many leaf-subset sizes. These algorithms compute the mean and variance for *all* possible leaf-subset sizes in $O((d+\log n)n\log n)$ time.

We implemented the algorithms that perform batched computations of the weighted moments for PD and MPD. We developed two kinds of implementations; we implemented the aforementioned algorithms that run in $O((d+\log n)n\log n)$ time, but also algorithms that run in $O((d+\log n)n^2)$ time and are numerically more stable. For both types of implementations, we conducted experiments and measured their performance in practice. Despite the difference in the theoretical performance, we show that the algorithms that run in $O((d+\log n)n^2)$ time perform better in practice. The latter algorithms are highly parallelisable; with simple adjustments we were able to boost their performance and process fast very large phylogenies. For a tree of 71,181 leaves, our implementations computed the weighted moments for all 71,181 subset sizes in less than two and a half minutes for the PD, and in less than six minutes for the MPD. We compared the performance of the latter implementations with a program that estimates the weighted moments based on an inexact sampling method. In this comparison

our algorithms were found to be remarkably faster, making it possible to process much larger datasets than before. We intend to make our implementations publicly available through the R package `PhyloMeasures` [11].

2 Definitions and Notation

Notation Related to Phylogenetic Trees. Let $\mathcal{T}$ be a phylogenetic tree. We use E to denote the edges of $\mathcal{T}$, and for any edge $e \in E$ we use $w(e)$ to represent the weight of e. We indicate the set of nodes in $\mathcal{T}$ by V, and we indicate the set of leaf nodes in this tree by S. We use n to represent the total number of nodes in $\mathcal{T}$, and we use s to indicate the number of leaves in $\mathcal{T}$. We consider that $\mathcal{T}$ is a rooted tree; in the case of unrooted trees, we pick an arbitrary node which is not a leaf, and consider this node as the root. We define the depth of $\mathcal{T}$ as the maximum number of edges that appear on a simple path between the root of $\mathcal{T}$ and a leaf. We consider that the maximum degree in $\mathcal{T}$ (the maximum number of nodes adjacent to a single node of $\mathcal{T}$) is upper-bounded by a constant. Except the root, all other internal nodes in $\mathcal{T}$ have degree greater than two. The results described in this work can be easily extended to trees of non-constant maximum degree, by converting such a tree into a binary one in $O(n)$ time. Let R be a subset of $|R| = r$ leaves in $\mathcal{T}$, and let e be an edge in E. We use $S_R(e)$ to denote the set of leaves in R which appear in the subtree of e. We indicate the number of these leaves by $sr(e)$. We represent the (unique) minimum-size subtree of $\mathcal{T}$ that spans all the leaves in R by $\mathcal{T}(R)$. Let $u, v \in S$ be two leaves in $\mathcal{T}$ and let π be the simple path that connects these leaves. We define the *cost* of π as the sum of the weights of the edges that appear on this path. We represent this cost as $\text{cost}(u, v)$.

Phylogenetic Measures. Two of the most popular phylogenetic measures are the Phylogenetic Diversity (PD) and the Mean Pairwise Distance (MPD). The value of the PD for R is equal to:

$$\text{PD}(\mathcal{T}, R) = \sum_{e \in \mathcal{T}(R)} w(e).$$

Hence, the value of the PD for R is the cost of the minimum-size subtree $\mathcal{T}(R)$ in $\mathcal{T}$ that spans all leaves in R.

Let r be the number of elements in R. The MPD of R is equal to the average cost of a simple path in $\mathcal{T}$ that connects any two distinct leaves in R. More formally:

$$\text{MPD}(\mathcal{T}, R) = \frac{2}{r(r-1)} \sum_{\{u,v\} \in R} \text{cost}(u, v).$$

Probability Distribution. Let $\mathcal{T}$ be a phylogenetic tree, and let each leaf node $v \in S$ be associated with a probability value $p(v)$. We consider the following random process for selecting a subset of exactly r leaves; each leaf $v \in S$ is initially sampled independently with probability $p(v)$, and if the resulting subset R of sampled leaves consists of exactly r elements then we output R, otherwise we repeat the process. Therefore, the probability of selecting a subset which does not have exactly r elements is zero. The probability that a specific subset R of r leaves is selected according to this process is:

$$\frac{1}{C_r} \prod_{v \in R} p(v) \prod_{u \in S \setminus R} (1 - p(u)), \tag{1}$$

where C_r is a normalising constant equal to:

$$C_r = \sum_{\substack{G \subseteq S \\ |G|=r}} \prod_{x \in G} p(x) \prod_{y \in S \setminus G} (1 - p(y)). \tag{2}$$

In other words, the probability of selecting a specific subset R of r elements is equal to the probability of selecting R with n (non-identical) independent Bernoulli trials, divided by the sum of probabilities of all possible subsets of size r that are picked with such trials. The distribution that is entailed from this model is similar to the Poisson binomial distribution, restricted to subsets of fixed size r [2]. We call this constrained variant the *Restricted Poisson Binomial* distribution, or RPB for short. Note that in the RPB model the selection of two leaves u and v is not statistically independent; this is a consequence of considering only leaf subsets of the same size.

3 Description of Algorithms

3.1 Computing the Mean and Variance of Edge-Decomposable Measures

Let $\mathcal{T}$ be a phylogenetic tree of constant degree, and consider that every leaf v in this tree is assigned a probability value $p(v) \in [0, 1]$. We next focus on the problem of computing the expected value and the variance of phylogenetic measures when subsets of exactly r leaves are selected from $\mathcal{T}$ according to the RPB distribution. In particular, we examine this problem for phylogenetic measures that belong to a certain class. We call the measures of this class *edge-decomposable* measures. Intuitively, we call a measure edge-decomposable if for any input pair $\mathcal{T}$, R we can express $f(\mathcal{T}, R)$ as a sum of terms such that: each term corresponds to exactly one edge $e \in E$, and for each edge e the corresponding term can be evaluated in constant time if we know already the edge weight $w(e)$ and $sr(e)$ (the number of leaves in R appearing in the subtree of e). More formally, we can express this as follows:

Definition 1. *Let $\mathcal{T}$ be a phylogenetic tree and let R be a subset of r leaves in $\mathcal{T}$. A phylogenetic measure f is edge-decomposable if:*

(I) The value of f can be expressed as a sum of the form:

$$f(\mathcal{T}, R) = \sum_{e \in E} w(e) \cdot c(sr(e), r), \tag{3}$$

where c is a function whose definition does not depend on $\mathcal{T}$ or R. We call c the contribution *function of f.*

(II) Given values α and β, we can evaluate $c(\alpha, \beta)$ in constant time.

The proof of the following lemma appears in the full version of this paper [10].

Lemma 1. *Measures* PD *and* MPD *are edge-decomposable.*

Next we sketch an algorithm that computes the expected value of any edge-decomposable measure according to the RPB model. Let f be such a measure and let c be its contribution function. Let $\mathcal{T}$ be a tree such that each leaf $v \in S$ is assigned a probability value $p(v) \in [0, 1]$. Let $\mathbb{E}_r[f(\mathcal{T}, R)]$ represent the expected value of f among all subsets $R \subseteq S(\mathcal{T})$ of exactly r elements selected based on the RPB distribution. This expected value is equal to:

$$\mathbb{E}_r[f(\mathcal{T}, R)] = \sum_{e \in E} \sum_{i=0}^{s(e)} w(e) \cdot c(i, r) \cdot \mathbb{P}(sr(e) = i), \tag{4}$$

where $\mathbb{P}(sr(e) = i)$ is the probability that exactly i out of the r elements of a leaf subset fall in the subtree of e. This probability is equal to:

$$\mathbb{P}(sr(e) = i) = \frac{1}{C_r} \sum_{\substack{R \subseteq S \\ |R|=r \text{ and } sr(e)=i}} \prod_{v \in R} p(v) \prod_{u \in S \setminus R} (1 - p(u)), \tag{5}$$

where C_r is the normalising constant defined in Eq. (2). From (5) and (2), we observe that computing $\mathbb{P}(sr(e) = i)$ boils down to calculating two sums of products. To compute these sums efficiently, the key idea is to express them in terms of coefficients of certain polynomials. More specifically, let G be a subset of the leaves in $\mathcal{T}$. We define Pol_G to be the following univariate polynomial:

$$\mathrm{Pol}_G(x) = \prod_{v \in G} \left(p(v) \cdot x + (1 - p(v))\right).$$

Consider rewriting the above polynomial as a sum of the form $\sum_j a_j x^j$. We call this sum the *summation* representation of Pol_G. In this representation, consider the coefficient of the k-th power of x. We indicate this coefficient by $\mathrm{CF}(G, k)$. This is equal to:

$$\mathrm{CF}(G, k) = \sum_{\substack{R \subseteq G \\ |R|=k}} \prod_{v \in R} p(v) \prod_{u \in G \setminus R} (1 - p(u)).$$

Based on this observation, it follows directly that $C_r = \mathrm{CF}(S, r)$. More than that, we can express the probability value $\mathbb{P}(sr(e) = i)$ by rewriting (5) as follows:

$$\begin{aligned}\mathbb{P}(sr(e) = i) &= \frac{1}{C_r} \sum_{\substack{R \subseteq S \\ |R|=r \text{ and } sr(e)=i}} \prod_{v \in R} p(v) \prod_{u \in S \setminus R} (1 - p(u)) \\ &= \frac{1}{C_r} \sum_{\substack{G \subseteq S(e) \\ |G|=i}} \prod_{v \in G} p(v) \prod_{u \in S(e) \setminus G} (1 - p(u)) \sum_{\substack{M \subseteq S \setminus S(e) \\ |M|=r-i}} \prod_{y \in M} p(y) \prod_{z \in (S \setminus (S(e) \cup M))} (1 - p(z)) \\ &= \frac{\mathrm{CF}(S(e), i) \cdot \mathrm{CF}(S \setminus S(e), r - i)}{\mathrm{CF}(S, r)}. \end{aligned} \tag{6}$$

Hence, to compute value $\mathbb{P}(sr(e) = i)$ it suffices to construct the summation representations of polynomials $\mathrm{Pol}_{S(e)}$, $\mathrm{Pol}_{S \setminus S(e)}$, and Pol_S, and then extract the required coefficients.

Let G be a subset of leaves $G \in S$, consisting of g leaves. We can compute Pol_G in $O(g \log^2 g)$ time in the following manner: first we construct all binomials $p(v) \cdot x + (1 - p(v))$ such that $v \in G$. Then, we partition the set of these polynomials into pairs, and we multiply the elements of each pair using the Fast Fourrier Transform (FFT) [7]. We repeat this process on the resulting polynomials, until we end up with a single polynomial. Multiplying two polynomials of maximum degree k takes $O(k \log k)$ time with the FFT. At each repetition, the sum of the degrees of the processed polynomials is equal to g, and we perform $O(\log g)$ repetitions in total; this leads to $O(g \log^2 g)$ time for the entire algorithm.

Using the above process, and based on (4) and (6), we could consider the following approach to compute the mean of f: first we compute $C_r = \mathrm{CF}(S, r)$ by constructing Pol_S in $O(n \log^2 n)$ time and extracting the r-th coefficient. Then, for each edge $e \in \mathcal{T}$ we construct from scratch polynomials $\mathrm{Pol}_{S(e)}$ and $\mathrm{Pol}_{S \setminus S(e)}$, and use the coefficients of these polynomials to compute values $\mathbb{P}(sr(e) = i)$ for every integer $i \in [0, s(e)]$. The described approach would require in total $O(n^2 \log^2 n)$ time; for every edge e we need to spend $O(n \log^2 n)$ time to construct $\mathrm{Pol}_{S(e)}$ and $\mathrm{Pol}_{S \setminus S(e)}$ since one of these polynomials has degree which is at least $n/2$. Yet, we can design an algorithm which is more efficient when $\mathcal{T}$ is relatively balanced. In fact, we can achieve a similar result not only for the mean, but also for the variance of f. We provide the next lemma.

