Abstract of “Genome-wide algorithms for haplotype assembly, haplotype phasing, and IBD inference” by Derek Aguiar, Ph.D., Brown University, May 2014.

Determining the sequences of alleles co-inherited on a single chromosome, or *haplotypes*, is fundamentally important in genomics, molecular biology, and genomic medicine. Experimental methods for determining haplotypes are currently labor intensive, expensive, and do not scale. The computation of haplotypes from genome sequencing, or *haplotype assembly*, employs graph-theoretic and combinatorial algorithms intertwined with statistical models of DNA. The related problem of haplotype reconstruction from a population sample, or *haplotype phasing*, uses the statistical linkage between neighboring alleles and identical-by-descent (IBD) evolutionary relationships to reconstruct the haplotype sequences. This dissertation introduces graph-theoretic, combinatorial, and statistical algorithms for genome-wide haplotype reconstruction and IBD haplotype tract inference. Specifically, we present:

- **DELISHUS**, a mathematical model and exact polynomial-time algorithm for computing deletion haplotypes in SNP array data.
- The **HapCompass** algorithm for diploid genomes (e.g. humans) which models haplotype reconstruction as local optimizations on the cycle basis of a graph theoretic representation of variant alleles captured by sequence reads. This framework provides an algorithmic design strategy for a range of haplotype reconstruction problems and incorporates population genetics and identity-by-descent theory into the haplotype reconstruction model.
- The first model and algorithm for haplotype assembly of polyploid genomes, that is, organisms with more than two sets of homologous chromosomes (common in plant and tumor genomes).
- **Tractatus**, the first theoretically guaranteed exact and linear time algorithm for identical-by-descent multi-tract inference.

We compare our approaches with a variety of competing algorithms and investigate the feasibility of genome-wide haplotype reconstruction from computational and experimental perspectives.

Genome-wide algorithms for haplotype assembly, haplotype phasing, and IBD inference

by

Derek Aguiar

B.Sc., Computer Science and Computer Engineering, University of Rhode Island, 2007

M.Sc., Computer Science, Brown University, 2010

A dissertation submitted in partial fulfillment of the
requirements for the Degree of Doctor of Philosophy
in the Department of Computer Science at Brown University

Providence, Rhode Island

May 2014

This dissertation by Derek Aguiar is accepted in its present form by
the Department of Computer Science as satisfying the dissertation requirement
for the degree of Doctor of Philosophy.

Date ____________ ________________________________

Sorin Istrail, Director

Recommended to the Graduate Council

Date ____________ ________________________________

Franco Preparata, Reader

Date ____________ ________________________________

Eli Upfal, Reader

Date ____________ ________________________________

Eric Morrow, Reader

Approved by the Graduate Council

Date ____________ ________________________________

Peter M. Weber
Dean of the Graduate School

CV

Personal

work address	Box 1910, Computer Science Department Brown University Providence, RI 02912
personal phone	508-642-8082
email	rap@cs.brown.edu
web	http://www.derekaguiar.com

Research Interest Keywords

Computational Molecular Biology: haplotype phasing, haplotype assembly, autism genomics, population genomics, genomic regulatory network visualization and analysis, immunogenomics, RNA-Seq pipelines, deletion inference

Design and analysis of algorithms: graph theory, graph algorithms, combinatorial optimization, approximation algorithms

Education

Current Doctor of Philosophy in Computer Science, **Brown University** *Class of 2014*
Sept. 2009 Concentration: Computational Molecular Biology
Advisor: Professor Sorin Istrail
Dissertation: *Genome-wide algorithms for haplotype assembly, haplotype phasing, and IBD inference*

Sept. 2009 Master of Science in Computer Science, **Brown University** *Class of 2010*
Sept. 2008 Concentration: Computational Molecular Biology
Advisor: Professor Sorin Istrail

May 2007 Bachelor of Science Degree in Computer Science *Class of 2007*
Sept. 2003 Bachelor of Science Degree in Computer Engineering
Minor: Mathematics
University of Rhode Island, Kingston
Major GPA: 3.98/4.00
Cumulative GPA: 3.96/4.00

Publications

Derek Aguiar, Eric Morrow, Sorin Istrail, Tractatus: an exact and subquadratic algorithm for inferring identity-by-descent multi-shared haplotype tracts. In RECOMB, vol. 8394, pp. 1-17, 2014.

Derek Aguiar, Wendy S.W. Wong, Sorin Istrail, Tumor haplotype assembly algorithms for cancer genomics. In Pac Symp Biocomput., vol. 19, pp. 3-14, 2014.