Lemma 2. *Let $\mathcal{T}$ be a phylogenetic tree that has n nodes and depth d, and let f be an edge-decomposable measure. Let r be a non-negative integer. We can compute the mean and variance of f for leaf-subsets of size r according to the* RPB *distribution in $O((d + \log n) n \log n)$ time, using $O(n)$ space.*

3.2 Batched Computations for the Moments of Popular Measures

So far, we showed that we can efficiently compute the mean and the variance for any edge-decomposable measure f; we showed that this can be done for a single subset size r in $O((d + \log n) n \log n)$ time according to the RPB model. The

described algorithms are generic, and work for any edge-decomposable measure f; these algorithms use the contribution function of f as a black box for their computations, and they do not take into account how this function is defined. This gives rise to the following question; given a specific phylogenetic measure, can we derive more efficient algorithms which are especially designed for this measure? Indeed, in the rest of this section we show that we can do this for two popular measures; these are the PD and MPD. In particular, let $\mathcal{T}$ be a tree that has s leaves and n nodes in total. For each of these measures we provide algorithms that calculate the mean and the variance in $O((d+\log n)n \log n)$ time for *all* subset sizes in the range $\{1, 2, \ldots, s\}$. That means, instead of spending $O((d + \log n)n \log n)$ time for computing the statistical moments of a measure for a single subset size r, we can compute the moments for all s possible leaf subset sizes in asymptotically the same time. We state formally our results for PD and MPD in the following theorem. The proof of the theorem appears in the full paper.

Theorem 1. *Let $\mathcal{T}$ be a phylogenetic tree that has n nodes and depth d. We can compute the mean and the variance of the* PD *and* MPD *on $\mathcal{T}$ in the* RPB *model for all possible leaf-subset sizes in $O((d + \log n)n \log n)$ time in total. The space required for these computations is $O(n)$. In a more general setting, let $A(k)$ be the time complexity of a polynomial multiplication algorithm when applied on two polynomials of maximum degree k. We can compute the mean and the variance of the* PD *and* MPD *on $\mathcal{T}$ in the* RPB *model for all possible leaf-subset sizes in $O((d + \log n)A(n))$ time in total, using $O(n)$ space.*

4 Implementations and Experiments

Based on our theoretical results, we implemented algorithms that compute in a batched manner the mean and variance of the PD and MPD. More specifically, for each of these two measures we implemented two algorithms; the first algorithm for each measure computes the mean and the variance for all leaf-subset sizes in $O((d+\log n)n \log n)$ operations and uses the FFT. The second algorithm implemented for each measure computes the same output while performing multiplications of polynomials in a naive manner; the multiplication subroutine that we use here takes $O(k^2)$ time to compute the product of two polynomials of maximum degree k. Therefore, and according to Theorem 1, the running time of the latter implementation is $O((d+\log n)n^2)$. We refer to the implementations that make use of the FFT as `fft_pd` and `fft_mpd`. We call these the FFT-*b*ased implementations. We refer to the other two implementations as `naive_pd` and `naive_mpd`. We call those two the *naive-based* implementations. All described implementations were developed in C++, and for each implementation we also developed an interface to R.

We conducted experiments on our implementations using two different phylogenetic tree datasets. The first dataset is a phylogenetic tree of all mammal species [1]. It has 4,510 leaf nodes, 6,618 nodes in total, and depth 39. To speed

up our algorithms, we converted this tree into a binary one by adding extra interior nodes and edges of zero weight. This can be performed in $O(n)$ time, and maintains the same mean and variance as for the initial one for any edge-decomposable measure. The resulting tree has 9,019 nodes, and depth 42. We refer to the latter tree as `mammals`. The second dataset is a tree that represents the phylogeny of eukaryotic organisms [4]. This tree is unrooted, so we picked an internal node and used that as the root. We converted this tree into a binary one, and the resulting dataset has 142,361 nodes (71,181 leaves) and depth 86. We refer to the this tree as `eukaryotes`.

All experiments were performed on a 64-bit computer with an Intel i7-3770 CPU. This CPU consists physically of eight virtual cores where each core is a 3.40 GHz processor. The main memory of the computer is 16 Gigabytes, and the operating system installed on this computer is Ubuntu version 14.04.

In the first set of experiments, we measured the running time of all four implementations on subtrees that we extracted from `mammals`. We extracted eighteen trees from `mammals` such that each tree consisted of $250k + 260$ leaves, with k ranging from zero to seventeen. For each extracted tree, we ran our implementations and computed the mean and the variance for all possible leaf-subset sizes. The running times measured for this experiment are illustrated in Fig. 1. We see that the naive-based implementations outperform the FFT-based ones. Also, the implementations for the PD are faster than the ones for MPD (this is expected since for the MPD computations we need to construct more polynomials than for PD). For the largest tree that we used (the complete `mammals` tree with 4,510 leaves) implementation `fft_pd` took 10.6 s to compute the required moments, while `naive_pd` took 3.08. For the same tree, program `fft_mpd` took 24.83 s and `naive_mpd` took 7.83 s. We also repeated this experiment using an inexact method. More specifically, for a given subset-size r we select at random from `mammals` one thousand leaf-subsets of this size. For this we used function `sample` in software platform R; this function samples a subset from a weighted set of elements by sequentially picking an element t with probability $p(t)$ divided by the sum of probabilities of all remaining items. For each sample, the PD and MPD values where computed using the functions of package `PhyloMeasures`, and the mean and the variance of each measure was calculated from these values. We refer to this inexact method as the *heuristic*. The heuristic method took more than an hour to execute for this tree, both for the PD and the MPD. Even for the smallest tree that we considered in this experiment (260 leaves), the heuristic method took 19.22 and 26.92 s for the PD and MPD respectively.

This difference between the naive-based and FFT-based programs seems to contradict the theoretical complexity of the corresponding algorithms. One explanation is that the FFT has a larger constant hidden in its asymptotical time complexity than the naive multiplication algorithm. It is the case that the gap in performance becomes smaller as the tree size increases. Even so, preliminary experiments showed that also for much larger trees (such as the complete `eukaryotes`, with $> 70{,}000$ leaves) the FFT-based algorithms were slower.

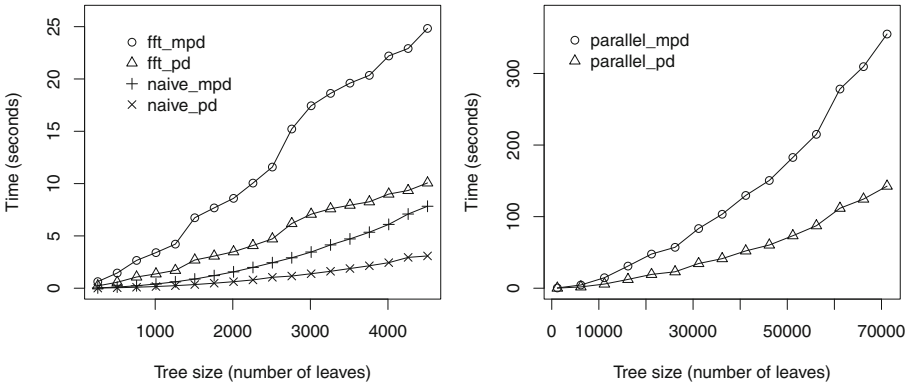


Fig. 1. Left: the measured running times of the FFT-based and naive-based implementations on trees extracted from `mammals` dataset. **Right:** the running times of the two parallelized implementations on trees extracted from `eukaryotes`.

Also, as already mentioned, the FFT is a numerically unstable method. The robustness of this method depends on the precision of evaluating trigonometric functions, to generate the so-called complex roots of unity (also known as "twindle" factors) [9]. As a result, for our FFT-based implementations and for all the datasets that we considered, almost the entire output was wrong. For this reason, and because of the difference in performance, we focused on improving the efficiency of the naive-based implementations. We describe this in more detail with the next set of experiments.

In the second set of experiments, we boosted the naive-based implementations by introducing parallelism. We did this by re-designing the naive operator that performs polynomial multiplications. More specifically, let P_α and P_β be two polynomials of maximum degree m, and let g denote the number of available processors. When computing the product $P_\gamma = P_\alpha \otimes P_\beta$, we split the computation of the at most $2m$ coefficients of P_γ into $2\,\mathrm{m/g}$ groups, and fed each group to a different processor. This simple adjustment leads to an algorithm whose time complexity is $O((d+\log n)(n^2/g+n\log n))$. We made this adjustment to both of our naive-based methods. We refer to these adjusted implementations as `parallel_pd` and `parallel_mpd`. We evaluated the running time of the parallelized implementations on trees extracted from the `eukaryotes` dataset. These trees consist of $5000k + 1181$ leaves, with k ranging from zero to fourteen. For these experiments, at any point during the executions, the maximum number of active parallel threads was set to eight, the number of available processors on our computer. The results of the experiments appear in Fig. 1.

We see that both implementations perform very fast even for datasets with many thousands of leaves. For the complete `eukaryotes` tree, `parallel_pd` took 143 s to execute, and `parallel_mpd` took 355 s. Recall that, during these executions, each program calculated the mean and the variance for as many leaf-subset sizes as the number of leaves in the tree.

In the third set of experiments we demonstrate how our implementations can speedup the computations for standard case studies in Ecology. To do this, we used the `mammals` tree and a dataset that represents mammal communities around the world. The community dataset that we used is a presence-absence matrix which is structured as follows; each column of the matrix corresponds to a mammal species, and each row corresponds to a geographical region. Each entry M_{ij} in the matrix has value one or zero, based on whether the i-th species resides in the region represented by the j-th row.

To construct the matrix, we produced a raster of the world with a resolution of 193 km. Then, we used polygons that represent the geographical areas where mammal species reside. These polygons were acquired from the International Union for Conservation of Nature (IUCN) [5]. We overlayed the polygons on the world raster, and extracted for each cell the community of species whose polygons overlap with the cell. We then represented each extracted community as a separate row in the presence-absence matrix. We refer to the resulting matrix as `communities`. The `communities` matrix consists of 4,971 rows and 4,173 columns. The number of columns in this matrix is smaller than the number of leaves in `mammals` because we excluded marine mammals, and because data was absent for a few of the other species. To each species v in `mammals` we assigned a probability value $p(v)$ equal to the sum of entries in the corresponding column of `communities`, divided by the total number of rows in the matrix. The species that were not represented by a column in `communities` were assigned a probability value equal to zero.

For each species set R in `communities` we used our implementations to compute an index based on the MPD. This index is called the reverse-Net Relatedness Index (rNRI), the reverse of the NRI index [13]. For a species set R of r species this is equal to: $\text{rNRI}(\mathcal{T}, R) = \frac{\text{MPD}(\mathcal{T},R)-\mu_r(\mathcal{T})}{\sqrt{\text{var}_r(\mathcal{T})}}$, where $\mathcal{T}$ is the phylogenetic tree, $\mu_r(\text{MPD}, \mathcal{T})$ is the mean value of MPD among all leaf subsets of r elements, and $\text{var}_r(\text{MPD}, \mathcal{T})$ is the variance of MPD among these subsets. The rNRI values were computed using the interface of our implementations in R. To calculate $\text{MPD}(\mathcal{T}, R)$ for a single subset R we used the efficient implementation of this measure that appears in the R package `PhyloMeasures`. Figure 2 shows the world grid colored according to the computed rNRI values (observe that

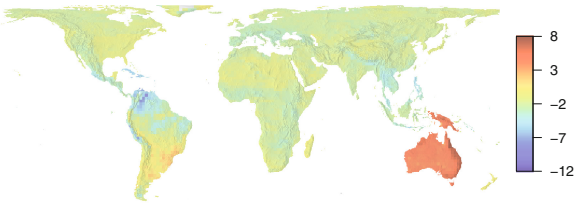


Fig. 2. The world raster of 193 Km resolution, colored according to the rNRI values of mammal communities. Red regions represent areas of higher relative phylogenetic biodiversity, while blue regions indicate lower diversity.

Australia shows remarkably high rNRI values, due to mixing of many marsupial and placental mammal lineages, leading to high pairwise distances; in contrast, northern South America shows particularly low rNRI values, indicating a concentration of closely related lineages).

Using `parallel_mpd`, the time taken to compute all the rNRI values is 2.6 s. The time taken to compute these values with `naive_mpd` is 8.5 s. Using the heuristic method, it took more than 65 min and 58 s to compute the rNRI values for all the species sets in `communities`. Comparing this with the performance of our methods, we conclude that our algorithms provide a huge speedup for standard applications in Ecology. This allows to process much larger datasets than it was possible before. We intend to incorporate our parallelized implementations to the R package `PhyloMeasures`.

References

1. Bininda-Emonds, O.R.P., Cardillo, M., Jones, K.E., MacPhee, R.D.E., Beck, R.M.D., Grenyer, R., Price, S.A., Vos, R.A., Gittleman, J.L., Purvis, A.: The delayed rise of present-day mammals. Nature **446**, 507–512 (2007)
2. Chen, S.X., Liu, J.S.: Statistical applications of the poisson-binomial and conditional bernoulli distributions. Stat. Sin. **7**, 875–892 (1997)
3. Faller, B., Pardi, F., Steel, M.: Distribution of phylogenetic diversity under random extinction. J. Theor. Biol. **251**, 286–296 (2008)
4. Goloboff, P.A., Catalano, S.A., Mirandeb, J.M., Szumika, C.A., Ariasa, J.S., Kallersjoc, M., Farris, J.S.: Phylogenetic analysis of 73 060 taxa corroborates major eukaryotic groups. Cladistics **25**, 211–230 (2009)
5. The Webpage of the International Union for Conservation of Nature. http://www.iucn.org/
6. Kraft, N.J.B., Cornwell, W.K., Webb, C.O., Ackerly, D.A.: Trait evolution, community assembly, and the phylogenetic structure of ecological communities. Am. Nat. **170**, 271–283 (2007)
7. Van Loan, C.: Computational Frameworks for the Fast Fourier Transform, vol. 10. Siam, Philadelphia (1992)
8. Steel, M.: Tools to construct, study big trees: A mathematical perspective. In: Hodkinson, T., Parnell, J., Waldren, S. (eds.) Reconstructing the Tree of Life: Taxonomy and Systematics of Species Rich Taxa, pp. 97–112. CRC Press, Boca Raton (2007)
9. Tasche, M., Zeuner, H.: Improved roundoff error analysis for precomputed twiddle factors. J. Comp. Anal. Appl. **4**(1), 1–18 (2002)
10. Tsirogiannis, C., Sandel, B.: Fast Phylogenetic Biodiversity Computations Under a Non-Uniform Random Distribution. http://www.madalgo.au.dk/~constant/abundance_model.pdf
11. Tsirogiannis, C., Sandel, B.: PhyloMeasures: a Package for Computing Phylogenetic Biodiversity Measures and their Statistical Moments. Ecography (2015). http://dx.doi.org/10.1111/ecog.01814
12. Tsirogiannis, C., Sandel, B., Kalvisa, A.: New algorithms for computing phylogenetic biodiversity. In: Brown, D., Morgenstern, B. (eds.) WABI 2014. LNCS, vol. 8701, pp. 187–203. Springer, Heidelberg (2014)
13. Webb, C.O., Ackerly, D.D., McPeek, M.A., Donoghue, M.J.: Phylogenies and community ecology. Annu. Rev. Ecol. Syst. **33**, 475–505 (2002)

Short Abstracts

SLICER: Inferring Branched, Nonlinear Cellular Trajectories from Single Cell RNA-seq Data

Joshua D. Welch[1], Ziqing Liu[2], Li Wang[2], Junjie Lu[3], Paul Lerou[3], Jeremy Purvis[4], Li Qian[2], Alexander Hartemink[5], and Jan F. Prins[1]

[1] Department of Computer Science, The University of North Carolina at Chapel Hill, Chapel Hill, USA
{jwelch,prins}@cs.unc.edu
[2] Department of Pathology, The University of North Carolina at Chapel Hill, Chapel Hill, USA
[3] Department of Pediatric Newborn Medicine, Harvard Medical School, Boston, USA
[4] Department of Genetics, The University of North Carolina at Chapel Hill, Chapel Hill, USA
[5] Department of Computer Science, Duke University, Durham, USA

1 Abstract

Understanding the dynamic regulation of gene expression in cells requires the study of important temporal processes, such as differentiation, the cell division cycle, or tumorigenesis. However, in such cases, the precise sequence of changes is generally not known, few if any marker genes are available, and individual cells may proceed through the process at different rates. These factors make it very difficult to judge a given cell's position within the process. Additionally, bulk RNA-seq data may blur aspects of the process because cells at sampled at a given wallclock time may be at differing points along the process. The advent of single cell RNA-seq enables study of sequential gene expression changes by providing a set of time slices or "snapshots" from individual moments in the process. To combine these snapshots into a coherent picture, we need to infer an "internal clock" that tells, for each cell, where it is in the process.

Several techniques, most notably Monocle and Wanderlust, have recently been developed to address this problem. Monocle and Wanderlust have both been successfully applied to reveal biological insights about cells moving through a biological process. However, a number of aspects of the trajectory construction problem remain unexplored. For example, both Monocle and Wanderlust assume that the set of expression values they receive as input have been curated in some way using biological prior knowledge. Wanderlust was designed to work on data from protein marker expression, a situation in which the number of markers is relatively small (dozens, not hundreds of markers) and the markers are hand-picked based on prior knowledge of their involvement in the process. In the initial application of Monocle, genes were selected based on differential expression analysis of bulk RNA-seq data collected at initial and final timepoints. In addition, Monocle uses ICA, which assumes that the trajectory lies along a linear projection of the data. In general, this linearity assumption may

M. Singh (Ed.): RECOMB 2016, LNBI 9649, pp. 239–240, 2016.
DOI: 10.1007/978-3-319-31957-5

not hold in biological systems. In contrast, Wanderlust can capture nonlinear trajectories, but works in the original high-dimensional space, which may make it more susceptible to noise, particularly when given thousands of genes, many of which are unrelated to the process being studied. Another challenging aspect of trajectory construction is the detection of branches. For example, a developmental process may give rise to multiple cell fates, leading to a bifurcation in the manifold describing the process. Wanderlust assumes that the process is non-branching when constructing a trajectory. Monocle provides the capability of dividing a trajectory into a branches, but requires the user to specify the number of branches.

In this paper, we present SLICER (Selective Locally linear Inference of Cellular Expression Relationships), a new approach that uses locally linear embedding (LLE) to reconstruct cellular trajectories. SLICER provides four significant advantages over existing methods for inferring cellular trajectories: (1) the ability to automatically select genes to use in building a cellular trajectory with no need for biological prior knowledge; (2) use of locally linear embedding, a nonlinear dimensionality reduction algorithm, for capturing highly nonlinear relationships between gene expression levels and progression through a process; (3) automatic detection of the number and location of branches in a cellular trajectory using a novel metric called geodesic entropy; and (4) the capability to detect types of features in a trajectory such as "bubbles" that no existing method can detect. Comparisons using synthetic data show that SLICER outperforms existing methods, particularly when given input that includes genes unrelated to the trajectory. We demonstrate the effectiveness of SLICER on newly generated single cell RNA-seq data from human embryonic stem cells and murine induced cardiomyocytes.

Multi-track Modeling for Genome-Scale Reconstruction of 3D Chromatin Structure from Hi-C Data

Chenchen Zou[1,8], Yuping Zhang[2,3,4,5,8], and Zhengqing Ouyang[1,3,6,7,8]

[1] The Jackson Laboratory for Genomic Medicine, Farmington, CT, USA
zhengqing.ouyang@jax.org
[2] Department of Statistics, University of Connecticut, Storrs, CT, USA
[3] Institute for Systems Genomics, University of Connecticut, Storrs, CT, USA
[4] Institute for Collaboration on Health, Intervention, and Policy, University of Connecticut, Storrs, CT, USA
[5] Center for Quantitative Medicine, University of Connecticut Health Center, Farmington, CT, USA
[6] The Connecticut Institute for the Brain and Cognitive Sciences, University of Connecticut, Storrs, CT, USA
[7] Department of Biomedical Engineering, University of Connecticut, Storrs, CT, USA
[8] Department of Genetics and Genome Sciences, University of Connecticut Health Center, Farmington, CT, USA

Abstract. Genome-wide chromosome conformation capture (Hi-C) has been widely used to study chromatin interactions and the 3D structures of the genome. However, few computational approaches are existing to quantitatively analyze Hi-C data, thus hindering the investigation of the association between 3D chromatin structure and genome function. Here, we present HSA, a novel approach to reconstruct 3D chromatin structures at the genome-scale by modeling multi-track Hi-C data. HSA models chromatin as a Markov chain under a generalized linear model framework, and uses simulated annealing to globally search for the latent structure underlying the cleavage footprints of different restriction enzymes. HSA is robust, accurate, and outperforms or rivals existing computational tools when evaluated on simulated and real datasets in diverse cell types.

Keywords: 3D chromatin structure · Multi-track modeling · Genome-wide · Hi-C

M. Singh (Ed.): RECOMB 2016, LNBI 9649, p. 241, 2016.
DOI: 10.1007/978-3-319-31957-5

Revealing the Genetic Basis of Immune Traits in the Absence of Experimental Immunophenotyping

Yael Steuerman and Irit Gat-Viks

Department of Cell Research and Immunology, The George S. Wise Faculty of Life Sciences, Tel Aviv University, 6997801 Tel Aviv, Israel
iritgv@post.tau.ac.il

Introduction and Motivation. The immune system consists of hundreds of immune cell types working coordinately to maintain tissue homeostasis. Thus, discovering the genetic control underlying inter-individual variation in the abundance of immune cell subpopulations requires simultaneous quantification of numerous immune cell types. Current experimental technologies, such as fluorescence-activated cell sorting (FACS) [1], can follow the dynamics of only a limited number of cell types, hence hindering a comprehensive analysis of the full genetic complexity of immune cell quantities. One possible way to attain a global immunophenotyping is to mathematically infer, by means of a deconvolution technique [2–5], the abundance of a variety of immune cell subpopulations based on gene-expression profiles from a complex tissue, without the need of direct cell sorting measurements. Based on these predicted immunophenotypes, a genome-wide association study can be applied to uncover the genetic basis for these immune traits.

Methods. We developed a novel computational methodology to identify significant associations between immune traits and polymorphic DNA loci. Our method combines (i) prior knowledge on the transcriptional profile of various immune cell-types; (ii) gene-expression data in a given cohort of individuals; and (iii) genotyping data of the same individuals. Our method utilizes a deconvolution method which computationally infers the global dynamics of immune cell subsets for each individual. Specifically, we exploit associations between cell types, genes and genotypes to select an informative group of marker genes, rather than the full transcriptional profile, to attain a more accurate deconvolution-based model.

Results. We applied our method to both synthetic and real biological data to evaluate its ability to uncover the genetic basis of immune traits. Our analysis of synthetic data confirms that our method can handle non-conventional artifacts and outperforms the standard approach. Overall, the methodology presented is general and can be applied using various deconvolution tools and in the context of various biological applications, in both human and mouse.

Acknowledgments. This work was supported by the European Research Council (637885) (Y.S., I.G-V), Broad-ISF program (1168/14) (Y.S.) and the Edmond J. Safra Center for Bioinformatics at Tel Aviv University (Y.S.). Research in the I.G-V. lab is supported by the Israeli

M. Singh (Ed.): RECOMB 2016, LNBI 9649, pp. 242–243, 2016.
DOI: 10.1007/978-3-319-31957-5

Science Foundation (1643/13) and the Israeli Centers of Research Excellence (I-CORE): Center No. 41/11. I.G-V. is a Faculty Fellow of the Edmond J. Safra Center for Bioinformatics at Tel Aviv University and an Alon Fellow.