Sarah Tulin[1], **Derek Aguiar**[1], Sorin Istrail, Joel Smith, A quantitative reference transcriptome for Nematostella vectensis early embryonic development: a pipeline for de novo assembly in emerging model systems. EvoDevo, vol. 4, no 16, 2013.

[1] denotes co-first authorship when ambiguous

Derek Aguiar, Sorin Istrail, Haplotype assembly in polyploid genomes and identical by descent shared tracts. In proceedings of ISMB 2013 and Bioinformatics, vol. 29, no. 13, pp. i352-i360, 2013.

Derek Aguiar, Bjarni V. Halldorsson, Eric M. Morrow, Sorin Istrail, DELISHUS: an efficient and exact algorithm for genome-wide detection of deletion polymorphism in autism, In proceedings of ISMB 2012 and Bioinformatics, vol. 28, no. 12, pp. i154-i162, 2012.

Derek Aguiar, Sorin Istrail, HAPCOMPASS: A fast cycle basis algorithm for accurate haplotype assembly of sequence data, In Journal of Computational Biology, vol. 19, no. 6, pp. 577-590, 2012.

Bjarni V. Halldorsson[1], **Derek Aguiar**[1], and Sorin Istrail. Haplotype phasing by multi-assembly of shared haplotypes: phase-dependent interactions between rare variants. In Pac Symp Biocomput., pages 88-99, 2011.

Bjarni V. Halldorsson[1], **Derek Aguiar**[1], Ryan Tarpine[1], and Sorin Istrail. The clark phaseable sample size problem: Long-Range phasing and loss of heterozygosity in GWAS. Journal of Computational Biology, 18(3):323-333, March 2011.

Bjarni V. Halldorsson[1], **Derek Aguiar**[1], Ryan Tarpine[1], and Sorin Istrail. The clark phase-able sample size problem: Long-Range phasing and loss of heterozygosity in GWAS. In Bonnie Berger, editor, RECOMB, volume 6044, pages 158-173, 2010.

Sorin Istrail, Ryan Tarpine, Kyle Schutter, and **Derek Aguiar**. Practical computational methods for regulatory genomics: A cisGRN-lexicon and cisGRN-browser for gene regulatory networks. In Istvan Ladunga, editor, Computational Biology of Transcription Factor Binding, volume 674 of Methods in Molecular Biology, pages 369-399. Humana Press, 2010.

Posters

D. Aguiar, A. Huang, R. Kantor, E. Morrow and S. Istrail, "Haplotype assembly in the presence of hemizygosity, haplotype sharing, polyploidy, and viral quasispecies," HiT-Seq workshop in the 21st Annual International Conference on Intelligent Systems for Molecular Biology, July 2013, Berlin, Germany. (selected for oral presentation)

S. Tulin, D. Aguiar, S. Istrail, and J. Smith, "Nematostella reference transcriptome and high throughput gene regulatory network construction," SDB 71st Annual Meeting, July 2012, Montreal, Canada.

D. Aguiar and S. Istrail, "HAPCOMPASS: A fast cycle basis algorithm for accurate haplotype assembly of next-generation sequence data," 20th Annual Intelligent Systems for Molecular Biology, July 2012, Long Beach, CA.

D. Aguiar, R. Tarpine, F. Lam, B. Halldorsson, E. Morrow, and S. Istrail, "Long-Range Haplotype Phasing by Multi-Assembly of Shared Haplotypes: Phase-Dependent Interactions Between Rare Variants," Genomics of Common Disease, Wellcome Trust Sanger Genome Center, Cambridge UK, December 2011

D. Aguiar, R. Tarpine, F. Lam, B. Halldorsson, E. Morrow, and S. Istrail, "Long-Range Haplotype Phasing by Multi-Assembly of Shared Haplotypes: Phase-Dependent Interactions Between

Rare Variants," The Gordon Research Conference on Human Genetics and Genomics, July 17-22, 2011, Salve Regina University, Newport, RI.

R. Tarpine, J. Hart, T. Johnstone, D. Aguiar, S. Istrail, "Report on the Cyrene Project: A cis-Lexicon Containing the Regulatory Architecture of 557 Regulatory Genes Experimentally Validated Using the *Davidson Criteria*," The Developmental Biology of the Sea Urchin Meeting, April 27-30, 2011, Woods Hole, MA.