References

1. Ibrahim, S.F., Van Den Engh, G.: Flow cytometry and cell sorting. In: Advances in Biochemical Engineering/Biotechnology, pp. 19–39 (2007)
2. Abbas, A.R., Wolslegel, K., Seshasayee, D., Modrusan, Z., Clark, H.F.: Deconvolution of blood microarray data identifies cellular activation patterns in systemic lupus erythematosus. PLoS One **4**, 16 (2009)
3. Altboum, Z., Steuerman, Y., David, E., Barnett-Itzhaki, Z., Valadarsky, L., Keren-Shaul, H., Meningher, T., Mendelson, E., Mandelboim, M., Gat-Viks, I., Amit, I.: Digital cell quantification identifies global immune cell dynamics during influenza infection. Mol. Syst. Biol. **10**, 720 (2014)
4. Newman, A.M., Liu, C.L., Green, M.R., Gentles, A.J., Feng, W., Xu, Y., Hoang, C.D., Diehn, M., Alizadeh, A.A.: Robust enumeration of cell subsets from tissue expression profiles. Nat. Methods **12**, 453–457 (2015)
5. Shen-Orr, S.S., Tibshirani, R., Khatri, P., Bodian, D.L., Staedtler, F., Perry, N.M., Hastie, T., Sarwal, M.M., Davis, M.M., Butte, A.J.: Cell type–specific gene expression differences in complex tissues. Nat. Methods **7**, 287–289 (2010)

Shall We Dense? Comparing Design Strategies for Time Series Expression Experiments

Emre Sefer and Ziv-Bar Joseph

Department of Computational Biology, School of Computer Science,
Carnegie Mellon University, Pittsburgh, USA
{esefer,zivbj}@cs.cmu.edu

Extended Abstract

Recent advances in sequencing technologies have enabled high throughout profiling of several types of molecular datasets including mRNAs, miRNAs, methylation, and more. Many studies profile one or more of these types of data in a time course. An important experimental design question in such experiments is the number of repeats that is required for accurate reconstruction of the signal being studied. While several studies examined this issue for *static* experiments much less work has focused on the importance of repeats for time series analysis.

Obviously, the more points that can be profiled between the start and end points, the more likely it is that the reconstructed trajectory is accurate. However, in practice the number of time points that are used is usually very small. The main limiting factor is often the budget. While technology has greatly improved, high-throughput NGS studies still cost hundreds of dollars per specific experiment. This is a major issue for time series studies, especially those that need to profile multiple types of biological datasets (mRNA, miRNAs, methylation etc.) at each selected point. Another issue that can limit the number of experiments performed (and so the total number of time points that can be used) is biological sample availability. Thus, when designing such experiments researchers often need to balance the overall goals of reconstructing the most accurate temporal representation of the data types being studied and the need to limit the number of experiments as discussed above.

Given these constraints, an important question when designing high-throughput time-series studies is the need for *repeat* experiments. On the one hand, repeats are a hallmark of biological experiments providing valuable information about noise and reliability of the measured values. On the other, as discussed above, repeats reduce the number of time points that can be profiled which may lead to missing key events between sampled points. Further, if we assume that the biological data being profiled can be represented by a (smooth) continuous curve, which is often the case, then the autocorrelation between successive points can also provide information about noise in the data. In such cases, more time points, even at the expense of fewer or no repeats, may prove to be a better strategy.

Indeed, when looking at datasets deposited in GEO (roughly 25 % of all GEO datasets are time-series), we observe that most of these do not use repeats.

M. Singh (Ed.): RECOMB 2016, LNBI 9649, pp. 244–245, 2016.
DOI: 10.1007/978-3-319-31957-5

However, to the best of our knowledge, no analysis to date was performed to determine the trade-offs between a dense sampling strategy (profiling more time points) and repeat sampling (profiling fewer points, with more than one experiment per point). To study this issue, we use both theoretical analysis and analysis of real data. In our theoretical analysis, we consider a large number of piecewise linear curves and noise levels and compare the expected errors when using the two sampling methods. While the profiles in these biological datasets are usually not piecewise linear, such curves represent important types of biological responses (for example, gradual or single activation, cyclic behavior, increase and then return to baseline, etc.). We also analyze time-series gene expression data to determine the performance of these strategies on real biological data.

Overall, for both, theoretical analysis when using reasonable noise levels and real biological data, we see that dense sampling outperforms repeat sampling indicating that for such data autocorrelation can indeed be a useful feature when trying to reduce the impact of noise on the reconstructed curves. Our results support the commonly used (though so far not justified) practice of reducing or eliminating repeat experiments in time-series high-throughput studies.

Supporting code and datasets: www.cs.cmu.edu/~esefer/genetheoretical

Enabling Privacy Preserving GWAS in Heterogeneous Human Populations

Sean Simmons[1,2], Cenk Sahinalp[2,3], and Bonnie Berger[1]

[1] Department of Mathematics and CSAIL, MIT, Cambridge, MA, USA
bab@mit.edu
[2] School of Computing Science, Simon Fraser University, Burnaby, BC, Canada
[3] School of Informatics and Computing, Indiana University, Bloomington, IN, USA

Extended Abstract

With the projected rise of genotyping in the clinic, there has been increasing interest in using patient data to perform genomewide association studies (GWAS) [5, 9]. The idea is to allow doctors and researchers to query patient electronic health records (EHR) to see which diseases are associated with which genomic alterations, avoiding the expense and time required to recruit and genotype patients for a standard GWAS. Such a system, however, leads to major privacy concerns for patients [6]. These privacy concerns have led to tight regulations over who can access this patient data–often it is limited to individuals who have gone through a time consuming application process.

Various approaches have been suggested for overcoming this bottleneck. Specifically, there has been growing interest in using a cryptographic tool known as differential privacy [2] to allow researchers access to this genomic data [3, 4, 8, 11, 12]. Previous approaches for performing differentially private GWAS are based on rather simple statistics that have some major limitations; in particular, they do not correct for a problem known as population stratification, something that is needed when dealing with the genetically diverse populations in many genetic databases. Population stratification is the name given to systematic genomic differences between human populations [10]. It turns out that these differences make it difficult for GWAS to find biologically meaningful associations between common alleles in the population and phenotypes. In order to avoid this problem, various methods have been suggested (EIGENSTRAT [7], LMMs [10], genomic control [1]).

In this work we focus on producing GWAS results that can handle population stratification while still preserving private phenotype information (namely disease status). In particular, we develop a framework that can turn commonly used GWAS statistics (such as LMM based statistics and EIGENSTRAT) into tools for performing privacy preserving GWAS. We demonstrate this framework on one such statistic, EIGENSTRAT [7]. Our method, denoted PrivSTRAT, uses a differentially private framework to protect private phenotype information (disease status) from being leaked while conducting GWAS. Importantly, ours is the first method able to correct for population stratification while preserving privacy in GWAS results. This advance introduces the possibility of applying a

M. Singh (Ed.): RECOMB 2016, LNBI 9649, pp. 246–247, 2016.
DOI: 10.1007/978-3-319-31957-5

differentially private framework to large, genetically diverse groups of patients (such as those present in EHR!).

We test the resulting differentially private EIGENSTRAT statistic, PrivSTRAT, on both simulated and real GWAS datasets to demonstrate its utility. Our results show that for many common GWAS queries, PrivSTRAT is able to achieve high accuracy while enforcing realistic privacy guarantees.

Implementation available at: http://groups.csail.mit.edu/cb/PrivGWAS.

References

1. Devlin, B., Roeder, K.: Genomic control for association studies. Biometrics **55**(4), 997–1004 (1999)
2. Dwork, C., Pottenger, R.: Towards practicing privacy. J. Am. Med. Inform. Assoc. **20**(1), 102–108 (2013)
3. Jiang, X., Zhao, Y., Wang, X., Malin, B., Wang, S., Ohno-Machado, L., Tang, H.: A community assessment of privacy preserving techniques for human genomes. BMC Med. Inform. Decis. Making 14(S1) (2014)
4. Johnson, A., Shmatikov, V.: Privacy-preserving data exploration in genome-wide association studies. In: KDD, pp. 1079–1087 (2013)
5. Lowe, H., Ferris, T., Hernandez, P., Webe, S.: STRIDE - an integrated standards-based translational research informatics platform. In: AMIA Annual Symposium Proceedings, pp. 391–395 (2009)
6. Murphy, S., Gainer, V., Mendis, M., Churchill, S., Kohane, I.: Strategies for maintaining patient privacy in I2B2. JAMIA **18**, 103–108 (2011)
7. Price, A., Patterson, N., Plenge, R., Weinblatt, M., Shadick, N., Reich, D.: Principal components analysis corrects for stratification in genome-wide association studies. Nat. Genet. **38**, 904–909 (2006)
8. Uhler, C., Fienberg, S., Slavkovic, A.: Privacy-preserving data sharing for genome-wide association studies. J. Priv. Confidentiality **5**(1), 137–166 (2013)
9. Weber, G., Murphy, S., McMurry, A., MacFadden, D., Nigrin, D., Churchill, S., Kohane, I.: The shared health research information network (SHRINE): A prototype federated query tool for clinical data repositories. JAMIA **16**, 624–630 (2009)
10. Yang, J., Zaitlen, N., Goddard, M., Visscher, P., Price, A.: Advantages and pitfalls in the application of mixed-model association methods. Nat. Genet. **46**(2), 100–106 (2014)
11. Yu, F., Rybar, M., Uhler, C., Fienberg, S.E.: Differentially-private logistic regression for detecting multiple-SNP association in GWAS databases. In: Domingo-Ferrer, J. (ed.) PSD 2014. LNCS, vol. 8744, pp. 170–184. Springer, Heidelberg (2014)
12. Zhao, Y., Wang, X., Jiang, X., Ohno-Machado, L., Tang, H.: Choosing blindly but wisely: differentially private solicitation of DNA datasets for disease marker discovery. JAMIA **22**, 100–108 (2015)

Efficient Privacy-Preserving Read Mapping Using Locality Sensitive Hashing and Secure Kmer Voting

Victoria Popic(✉) and Serafim Batzoglou

Department of Computer Science, Stanford University, Stanford, CA, USA
{viq,serafim}@stanford.edu

Recent sequencing technology breakthroughs have resulted in an exponential increase in the amount of available sequencing data, enabling major scientific advances in biology and medicine. At the same time, the compute and storage demands associated with processing such datasets have also dramatically increased. Outsourcing computation to commercial low-cost clouds provides a convenient and cost-effective solution to this problem. However, exposing genomic data to an untrusted third-party also raises serious privacy concerns [1]. Read alignment is a critical and computationally intensive first step of most genomic data analysis pipelines. While significant effort has been dedicated to optimize this task, few approaches have addressed outsourcing this computation securely to an untrusted party. The few secure solutions that exist either do not scale to whole genome sequencing datasets [2] or are not competitive with the state of the art in read mapping [3].

In this work we present BALAUR, a privacy preserving read mapping technique that securely outsources a significant portion of the read-mapping task to the public cloud, while being highly competitive with existing state-of-the-art aligners. Our approach is to reduce the alignment task to a secure voting procedure based on matches between read and reference kmers, taking advantage of the high similarity between the reads and their corresponding positions in the reference. At a high level, BALAUR can be summarized in the following two phases: (1) fast identification of a few candidate alignment positions in the genome using the locality sensitive hashing scheme MinHash [4] on the private client and (2) secure kmer voting against each such candidate position to determine the optimal read mappings on the public server. To outsource Phase 2 securely to the cloud, voting is performed using encrypted kmers of each read and its selected reference candidate contigs. In order to prevent frequency attacks using background knowledge (e.g. kmer repeat statistics), our encryption scheme uses the traditional cryptographic hashing scheme SHA-1, along with unique per-read keys and intra-read repeat masking, which prevents the adversary from detecting kmers that are equal across and inside each read or contig. We compare the performance of BALAUR with several popular and efficient non-cryptographic state-of-the-art read aligners, such as BWA-MEM [5] and Bowtie 2 [6], using simulated and real whole human genome sequencing datasets. We demonstrate that our approach achieves

M. Singh (Ed.): RECOMB 2016, LNBI 9649, pp. 248–249, 2016.
DOI: 10.1007/978-3-319-31957-5

similar accuracy and runtime performance on typical short-read datasets, while being significantly faster than state of the art in long read mapping.

References

1. Erlich, Y., Narayanan, A.: Routes for breaching and protecting genetic privacy. Nat. Rev. Genet. **15**(6), 409–421 (2014)
2. Huang, Y., Evans, D., Katz, J., Malka, L.: Faster secure two-party computation using garbled circuits. In: USENIX Security Symposium, vol. 201 (2011)
3. Chen, Y., Peng, B., Wang, X., Tang, H.: Large-scale privacy-preserving mapping of human genomic sequences on hybrid clouds. In: NDSS (2012)
4. Broder, A.Z., Charikar, M., Frieze, A.M., Mitzenmacher, M.: Min-wise independent permutations. J. Comput. Syst. Sci. **60**(3), 630–659 (2000)
5. Li, H.: Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. arXiv preprint arXiv:1303.3997 (2013)
6. Langmead, B., Salzberg, S.L.: Fast gapped-read alignment with Bowtie 2. Nat. Methods **9**(4), 357–359 (2012)

Finding Mutated Subnetworks Associated with Survival in Cancer

Tommy Hansen[1] and Fabio Vandin[1,2,3]

[1] Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Denmark
tvhansen33@gmail.com, vandinfa@dei.unipd.it
[2] Department of Information Engineering, University of Padova, Padova, Italy
[3] Department of Computer Science, Brown University, Providence, RI, USA

Next-generation sequencing technologies allow the measurement of somatic mutations in a large number of patients from the same cancer type. One of the main goals in the analysis of these mutations is the identification of mutations associated with clinical parameters, for example survival time. This goal is hindered by the extensive genetic heterogeneity in cancer, with different genes mutated in different patients. This heterogeneity is due to the fact that genes and mutations act in the context of *pathways*, and it is therefore crucial to study mutations in the context of interactions among genes. In this work we study the problem of identifying subnetworks of a large gene-gene interaction network that have somatic mutations associated with survival from genome-wide mutation data of a large cohort of cancer patients. We formally define the associated computational problem by using a score for subnetworks based on the test statistic of the log-rank test, a widely used statistical test for comparing the survival of two given populations. We show that the computational problem is NP-hard in general and that it remains NP-hard even when restricted to graphs with at least one node of large degree, the case of interest for gene-gene interaction networks.

We propose a novel randomized algorithm, called Network of Mutations Associated with Survival (NoMAS), to find subnetworks of a large interaction network whose mutations are associated with survival time. NoMAS is based on the color-coding technique, but differently from previous applications of color-coding our score is not additive, therefore NoMAS does not inherit the guarantees given by color-coding for the identification of the optimal solution. Nonetheless, we prove that under a reasonable model for mutations in cancer NoMAS does identify the optimal solution with high probability when the subnetwork size is not too large and given mutations from a sufficiently large number of patients. We implemented NoMAS and tested it on simulated and cancer data. The results show that our method does indeed find the optimal solution and performs better than greedy approaches commonly used to solve optimization problems on networks. Moreover, on two large cancer datasets NoMAS identifies subnetworks with significant association to survival, while none of the genes in the subnetwork has significant association with survival when considered in isolation.

This work is supported, in part, by the University of Padova under project CPDA121378/12 and by NSF grant IIS-1247581.

M. Singh (Ed.): RECOMB 2016, LNBI 9649, p. 250, 2016.
DOI: 10.1007/978-3-319-31957-5

Multi-state Perfect Phylogeny Mixture Deconvolution and Applications to Cancer Sequencing

Mohammed El-Kebir[1], Gryte Satas[1], Layla Oesper[1,2], and Benjamin J. Raphael[1]

[1] Center for Computational Molecular Biology and Department of Computer Science, Brown University, Providence, RI 02912, USA
[2] Department of Computer Science, Carleton College, Northfield, MN 55057, USA

Abstract. The reconstruction of phylogenetic trees from mixed populations has become important in the study of cancer evolution, as sequencing is often performed on bulk tumor tissue containing mixed populations of cells. Recent work has shown how to reconstruct a perfect phylogeny tree from samples that contain mixtures of two-state characters, where each character/locus is either mutated or not. However, most cancers contain more complex mutations, such as copy-number aberrations, that exhibit more than two states. We formulate the Multi-State Perfect Phylogeny Mixture Deconvolution Problem of reconstructing a multi-state perfect phylogeny tree given mixtures of the leaves of the tree. We characterize the solutions of this problem as a restricted class of spanning trees in a graph constructed from the input data, and prove that the problem is NP-complete. We derive an algorithm to enumerate such trees in the important special case of cladisitic characters where the ordering of the states of each character is given. We apply our algorithm to simulated data and to two cancer datasets. On simulated data, we find that for a small number of samples, the Multi-State Perfect Phylogeny Mixture Deconvolution Problem often has many solutions, but that this ambiguity declines quickly as the number of samples increases. On real data, we recover copy-neutral loss of heterozygosity, single-copy amplification and single-copy deletion events, as well as their interactions with single-nucleotide variants.

M. Singh (Ed.): RECOMB 2016, LNBI 9649, p. 251, 2016.
DOI: 10.1007/978-3-319-31957-5

Tree Inference for Single-Cell Data

Katharina Jahn[1,2], Jack Kuipers[1,2], and Niko Beerenwinkel[1,2]

[1] Department of Biosystems Science and Engineering, ETH Zurich, Basel, Switzerland
[2] SIB Swiss Institute of Bioinformatics, Basel, Switzerland

The genetic heterogeneity found within tumour cells is considered a major cause for the development of drug resistance during cancer treatment. Subclonal cell populations may possess a distinct set of genetic lesions that render them non-susceptible to the selected therapy resulting in an eventual tumour regrowth originating from the surviving cells. To develop more efficient therapies it is therefore paramount to understand the subclonal structure of the individual tumour along with its mutational history.

Classical next-generation sequencing techniques provide admixed mutation profiles of millions of cells whose deconvolution into subclones is often an underdetermined problem that limits the resolution at which the subclonal composition can be reconstructed. Recent technological advances now allow for the sequencing of individual cells. While this progress comes at the cost of higher error rates, it still provides the possibility to reconstruct mutational tumour histories at an unprecedented resolution.

We present a stochastic search algorithm to identify the evolutionary history of a tumour from noisy and incomplete mutation profiles of single cells. Our approach, termed SCITE, comprises a flexible MCMC sampling scheme that allows us to compute the maximum likelihood mutation tree and to sample from its posterior probability distribution. Tree reconstruction can include attachment of the single-cell samples and can be combined with estimating the error rates of the sequencing experiments. We evaluate SCITE on real cancer data showing its scalability to present day single-cell sequencing data and improved accuracy in tree reconstruction over existing approaches. In addition, we estimate from simulation studies the number of cells necessary for reliable mutation tree reconstruction which could inform the design of future single-cell sequencing projects.

K. Jahn and J. Kuipers—Equal contributors

M. Singh (Ed.): RECOMB 2016, LNBI 9649, p. 252, 2016.
DOI: 10.1007/978-3-319-31957-5

mLDM: A New Hierarchical Bayesian Statistical Model for Sparse Microbial Association Discovery

Yuqing Yang[1,2], Ning Chen[1], and Ting Chen[1,2,3,4]

[1] Bioinformatics Division and Center for Synthetic & Systems Biology, TNLIST, Beijing, China
{ningchen,tingchen}@tsinghua.edu.cn
[2] Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
[3] State Key Lab of Intelligent Technology and Systems, Tsinghua University, Beijing 100084, China
[4] Program in Computational Biology and Bioinfomatics, University of Southern California, Los Angeles, CA 90089, USA

Understanding associations among microbes and associations between microbes and their environmental factors from metagenomic sequencing data is a key research topic in microbial ecology, which could help us to unravel real interactions (e.g., commensalism, parasitism, competition, etc.) in a community as well as understanding community-wide dynamics. Although several statistical tools have been developed for metagenomic association studies, they either suffer from compositional bias or fail to take into account environmental factors that directly affect the composition of a microbial community, leading to some false positive associations. For example, two unrelated microbes may appear to be associated just because they both respond to the same environmental perturbation.

We propose metagenomic Lognormal-Dirichlet-Multinomial (mLDM), a hierarchical Bayesian model with sparsity constraints to bypass compositional bias and discover new associations among microbes and associations between microbes and their environmental factors. mLDM is able to: (1) infer both conditionally dependent associations among microbes and direct associations between microbes and environmental factors; (2) consider both compositional bias and variance of metagenomic data; and (3) estimate absolute abundance for microbes. These associations can capture the direct relationships underlying pairs of microbes and remove the indirect connections induced from other common factors. mLDM discovers the metagenomic associations using a hierarchical Bayesian graphical model with sparse constraints, where the metagenomic sequencing data generating process is captured by the hierarchical latent variable model. Specifically, we assume that the read counts are proportional to

This paper was selected for oral presentation at RECOMB 2016 and an abstract is published in the conference proceedings. The work is supported by the NSFC grant (Nos: 61305066, 61561146396, 61332007, 61322308), NIH grant (NIH/NHGRI 1U01 HG006531-01) and NSF grants (NSF/OCE 1136818 and NSF/DMS ATD 7031026).

M. Singh (Ed.): RECOMB 2016, LNBI 9649, pp. 253–254, 2016.
DOI: 10.1007/978-3-319-31957-5

the latent microbial ratios which are determined by their absolute abundance. The microbial absolute abundance is influenced by two factors: (1) environmental factors, whose effects on the microbes are denoted by a linear regression model; and (2) the associations among microbes encoded by a latent vector, which is determined by the matrix that records microbial associations and the mean vector that affects the basic absolute abundance of microbes. By introducing sparsity regularization, mLDM can capture both the conditionally dependent associations among microbes and the direct associations between microbes and environmental factors. The task is formulated as solving an optimization problem, which can be solved using coordinate descent or proximal methods. For model selection, we choose the best parameters via extended Bayesian information criteria (EBIC).

To show the effectiveness of the proposed mLDM model, we conducted several experiments using synthetic data, the western English Channel time-series sequencing data, and the Ocean TARA data, and compared it with several state-of-the-art methodologies, including PCC, SCC, LSA, CCREPE, SparCC, CCLasso, glasso (graphical lasso), SPIEC-EASI (mlasso) and SPIEC-EASI (glasso). The results demonstrate that the association network computed by the mLDM model, is closest to the true network, and that the mLDM model can recover most of the conditionally dependent associations. For the latter two experimental datasets, mLDM can discover most known interactions in addition to several potentially interesting associations.

Low-Density Locality-Sensitive Hashing Boosts Metagenomic Binning

Yunan Luo[1,4], Jianyang Zeng[1], Bonnie Berger[2,3], and Jian Peng[4]

[1] Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China
[2] Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, MA, USA
bab@mit.edu
[3] Department of Mathematics, MIT, Cambridge, MA, USA
[4] Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA
jianpeng@illinois.edu

1 Introduction

Metagenomic sequencing techniques produce large data sets of DNA fragments (e.g. reads or contigs) from environmental samples. To understand the microbial communities and functional structures within the samples, metagenomic sequence fragments need to be first assigned to their taxonomic origins from which they were derived (also called "binning") to facilitate downstream analyses.

Arguably the most popular metagenomic binning approaches are alignment-based methods. A sequence fragment is searched against a reference database consisting of full genomes of organisms, and the highest scoring organism is assigned as the taxonomic origin. Although efficient sequence alignment algorithms, including BWA-MEM [1], Bowtie2 [2] and (mega)BLAST [3], can readily be used for this purpose, the computational cost of alignment-based methods becomes prohibitive as the size of the sequence dataset grows dramatically, which is often the case in recent studies.

Another completely different binning approach is based on genomic sequence composition, which exploits the sequence characteristics of metagenomic fragments and applies machine learning classification algorithms to assign putative taxnomic origins to all fragments. Since classifiers, such as support vector machines, are trained on whole reference genome sequences beforehand, compositional methods normally are substantially faster than alignment-based methods on large datasets. The rationale behind compositional-based binning methods is based on the fact that different genomes have different conserved sequence composition patterns, such as GC content, codon usage or a particular abundance distribution of consecutive nucleotide k-mers. To design a good compositional-based algorithm, we need to extract informative and discriminative features from the reference genomes. Most existing methods, including PhyloPythia(S) [4, 5], use k-mer frequencies to represent sequence fragments, where k is typically small (e.g. 6 to 10). While longer k-mers, which capture compositional dependency

M. Singh (Ed.): RECOMB 2016, LNBI 9649, pp. 255–257, 2016.
DOI: 10.1007/978-3-319-31957-5

within larger contexts, could potentially lead to higher binning accuracy, they are more prone to noise and errors if used in the supervised setting. Moreover, incorporating long k-mers as features increases computational cost exponentially and requires significantly larger training datasets.