D. Aguiar, R. Tarpine, E. Ruggieri, J. Nadel, D. Moskowitz, S. Istrail, "Beyond GWAS: Robust Computational Analysis of the Multiple Sclerosis Genetic Consortium Data," Fourth Annual Center for Computational Biology Poster Session, April 28, 2010, Brown University, RI.

Invited Talks

18th Annual International Conference on Research in Computational Molecular Biology "Tractatus: an exact and subquadratic algorithm for inferring identical-by-descent multi-shared haplotype tracts." April 2014

19th Pacific Symposium on Biocomputing "Tumor Haplotype Assembly Algorithms for Cancer Genomics." January 2014

IPP Symposium: Putting Big Data to Work. "Ome sweet ome: the genome as a model for big data." April 25, 2013

21st Annual International Conference on Intelligent Systems for Molecular Biology "Haplotype assembly in polyploid genomes and identical by descent shared tracts." July 2013

HiT-Seq workshop in the 21st Annual International Conference on Intelligent Systems for Molecular Biology "Haplotype assembly in the presence of hemizygosity, haplotype sharing, polyploidy, and viral quasispecies." July 2013

20th Annual International Conference on Intelligent Systems for Molecular Biology "DELISHUS: An efficient and exact algorithm for Genome-Wide detection of deletion polymorphism in autism." July 2012

Brown Computational Biology Open House "SNPs and haplotypes and GWAS oh my!" Feb. 27, 2012

Second Annual IEEE ICCABS CANGS Workshop "Robust algorithms for inferring haplotype phase and deletion polymorphism from high throughput whole genome sequence data" Feb. 24, 2012

Brown Computer Science - Research Exchange Seminars with Tea (REST) "Computational Challenges in Genome-wide Association Studies" Nov. 30, 2010

Fourteenth International Conference on Research in Computational Molecular Biology "The Clark Phase-able Sample Size Problem Long-range Phasing and Loss of Heterozygosity in GWAS" August 12, 2010

Grants

Brown 2014 Seed Award

Awarded to my thesis advisor Sorin Istrail and collaborator Eric Morrow, this grant was largely based on our collaboration and work in autism genomics. The grant application was entitled *Genome-wide sequence analysis in severe autism and intellectual disability*.

NSF award *1321000*

I wrote significant portions of this successful NSF grant devoted to developing HapCompass, Tractatus and related algorithms. The grant was entitled *Genome-Wide Algorithms for Haplotype Reconstruction and Beyond: A Combined Haplotype Assembly and Identical-by-Descent Tracts Approach*.

NSF award *1048831*

I contributed to sections of this grant which was largely based on our RECOMB 2010 work on haplotype phasing. The grant was entitled *Haplotype Phasing Algorithms and Clark Consistency Graphs*.

Academic Experience

Active | Research Assistant, **Brown University**
July 2008 | Advisor: Professor Sorin Istrail

Active | Teacher's Assistant, **Brown University**
Spring 2011, 2013, 2014 | Advised by: Sorin Istrail for the course *Algorithmic Foundations of Computational Biology*

My responsibilities were the same as the Spring 2010 course. In addition, developed and delivered a week's lecture on Haplotype Assembly.

Active | Academic Web Administration
2009-current | Istrail Lab Web www.brown.edu/Research/Istrail_Lab/

Maintain and develop the Istrail Lab web. In 2012, rebuilt the Istrail Lab web in PHP.

2009-2011 | Brown Center for Computational Molecular Biology Web

Maintained and enhanced the Brown CCMB web.

Active | Peer-Reviewing
2012-current | Bioinformatics reviewer
2010-current | JCB reviewer
2011 | PNAS sub-reviewer

Completed | Organizational Assistance
May 2010 | Brown University CCMB Symposium

Partially hosted guests, created presentations and produced brochures for the marketing of the symposium.

2009 | Brown University Center for Computational Biology PhD Program

Created documents including presentations and movies (using Google Earth, Google SketchUp, and KML) to support the creation of the PhD program for the Brown Center for Computational Biology. The PhD program was approved in 2009.

Completed | Teacher's Assistant, **Brown University**
Fall 2010, 2012, 2013 | Advised by: Sorin Istrail for the course *Medical Bioinformatics*

My responsibilities were the same as the Fall 2009 course.

Completed
Spring 2010

Teacher's Assistant, **Brown University**
Advised by: Sorin Istrail for the course *Algorithmic Foundations of Computational Biology*

My responsibilities included co-developing the syllabus, maintaining the course webpage, editing homework assignments, grading student hand-ins, creating tests, and creating lecture notes.