2 Method

We introduce a novel compositional metagenomic binning algorithm, Opal, which robustly represents long k-mers in a compact way to better capture the long-range compositional dependencies in a fragment. The key idea behind our algorithm is built on locality-sensitive hashing (LSH), a dimensionality-reduction technique that hashes input high-dimensional data into low-dimensional buckets, with the goal of maximiz the probability of collisions for similar input data. To the best of our knowledge, it is the first time that LSH functions have been applied for compositional-based metagenomic binning. We propose to use them first to represent metagenomic fragments compactly and subsequently for machine learning classification algorithms to train metagenomic binning models. Since metagenomic fragments can be very long, sometimes from hundreds of bps to tens of thousands of bps, we hope to construct compositional profiles to encode long-range dependencies within long k-mers. To handle large ks, we develop string LSH functions to compactly encode global dependencies with k-mers in a low-dimensional feature vector, as oppose to directly using a 4^k-length k-mer profile vector. Although LSH functions are usually constructed in a uniformly random way, we propose a new and efficient design of LSH functions based on the idea of the low-density parity-check (LDPC) code invented by Robert G Gallager for noisy message transmission [6, 7]. A key observation is that Gallager's LDPC design not only leads to a family of LSH functions but also makes them efficient such that even a small number of random LSH functions can effectively encode long fragments. Different from uniformly random LSH functions, the Gallager LSH functions are constructed structurally and hierarchically to ensure the compactness of the feature representation and robustness when sequencing noise appears in the data. Methodologically, starting from a Gallager design matrix with row weight t, we construct m hash functions to encode high-order sequence compositions within a k-mer. In contrast to the $O(4^k)$ complexity it would take to represent contiguous k-mers, our proposed Gallager LSH adaptation requires only $O(m4^t)$ time. For very long k-mers, we construct the Gallager LSH functions in a hierarchical fashion to further capture compositional dependencies from both local and global contexts. It is also possible to use Opal as a "coarse search" procedure in the compressive genomics manner to reduce the search space of alignment-based methods [8]. We first apply the compositional-based binning classifier to identify a very small subset or group of putative taxonomic origins which are ranked very highly by the classifier. Then we perform sequence alignment between the fragment and the reference genomes of the top-ranked organisms. This natural combination of compositional-based and alignment-based methods provides metagenomic binning with high scalability, high accuracy and high-resolution alignments.

3 Results

To evaluate the performance of Opal, we trained an SVM model with features generated by the Gallager LSH method. When tested on a large dataset of 50 microbial species, Opal achieved better binning accuracy than the traditional method that uses contiguous k-mer profiles as features [4]. Moreover, our method is more robust to mutations and sequencing errors, compared to the method with the contiguous k-mer representation. Opal outperformed (in terms of robustness and accuracy) BWA-MEM [1], the state-of-the-art alignment-based method. Remarkably, we achieved up to two orders of magnitude improvement in binning speed on large datasets with mutations rates ranging from 5 % to 15 % over 20–50 microbial species; moreover, we found Opal to be substantially more accurate than BWA-MEM when the rate of sequencing error is high (e.g., 10–15 %). It is counterintuitive that a compositional binning approach is as robust as or even more robust than alignment-based approaches, particularly in the presence of high sequencing errors or mutations in metagenomic sequence data. Finally, we combined both compositional and alignment-based methods, by applying the compositional SVM with the Gallager LSH coding as a "coarse-search" procedure to reduce the taxonomic space for a subsequent alignment-based BWA-MEM "fine search." This integrated approach is almost 20 times faster than original BWA-MEM and also has substantially improved binning performance on noisy data. The above results indicate that Opal enables us to perform accurate metagenomic analysis for very large metagenomic studies with greatly reduced computational cost.

Acknowledgments. This work was partially supported by the US National Institute of Health Grant GM108348, the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301, the National Natural Science Foundation of China Grant 61033001, 61361136003 and 61472205.

References

1. Li, H.: Aligning sequence reads, clone sequences and assembly contigs with bwa-mem. arXiv preprints (2013). arXiv:1303.3997
2. Langmead, B., Salzberg, S.: Fast gapped-read alignment with bowtie 2. Nat. Methods **9**, 357–359 (2012)
3. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic local alignment search tool. J. Mol. Biol. **215**, 403–410 (1990)
4. Patil, K.R., Haider, P., Pope, P.B., Turnbaugh, P.J., Morrison, M., Scheffer, T., McHardy, A.C.: Taxonomic metagenome sequence assignment with structured output models. Nat. Methods **8**(3), 191–192 (2011)
5. McHardy, A.C., Martín, H.G., Tsirigos, A., Hugenholtz, P., Rigoutsos, I.: Accurate phylogenetic classification of variable-length DNA fragments. Nat. Methods **4**(1), 63–72 (2007)
6. Gallager, R.: Low-density parity-check codes. IEEE Trans. Inf. Theory **8**(1), 21–28 (1962)
7. MacKay, D., Neal, R.: Near shannon limit performance of low density parity check codes. Electron. Lett. **32**, 1645–1646 (1996)
8. Yu, Y.W., Daniels, N.M., Danko, D.C., Berger, B.: Entropy-scaling search of massive biological data. Cell Syst. **2**, 130–140 (2015)

metaSPAdes: A New Versatile *de novo* Metagenomics Assembler

Sergey Nurk[1], Dmitry Meleshko[1], Anton Korobeynikov[1], and Pavel Pevzner[1,2]

[1] Center for Algorithmic Biotechnology, Institute for Translational Biomedicine, Saint Petersburg State University, Saint Petersburg, Russia
[2] Department of Computer Science and Engineering, University of California, San Diego, USA
ppevzner@ucsd.edu

Metagenome sequencing has emerged as a technology of choice for analyzing bacterial populations and discovery of novel organisms and genes. While many metagenomics assemblers have been developed recently, assembly of metagenomic data remains difficult thus stifling biological discoveries.

We developed metaSPAdes tool that addresses the specific challenges of metagenomic assembly by combining new algorithmic ideas with methods previously proposed for assembling single cells [1] and highly polymorphic genomes [2].

metaSPAdes features (i) efficient analysis of strain mixtures, (ii) a novel repeat resolution approach that utilizes the local read coverage of the regions that are being reconstructed, (iii) a novel algorithm that, somewhat counterintuitively, utilizes strain differences to improve reconstruction of the consensus genomes of a strain mixture, and (iv) improved running time and reduced memory footprint to enable assemblies of large metagenomes.

We benchmarked metaSPAdes against the state-of-the-art metagenomics assemblers (MEGAHIT [3], IDBA-UD [4] and Ray-Meta [5]) across diverse datasets and demonstrated that it results in high-quality assemblies.

References

1. Bankevich, A., Nurk, S., Antipov, D., Gurevich, A., Dvorkin, M., Kulikov, A., Lesin, V., Nikolenko, S., Pham, S., Prjibelski, A., Pyshkin, A., Sirotkin, A., Vyahhi, N., Tesler, G., Alekseyev, M., Pevzner, P.: SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. J. Comput. Biol. **19**(5), 455–477 (2012)
2. Safonova, Y., Bankevich, A., Pevzner, P.: dipSPAdes: assembler for highly polymorphic diploid genomes. J. Comput. Biol. **22**(6), 528–545 (2015)
3. Li, D., Liu, C.M., Luo, R., Sadakane, K., Lam, T.W.: MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. Bioinformatics **31**(10), 1674–1676 (2015)
4. Peng, Y., Leung, H.C.M., Yiu, S.M., Chin, F.Y.L.: IDBA-UD: a de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth. Bioinformatics **28**(11), 1420–1428 (2012)
5. Boisvert, S., Raymond, F., Godzaridis, E., Laviolette, F., Corbeil, J.: Ray Meta: scalable de novo metagenome assembly and profiling. Genome Biol. **13**(12), R122 (2012)

M. Singh (Ed.): RECOMB 2016, LNBI 9649, p. 258, 2016.
DOI: 10.1007/978-3-319-31957-5

Distributed Gradient Descent in Bacterial Food Search

Shashank Singh[1], Sabrina Rashid[2], Saket Navlakha[3], and Ziv Bar-Joseph[4]

[1] Machine Learning Department and Department of Statistics, Carnegie Mellon University, Pittsburgh, PA 15213, USA
[2] Computational Biology Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA
[3] Integrative Biology Laboratory, The Salk Institute for Biological Studies, La Jolla, CA 92037, USA
[4] Machine Learning Department and Computational Biology Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA
zivbj@cs.cmu.edu

Extended Abstract

Communication and coordination play a major role in the ability of bacterial cells to adapt to changing environments and conditions. Recent work has shown that such coordination underlies several aspects of bacterial responses including their ability to develop antibiotic resistance. Here we show that a variant of a commonly used machine learning algorithm, *distributed gradient descent*, is utilized by large bacterial swarms to efficiently search for food when faced with obstacles in their environment. Similar to conventional gradient descent, by sensing the food gradient, each cell has its own belief about the location of the food source. However, given limits on the ability of each cell to accurately detect and move toward the food source in a noisy environment with obstacles, the individual trajectories may not produce the optimal path to the food source. Thus, in addition to using their own belief each cell also sends and receives messages from other cells (either by secreting specific proteins or by physical interaction), which are integrated to update its belief and determine its direction and velocity. The process continues until the swarm converges to the food source.

Our main contribution is to better understand the computation performed by cells during collective foraging. Current models of this process are largely based on differential equation methods which do not fully take into account how the topology of the cellular interaction network changes over time. Furthermore, the assumptions made by these models about the ability of cells to identify the source(s) of the messages and to utilize a large (effectively continuous valued) set of messages, are unrealistic given the limited computational powers bacteria cells possess.

Here, we develop a distributed gradient descent algorithm that makes biologically realistic assumptions regarding the dynamics of the cells, the size of the

S. Singh and S. Rashid—These authors contributed equally.

M. Singh (Ed.): RECOMB 2016, LNBI 9649, pp. 259–260, 2016.
DOI: 10.1007/978-3-319-31957-5

messages communicated, and their ability to identify senders, while still solving the bacterial food search problem more efficiently (in terms of the overall complexity of messages sent) and more quickly (in terms of the time it takes the swarm to reach the food source) when compared to current differential equation models. We prove that our model converges to a local minimum, under reasonable assumptions on how bacteria communicate and perform simulation studies and analysis of experimental data. These experiments indicate that our communication model is feasible and leads to improvements over prior methods and over single cell and single swarm behavior.

There are many parallel requirements of computational and biological systems, suggesting that each can learn from the other. We conclude by discussing how the efficient and robust bacterial gradient descent algorithms we developed can be used by distributed sensors or wireless networks that operate under strict communication and computation constraints.

Supporting movies: www.andrew.cmu.edu/user/sabrinar/Bacteria_Simulation_Movies/

AptaTRACE: Elucidating Sequence-Structure Binding Motifs by Uncovering Selection Trends in HT-SELEX Experiments

Phuong Dao[1], Jan Hoinka[1], Yijie Wang[1], Mayumi Takahashi[2], Jiehua Zhou[2], Fabrizio Costa[3], John Rossi[2], John Burnett[2], Rolf Backofen[3], and Teresa M. Przytycka[1(✉)]

[1] National Center of Biotechnology Information, National Library of Medicine, NIH, Bethesda, MD 20894 , USA
przytyck@ncbi.nlm.nih.gov
[2] Department of Molecular and Cellular Biology, Beckman Research Institute of City of Hope, Duarte, CA, USA
[3] Bioinformatics Group, Department of Computer Science, University of Freiburg, Freiburg, Germany

Abstract. Aptamers, short synthetic RNA/DNA molecules binding specific targets with high affinity and specificity, are utilized in an increasing spectrum of bio-medical applications. Aptamers are identified *in vitro* via the Systematic Evolution of Ligands by Exponential Enrichment (SELEX) protocol. SELEX selects binders through an iterative process that, starting from a pool of random ssDNA/RNA sequences, amplifies target-affine species through a series of selection cycles. HT-SELEX, which combines SELEX with high throughput sequencing, has recently transformed aptamer development and has opened the field to even more applications. HT-SELEX is capable of generating over half a billion data points, challenging computational scientists with the task of identifying aptamer properties such as sequence-structure motifs that determine binding. While currently available motif finding approaches suggest partial solutions to this question, none possess the generality or scalability required for HT-SELEX data, and they do not take advantage of important properties of the experimental procedure.

We present AptaTRACE, a novel approach for the identification of sequence-structure binding motifs in HT-SELEX derived aptamers. Our approach leverages the experimental design of the SELEX protocol and identifies sequence-structure motifs that show a signature of selection towards a preferred structure. In the initial pool, secondary structural contexts of each k-mer are distributed according to a background distribution. However, for sequence motifs involved in binding, in later selection cycles, this distribution becomes biased towards the structural context favored by the binding interaction with the target site. Thus, AptaTRACE aims at identifying sequence motifs whose tendency of residing in a hairpin, bugle loop, inner loop, multiple loop, dangling end, or of being paired converges to a specific structural context throughout the selection cycles

P. Dao and J. Hoinka—Equal contribution, these authors are listed in alphabetical order.

M. Singh (Ed.): RECOMB 2016, LNBI 9649, pp. 261–262, 2016.
DOI: 10.1007/978-3-319-31957-5

of HT-SELEX experiments. For each k-mer, we compute the distribution of its structural contexts in each sequenced pool. Then, we compute the relative entropy (KL-divergence) based score, to capture the change in the distribution of its secondary structure contexts from a cycle to a later cycle. The relative entropy based score is thus an estimate of the selection towards the preferred secondary structure(s).

We show our results of applying AptaTRACE to simulated data and an *in vitro* selection consisting of high-throughput data from 9 rounds of cell-SELEX. In testing on simulated data, AptaTRACE outperformed other generic motif finding methods in terms of sensitivity. By measuring selection towards sequence-structure motifs by the change in their distributions of the structural contexts and not based on abundance, AptaTRACE can uncover motifs even when these are present only in a small fraction of the pool. Moreover, our method can also help to reduce the number of selection cycles required to produce aptamers with the desired properties, thus reducing cost and time of this rather expensive procedure.

Fast Bayesian Inference of Copy Number Variants Using Hidden Markov Models with Wavelet Compression

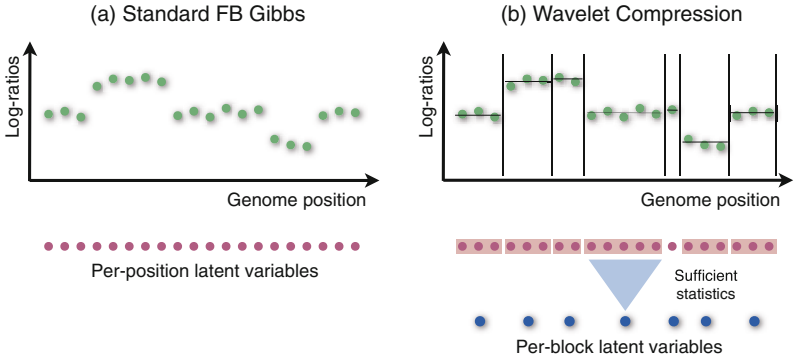
John Wiedenhoeft, Eric Brugel, and Alexander Schliep

Department of Computer Science, Rutgers University, Piscataway, NJ 08854, USA
{john.wiedenhoeft,alexander}@schlieplab.org

Hidden Markov Models (HMM) are statistical models frequently used in Copy Number Variant (CNV) detection. Classic frequentist maximum likelihood techniques for parameter estimation like Baum-Welch are not guaranteed to be globally optimal, and state inference via the Viterbi algorithm only yields a single MAP segmentation. Nevertheless, Bayesian methods like Forward-Backward Gibbs sampling (FBG) are rarely used due to long running times and slow convergence.

Here, we exploit that both state sequence inference and wavelet regression reconstruct a piecewise constant function from noisy data, though under different constraints. We draw upon a classic minimaxity result from wavelet theory to dynamically compress the data into segments of successive observation whose variance can be explained as emission noise under the current parameters in each FBG iteration, and are thus unlikely to yield state transitions indicating a break point. We further show that such a compression can be rapidly recomputed with little overhead using a simple data structure. Due to the summary treatment of subsequent observations in segments (or blocks) derived from the wavelet regression—see panels (a) and (b) below—we simultaneously achieve drastically reduced running times as well as improved convergence behavior of FBG. To the best of our knowledge this shows for the first time that a fully Bayesian HMM can be competitive with or outperform the current state of the art.

This makes routine diagnostic use and re-analysis of legacy data collections feasible; to this end, we also propose an effective automatic prior. An open source software implementation is available at http://schlieplab.org/Software/HaMMLET/.



M. Singh (Ed.): RECOMB 2016, LNBI 9649, p. 263, 2016.
DOI: 10.1007/978-3-319-31957-5

Allele-Specific Quantification of Structural Variations in Cancer Genomes

Yang Li[1], Shiguo Zhou[2], David C. Schwartz[2], and Jian Ma[1,3,4]

[1] Department of Bioengineering, University of Illinois at Urbana-Champaign, Champaign, USA
[2] Laboratory for Molecular and Computational Genomics, University of Wisconsin-Madison, Madison, USA
[3] Carl R. Woese Institute for Genomic Biology, University of Illinois at Urbana-Champaign, Champaign, USA
[4] School of Computer Science, Carnegie Mellon University, Pittsburgh, USA
jianma@cs.cmu.edu

One of the hallmarks of cancer genome is aneuploidy, which causes abnormal copy numbers of alleles. Structural variations (SVs) can further modify the aneuploid cancer genomes into a mixture of rearranged genomic segments with extensive range of somatic copy number alterations (CNAs). Indeed, aneuploid cancer genomes have significantly higher rate of CNAs and SVs. However, although methods have been developed to identify SVs and allele-specific copy number of genome (ASCNG) separately, no existing algorithm can simultaneously analyze SVs and ASCNG. Such integrated approach is particularly important to fully understand the complexity of cancer genomes.

In this work, we introduce a novel computational method `Weaver` to identify allele-specific copy number of SVs (ASCNS) as well as the inter-connectivity of them in aneuploid cancer genomes. To our knowledge, this is the first method that can simultaneously analyze SVs and ASCNG. Under the same method framework, `Weaver` also provides base-pair resolution ASCNG. Note that in this work we specifically focus on the quantification of SV copy numbers, which is the key novelty of our method. Our framework is flexible to allow users to choose their own variant calling (including SV) tools. We use the variant calling results to build a cancer genome graph, which is subsequently converted to a pairwise Markov Random Field (MRF). In the MRF, the ASCNS and SV phasing configuration, together with ASCNG, are hidden states in the nodes and the observations contain all sequencing information, including coverage, read linkage between SNPs as well as connections between SVs and SNPs. Therefore, our goal of finding the ASCNS and SV phasing together with ASCNG is formulated as searching the *maximum a posteriori* (MAP) solution for MRF. We apply Loopy Belief Propagation (LBP) framework to solve the problem.

We extensively evaluated the performance of `Weaver` using simulation. We also compared with single-molecule Optical Mapping analysis and evaluated using real data (including MCF-7, HeLa, and TCGA whole genome sequencing samples). We demonstrated that `Weaver` is highly accurate and can greatly refine the analysis of complex cancer genome structure. We believe `Weaver` provides a more integrative solution to study complex cancer genomic alterations.

M. Singh (Ed.): RECOMB 2016, LNBI 9649, p. 264, 2016.
DOI: 10.1007/978-3-319-31957-5

Assembly of Long Error-Prone Reads Using de Bruijn Graphs

Yu Lin[1], Max W. Shen[1], Jeffrey Yuan[1],
Mark Chaisson[2], and Pavel A. Pevzner[1]

[1] Department of Computer Science and Engineering,
University of California San Diego, San Diego, USA
[2] Department of Genome Sciences, University of Washington,
Washington, D.C., USA

When the first reads generated using Single Molecule Real Time (SMRT) technology were made available, most researchers were skeptical about the ability of existing algorithms to generate high-quality assemblies from error-prone SMRT reads. Roberts et al. [3] even referred to this widespread skepticism as the error myth and argued that new assemblers for error-prone reads need to be developed to debunk this myth.

Recent algorithmic advances resulted in accurate assemblies from error-prone reads generated by Pacific Biosciences and even from less accurate Oxford Nanopore reads. However, previous studies of SMRT assemblies were based on the overlap-layout-consensus (OLC) approach, which dominated genome assembly in the last decade, is inapplicable to assembling long reads. This is a misunderstanding since the de Bruijn approach, as well as its variation called the *A-Bruijn* graph approach [2], was originally developed to assemble rather long Sanger reads.

There is also a misunderstanding that the de Bruijn graph approach can only assemble highly accurate reads and fails while assembling error-prone SMRT reads, yet another error myth that we debunk. The A-Bruijn graph approach was originally designed to assemble inaccurate reads as long as any similarities between reads can be reliably identified. However, while A-Bruijn graphs have proven to be useful in assembling Sanger reads and mass spectra (highly inaccurate fingerprints of amino acid sequences of peptides [1]), the question of how to apply A-Bruijn graphs for assembling SMRT reads remains open. We show how to generalize de Bruijn graphs to assemble long error-prone reads and describe the ABruijn assembler, which results in more accurate genome reconstructions than the state-of-the-art algorithms for assembling Pacific Biosciences and Oxford Nanopore reads.

References

1. Bandeira, N., Pham, V., Pevzner, P., Arnott, D., Lill, J.R.: Automated de novo protein sequencing of monoclonal antibodies. Nat. Biotechnol. **26**, 1336–1338 (2008)
2. Pevzner, P.A., Tang, H., Tesler, G.: De novo repeat classification and fragment assembly. Genome Res. **14**, 1786–1796 (2004)
3. Roberts, R.J., Carneiro, M.O., Schatz, M.C.: The advantages of SMRT sequencing. Genome Biol. **14**, 405 (2013)

M. Singh (Ed.): RECOMB 2016, LNBI 9649, p. 265, 2016.
DOI: 10.1007/978-3-319-31957-5

Locating a Tree in a Reticulation-Visible Network in Cubic Time

Andreas D.M. Gunawan[1], Bhaskar DasGupta[2], and Louxin Zhang[1]

[1] Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607, USA
matzlx@nus.edu.sg

[2] Department of Mathematics, National University of Singapore, Singapore 119076, Singapore

In studies of molecular evolution, phylogenetic trees are rooted trees, whereas phylogenetic networks are rooted acyclic digraphs. Edges are directed away from the root and leaves are uniquely labeled with taxa in phylogenetic networks. An important bioinformatics task is checking the "consistency" of two evolutionary models. This has motivated researchers to study the problem of determining whether a tree is displayed by a network or not, which is called the tree containment problem (TCP) [2, 3]. The cluster containment problem (CCP) is related algorithmic problem that asks whether or not a subset of taxa is a cluster in a tree displayed by a network [2].

Both the TCP and CCP are NP-complete [3], even on a very restricted class of networks [4]. An open questions was posed by van Iersel *et al.* asking whether or not the TCP is solvable in polynomial time for binary reticulation-visible networks [1, 2, 4]. A network is reticulation-visible if every reticulation separates the root of the network from some leaves [2], where reticulations are internal nodes of indegree greater than one and outdegree one.

We give an affirmative answer to the open problem of van Iersel, Semple and Steel by presenting a cubic time algorithm for the TCP for arbitrary reticulation-visible networks. The key tool used in our answer is a powerful decomposition theorem. It also allows us to design a linear-time algorithm for the cluster containment problem for networks of this type and to prove that every galled network with n leaves has $2(n-1)$ reticulation nodes at most. The full version of this work can be found at arXiv.org (arXiv:1507.02119v2).