Completed
Fall 2009

Teacher's Assistant, **Brown University**
Advised by: Sorin Istrail for the course *Medical Bioinformatics*

My responsibilities included co-developing the syllabus, creating multiple assignments on linkage disequilibrium, haplotype phasing, and other related topics, managing the website, and delivering lectures on specialized research topics.

Completed
Summer 2007

Teacher's Assistant, **University of Rhode Island**, Kingston
Advised by: Donald Tufts for the course *Computer Communications*

During my senior year and based on my performance in the senior level course entitled Computer Communications, I was asked to independently research the topics covered with the objective of assisting in the redesign of the curriculum. The five lesson plans I created, including programming and homework assignments, incorporated: working at the frame level with Application, Transport, and Internet layer protocols (DNS, SMTP, TELNET, ARP, TCP/IP), and C Socket programming.

Scholarships & Honors

2014	RECOMB Student Travel Fellowship
2013	ISMB Student Travel Fellowship
2013	NSF EPSCoR Academy Travel Award
2012	ISMB Student Travel Fellowship
2012	IEEE ICCABS CANGS Workshop Student Travel Award
2010	RECOMB Student Travel Award
May 2007	President's Award for Excellence (Computer Science), URI
May 2007	President's Award for Excellence (Computer Engineering), URI
May 2007	Summa Cum Laude, URI
May 2007	Outstanding Graduating Senior in Computer Engineering, URI
May 2006	Outstanding Junior in Computer Engineering, URI

Memberships & Activities

2012-*present*	ISCB Membership
2006-*present*	IEEE Membership
2006-*present*	ACM Membership
2006-*present*	Phi Eta Sigma Honor Society
2006-*present*	Tau Beta Pi Honor Society
2007	Six Sigma Specialist

Industry Experience

May 2008
Oct. 2007

Software Engineer, Raytheon IDS, Portsmouth, RI
Zumwalt Total Ship Computing Environment Infrastructure

The majority of my work was spent developing and testing the data control and management software of a next-generation U.S. naval ship.

Oct. 2007 | Software Engineer, Raytheon IDS, Portsmouth, RI
June 2006 | *Joint Rapid Integrated Planning Service*

During my junior year I began working as a Software Engineer in Raytheon's Mission Innovation (MI) group, a small (20-30-people) division of Raytheon IDS that partners with universities to apply company technology and resources to world-threatening issues (e.g. climate change, biological diversity protection, civil defense, etc.). While in MI, I experienced the intellectual excitement of working in a small dedicated team of diverse individuals, in this case on civil-defense-related technology; we used Google Earth, KML, SQL, and .NET to develop a web-based collaborative disaster-planning tool (JRIPS). I am listed as co-inventor on a patent prepared by Raytheon.

Preface

The work presented in this dissertation was performed in the laboratory of Sorin Istrail, PhD and portions of this work have been published in Aguiar and Istrail (2012, 2013), Aguiar, Morrow, and Istrail (2014), Aguiar, Wong, and Istrail (2014), Aguiar et al. (2012), and Halldórsson et al. (2010, 2011). With the guidance of Sorin, I developed the theory, models, algorithms, and software presented herein, with the follow exceptions:

Development of the haplotype phasing algorithm described in Chapter 2 was a product of close collaboration with Ryan Tarpine and Bjarni Halldorsson. We collaborated with Bjarni Halldorsson to define the initial modeling for DELISHUS in Chapter 3. The applications of DELISHUS to autism was a product of the close collaboration with Dr. Eric Morrow. Wendy SW Wong simulated cancer data and provided valuable input for Chapter 5. Eric Morrow provided guidance and data for the experiments on autism GWAS data in Chapter 6.

Acknowledgements

I have many people to thank for the guidance and encouragement that has culminated in this dissertation.

First and foremost I thank my advisor Sorin Istrail who gave a naïve Masters student with very little research experience a chance to work in his lab. Sorin is the single most important reason I am writing this dissertation. From attending my first conference at RECOMB 2009 to presenting in my last conference while a graduate student at RECOMB 2014, Sorin has been the ideal mentor. As a researcher, Sorin has instilled in me several axioms: to build mathematically rigorous foundations for algorithms that balance complexity and practicality, to aim for challenging and open problems, and to collaborate closely with medical doctors and researchers that use our work in a non-abstract manner. As a teacher, Sorin has taught me to teach difficult problems and not be satisfied with the status quo. As a person, Sorin's journey from Romania to becoming a professor at Brown University is an inspiration. He has been integral in every aspect of my success as a graduate student from writing papers and delivering presentations to being supportive and understanding when I tear ligaments in my knee. Thank you Sorin.