References

1. Gambette, P., Gunawan, A.D.M., Labarre, A., Vialette, S., Zhang, L.: Locating a tree in a phylogenetic network in quadratic time. In: Przytycka, T.M. (ed.) RECOMB 2015. LNCS, vol. 9029, pp. 96–107. Springer, Heidelberg (2015)
2. Huson, D.H., Rupp, R., Scornavacca, C.: Phylogenetic Networks: Concepts, Algorithms and Applications. Cambridge University Press, Cambridge (2011)
3. Kanj, I.A., Nakhleh, L., Than, C., Xia, G.: Seeing the trees and their branches in the network is hard. Theor. Comput. Sci. **401**, 153–164 (2008)
4. van Iersel, L., Semple, C., Steel, M.: Locating a tree in a phylogenetic network. Inform. Process. Lett. **110**, 1037–1043 (2010)

M. Singh (Ed.): RECOMB 2016, LNBI 9649, p. 266, 2016.
DOI: 10.1007/978-3-319-31957-5

Joint Alignment of Multiple Protein-Protein Interaction Networks via Convex Optimization

Somaye Hashemifar, Qixing Huang, and Jinbo Xu

Toyota Technological Institute at Chicago, Chicago, USA
{somaye.hashemifar,pqx.huang,jinboxu}@gmail.com

Abstract. Protein-protein interaction (PPI) network alignment greatly benefits the understanding of evolutionary relationships among species and identifying conserved sub-networks. Although a few methods have been developed for multiple PPI networks alignment, the alignment quality is still far away from perfect. This paper presents a new method ConvexAlign for joint alignment of multiple PPI networks that can generate functionally much more consistent alignments than existing methods.

1 Introduction

This paper presents a novel method ConvexAlign for one-to-one global network alignment (GNA). A one-to-one alignment is a mapping in which one protein is cannot be aligned more than one protein in another network. ConvexAlign calculates the optimal alignment by maximizing a scoring function that integrates sequence similarity, network topology and interaction preserving. We formulate the problem as an integer program and relax it to a convex optimization problem, which enables us to simultaneously align all the PPI networks, without resorting to the widely-used seed-and-extension or progressive alignment methods. Then we use ADMM to solve the relaxed convex optimization problem. Our results show that ConvexAlign outperforms several popular alignment methods both topologically and biologically.

2 Method

Scoring function. Let $G = (V, E)$ denote a PPI network where V is the set of vertices (proteins) and E is the set of edges (interactions). A one-to-one multiple alignment between N networks is given by a binary matrix X where $X_{ij}(v_i, v_j) = 1$ if and only if v_i and v_j are aligned and there is at most one 1 in each row or column of X. It is easy to see X is positive semi-definite. Let C represent a matrix where each value C_{ij} indicates the similarity between two proteins v_i and v_j. Our goal is to find an alignment that maximizes the number of matched orthologous proteins and the number of preserved edges. We define the node score of an alignment $\mathcal{A}$ as follows: $f_{node}(\mathcal{A}) = \sum_{1 \le i < j \le N} C_{ij}, X_{ij}$.
We also define the edge score to count the number of preserved edges between all pairs of networks:

M. Singh (Ed.): RECOMB 2016, LNBI 9649, pp. 267–269, 2016.
DOI: 10.1007/978-3-319-31957-5

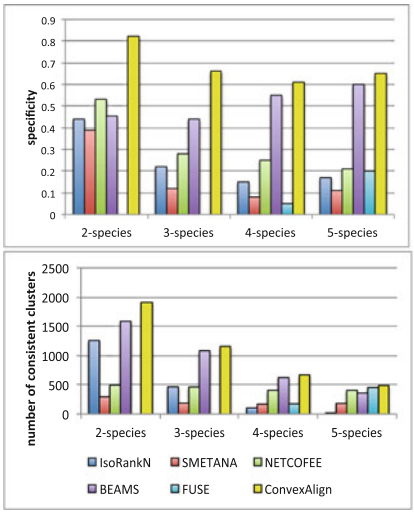


Fig. 1 Specificity and the number of consistent clusters generated by the competing methods for different c on real data where c is the number of species.

$$f_{edge}(\mathcal{A}) = \sum_{1 \le i < j \le N} \vec{1}, y_{ij}, \forall (v_i, v_i') \in E_i, \left(v_j, v_j'\right) \in E_j, 1 \le i < j \le N,$$

where $y_{ij} = X_{ij}\left(v_i, v_j\right) X_{ij}\left(v_i', v_j'\right)$. We aim to find the multiple alignment $\mathcal{A}$ that maximizes a combination of node and edge score as follows: $f = (1-\alpha) f_{node}(\mathcal{A}) + \alpha f_{edge}(\mathcal{A})$, where α describes the trade-off. By doing some calculations, the above objective function can be reformulated as

$$max \sum_{1 \le i < j \le N} (1-\alpha)\left\langle C_{ij}, X_{ij} \right\rangle + \alpha \left\langle \vec{1}, y_{ij} \right\rangle \tag{1}$$

$$y_{ij} \in \{0,1\}^{|E_i| \times |E_j|}, X_{ij} \in \{0,1\}^{|V_i| \times |V_j|}, 1 \le i < j \le N$$

$$B_{ij} y_{ij} \le \mathcal{F}_{ij}\left(X_{ij}\right), \mathrm{X_{ij}} \vec{1} \le \vec{1}, \mathrm{X}_{ij}^{T} \vec{1} \le \vec{1}, 1 \le i < j \le N$$

$$\mathrm{X} \succcurlyeq 0, \mathrm{X_{ii}} = \mathbf{I}_{|\mathrm{V_i}|}, 1 \le i \le N$$

where B_{ij} is coefficient and $\mathcal{F}ij$ is a linear operator that picks the corresponding element of X_{ij} for each constraint.

Optimization via Convex Relaxation. It is NP-hard to directly optimize (1) because the variables are binary. Therefore, we first relax the problem to obtain a convex optimization problem that can be solved to global optimum within polynomial time. We then use an ADMM method to solve the relaxed convex optimization problem that

can align all the proteins together. Finally, a greedy rounding strategy is applied to convert fractional solution to integral.

3 Results

We use both real and synthetic data to evaluate the performance of our method, ConvexAlign, with several popular methods. Tested on the PPI networks of five species human, yeast, fly, mouse and worm, ConvexAlign shows a better performance in terms of specificity and the number of functionally consistent clusters for all the clusters composed of proteins from $c = 2, 3, 4, 5$ species (Fig. 1). We have similar results on synthetic data.

Complexes Detection in Biological Networks via Diversified Dense Subgraphs Mining

Xiuli Ma[1], Guangyu Zhou[2], Jingjing Wang[2], Jian Peng[2], and Jiawei Han[2]

[1] Key Laboratory of Machine Perception (MOE), School of EECS, Peking University, Beijing, China
xlma@pku.edu.cn

[2] Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA
{gzhou6,jwang112,jianpeng,hanj}@illinois.edu

1 Introduction

Protein-Protein Interaction (PPI) networks, providing a comprehensive landscape of protein interacting patterns, enable us to explore biological processes and cellular components at multiple resolutions. For a biological process, a number of proteins need to work together to perform the job. Proteins densely interact with each other, forming large molecular machines or cellular building blocks. Identification of such densely interconnected clusters or protein complexes from PPI networks enables us to obtain a better understanding of the hierarchy and organization of biological processes and cellular components.

Most existing methods apply efficient graph clustering algorithms [1–3] on PPI networks, often failing to detect possible densely connected subgraphs and overlapped subgraphs. In this paper, we introduce a novel approximate algorithm to efficiently enumerate putative protein complexes from biological networks. The problem is formulated as finding a diverse set of dense subgraphs that cover as many as proteins as possible. To handle large networks, we take a divide-and-conquer approach to speedup the algorithm in a distributed manner. By comparing with existing clustering-based algorithms on several yeast and human PPI networks, we demonstrate that our method can detect more putative protein complexes and achieve better accuracy.

2 Method

We propose to model the problem of detecting complexes in biological networks as discovering the diversified maximal dense subgraphs. With the density measure, the dense subgraphs, that are protein complexes, can be defined explicitly and flexibly. Instead of enumerating all the dense subgraphs, we only find a small set of diversified maximal dense subgraphs. By maximal, we mean those complexes that are not subset of any other dense subgraphs thus cannot be further extended; By diversified, we mean a diverse set of dense subgraphs which cover

M. Singh (Ed.): RECOMB 2016, LNBI 9649, pp. 270–272, 2016.
DOI: 10.1007/978-3-319-31957-5

as many proteins as possible in the network. Combined into one goal, overlap is allowed but redundancy should be minimized.

In this paper, searching and diversifying are integrated tightly into one whole process. The key component of our algorithm is a set of efficient search trees that compactly traverse all the dense subgraphs by a depth-first construction. A node-specific potential is adopted to guide the search process. Furthermore, we identify two properties, the pseudo anti-monotonicity property for density and the sub-modularity property for diversity, and develop efficient pruning techniques based on these two properties. In this way, we extract the diversified dense subgraphs on the fly during the enumeration of the maximal dense subgraphs, thus greatly improve the scalability of the algorithm. Finally, the algorithm is scaled up in parallel to handle large-scale networks.

3 Result

We extensively evaluate the effectiveness and efficiency of our method on several PPI networks from yeast and human. We first evaluate the number of results and coverage for different methods on all the datasets under different density thresholds, showing our method can detect more complexes, while getting larger coverage. Then, we assess the quality of the predicted complexes by a composite score of three scores: fraction (frac), accuracy (acc) and maximum matching ratio (mmr), on both weighted and unweighted network (which is the binary version of the weighted datasets). In almost all the networks, our approach detects more putative complexes, and achieves higher accuracy and better one-to-one mapping with reference complexes in the ground truth databases than several state-of-art algorithms. The source code and supplementary data are available at https://github.com/zgy921028/MDSMine.

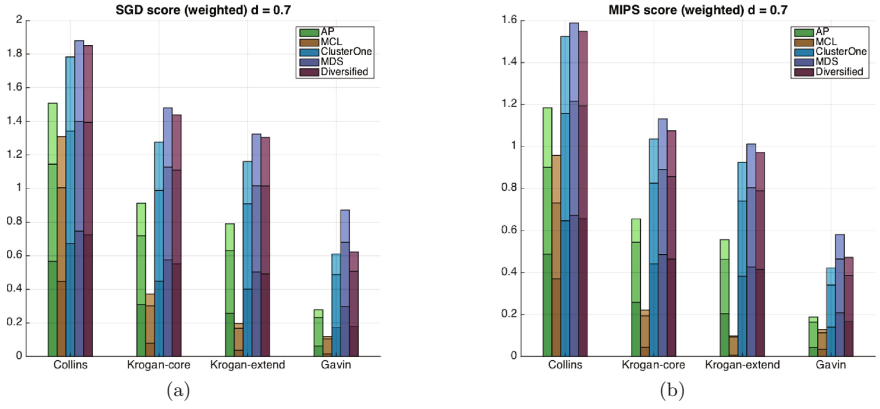


Fig. 1. Results (bottom-up: frac, acc, mmr) of various methods on 4 PPI weighted datasets using SGD (a) and MIPS (b) gold standard.

Acknowledgments. We thank the reviewers for their insightful comments. Xiuli Ma is supported by the National Natural Science Foundation of China under Grant No.61103025 and China Scholarship Council. This work was sponsored in part by the U.S. Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), National Science Foundation IIS-1017362, IIS-1320617, and IIS-1354329, HDTRA1-10-1-0120, and grant 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative (www.bd2k.nih.gov), and MIAS, a DHS-IDS Center for Multimodal Information Access and Synthesis at UIUC.

References

1. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. Science **315**, 972–976 (2007)
2. Nepusz, T., Yu, H., Paccanaro, A.: Detecting overlapping protein complexes in protein-protein interaction networks. Nat. Methods **9**(5), 471–472 (2012)
3. Van Dongen, S.: Graph clustering via a discrete uncoupling process. SIAM J. Matrix Anal. Appl. **30**(1), 121–141 (2008)

Author Index