Over the course of many collaborations, I have had the privilege of working with superb research scientists who are experts of their respective fields. Bjarni Halldorsson, Eric Morrow, Wendy Wong, Joel Smith, Marta Gomez-Chirarri, Russell Turner, and Sarah Tulin have all been extremely helpful throughout my graduate studies. I am also grateful to Eli Upfal and Franco Preparata for helping me throughout this process and providing valuable feedback as part of my thesis committee; and also for the occasional political debates at Sorin's dining room table.

The Istrail Lab has also provided a constant stream of talented students and post-docs that have helped me throughout the years. Ryan Tarpine, Austin Huang, and Alper Uzun are truly singular talents from which I have learned a great deal. I am also fortunate to have worked with great undergraduates including David Moskowitz, Ning Hou, Kyle Schutter, Allan Stewart, Tim Johnstone, James Hart, James Weis, Jacob Franco, Isaac Berkowitz, Aimee Lucido, and William

Turtle.

Several people were instrumental in my professional development prior to arriving at Brown. Thank you Ed Davis for giving me opportunities in high school to be creative. Edmund Lamagna taught the most difficult and, probably not coincidentally, the most interesting course at URI. Thank you for challenging me and igniting my interest in algorithms. Rob Lawrence, Dave Stuart, and Lucia Falcon were instrumental in my professional development in industry and provided sincere encouragement to pursue my doctoral degree.

Angel Murakami, Erin Klopfenstein, Louise Patterson, and Nathaniel Gill deserve an enormous thank you for all of the assistance they have given me. Even when situations were chaotic and stressful, they were always able to help when I asked.

The Bit and Byte departmental soccer teams provided an escape from the oftentimes stressful life of a graduate student. Over ten seasons I have had the pleasure of playing with many great soccer players and friends including Ugur Cetintemel, Pascal Van Hentenryck, Steve Zins, Erman Celen, Newton Ikhariale, Jadrian Miles, John Meehan, Geoff Sun, Serdar Kadioglu, Xin Jia, Yuda Dai, “It’s Chao time” Qian, Kazuhisa Shibata, Marco De Cecco, Eli Fox-Epstein, James Hays, Betsy Hilliard, Hannah Quay-de la Vallee, John Jannotti, Sunil Mallya, Joe Politz, and Bryce Richards.

A most sincere thank you goes to my family and friends who have helped me through rough times. I am so fortunate to have my Mom, Ed, Dad, and Lark who have been very supportive of my choice to pursue graduate school. I also thank my brothers Pat and Sean who have been great to hang with and discuss politics and religion. And thank you to Grandma Peg for being ever-so helpful and sweet, my Grandma Lucy, and the rest of my family. Besides my soccer friends, I have had great friends that help me take my mind off the stress of graduate school; thanks Ryan Wayne Replogle (RWR), Troy Cabral, Brendon Todd, Ray Rodgers, Nick Vital, Ryan LePage, Nic Catlin, John Ferreira, Brian Blake, Dave and Jim Gorman, Kathy, Sue, and Dawn, Derek Brodeur, Jonathan Jala, Chatterpaul Samaroo Joseph, and Joseph Corser. Natasha, thank you for being awesome; there aren’t many people that can surprise me but you still manage to do it. To Linus, Zoe, Toby, Sam, and Cody: meow, meow, meow, woof woof.

Finally, my thanks to the following organizations and individuals who have made my work possible with the generous gift of funding or data:

- the International Multiple Sclerosis Genetics Consortium for sharing the Multiple Sclerosis GWAS dataset.

- the Autism Genetic Resource Exchange (AGRE) Consortium and the participating AGRE

families. The AGRE is a program of Autism Speaks and is supported, in part, by grant 1U24MH081810 from the National Institute of Mental Health to Clara M. Lajonchere (PI).

- the National Science Foundation (award numbers 1048831 and 1321000 to Sorin Istrail).
- Lilian Janin and Anthony Cox at Illumina Cambridge Ltd for sharing and helping us with the EAGLE simulator.
- all of the families at the participating Simons Simplex Collection sites, and the Simons Foundation Autism Research Initiative.

Contents

List of Figures

Chapter 1

Introduction to Population Genomics

1.1 Molecular biology and genetics

Eukaryotic cells contain a set of deoxyribonucleic acid (*DNA*) molecules that collectively store the genetic material of the organism. The DNA molecule contains a sequence of nitrogen bases termed *nucleotides*, *bases*, or *basepairs* (bp) which encode the biological instructions expressed by a cell. DNA is physically organized in coiled molecular structures called chromosomes, which collectively constitute the *genome*. Advances in molecular biology have enabled researchers the ability to directly examine the genome sequence of many organisms and associate specific heritable sequences, or *genes*, with observable characteristics or *phenotypes*. Gene sequences are translated to RNA molecules that encode the instructions on how to synthesize proteins, which are the primary molecular functional units of the cell. It is hypothesized that genes are the basic unit of natural selection and genetic variation is responsible for much of the differences observed both within and between populations.

This flow of genetic information in the cell is summarized by the *central dogma of molecular biology* which suggests both the undirected flow of information between DNA and RNA molecules and the directional flow from RNA to proteins. In particular, normal cellular function includes replication of DNA, translation of DNA into RNA, and transcription of RNA into proteins. In special circumstances, RNA can be replicated or reverse transcribed into DNA (e.g. in HIV). DNA may also be directly transcribed into proteins but this is rare; the replication of protein to protein and translation of protein to DNA or RNA has not been observed. This theory is, of course, a simplification of the complexities that occur in the cell and current research suggests the existence of posttranslational modification of proteins, heritable variation in patterns of methylation in DNA, and a larger role for non-coding RNA.

1.1.1 Mendelian inheritance

Named for the Austrian monk, scientist, and a founder of modern genetics Gregor Mendel, Mendelian inheritance describes a set of fundamental rules governing the transmission of genetic material from parents to offspring. In his experiments, Mendel observed a convincingly consistent patterns when crossing pea plants with different observable phenotypes. For example, when crossing a pure yellow seed strain with a pure green strain, all seeds of the next generation were yellow. Mendel deduced that there was some fundamental unit of inheritance, which exists in pairs, segregates in the formation of the gametes, and unites in the formation of the offspring.

1.1.2 Variation

The laws governing Mendelian inheritance and the central dogma suggest that alterations in genome sequence can be passed to offspring and also propagate to synthesized proteins. The spectrum of DNA sequence variation ranges from single nucleotide polymorphisms (SNPs) to more complex structural variation (SV), for example deletions, insertions, or translocations of genomic material. Approximately 99.5% of any two individuals' genome sequences are shared within a population (Levy et al. 2007). The 0.5% of the nucleotide bases varying within a population explain, in part, the differences between individuals.

SNPs are the most abundant form of variation between two individuals in terms of number of variants. However, structural variation affects a larger number of nucleotide bases. These variations, which have shown to be increasingly important and an influential factor in many diseases (Stefansson et al. 2008), are not probed using SNP arrays. A further limitation of SNP arrays is that they are designed to probe only previously discovered, common variants. Rare variants, belonging perhaps only to a small set of carriers of a particular disease and hence potentially more deleterious, will not be detected using SNP arrays.

In the simplest case, a *haploid* genome contains a single copy of each chromosome. Other genomes, e.g. human, contain two copies of each chromosome and are termed *diploid*. Many plants and even some cells in human contains more than two copies and are referred to as polyploid or k-ploidy where k is the number of copies of the genome. Experimental techniques for determining the alternative forms of variants (alleles) produce unordered sets in which the chromosome of origin for each allele is unknown. The sequence of genomic alleles in a haploid genome with the non-varying DNA removed is referred to as a **haplotype**. When the sequence of alleles is experimentally determined for diploid or polyploid genomes, only the set of alleles at each variant are known and this sequence of ambiguous sets of alleles is termed a *genotype*.

1.1.3 Recombination and linkage disequilibrium

During the formation of the gametes in the process of meiosis, genetic material can be exchanged between homologous chromosomes forming a recombinant chromosome. This process of chromosomal crossover increases the amount of variation in a population by shuffling mutations on chromosomes. For example, humans have two sets of homologous chromosomes. Both gametes inherit a haploid copy of their parent's chromosome. In the absence of recombination, a copy of the haploid chromosome is inherited and alleles segregated on a single chromosome remain linked along its length. With recombination, contiguous segments of both homologous chromosomes are incorporated into the gametic chromosome. Segments of chromosome adjacent to a recombination breakpoint may include allelic relationships not observed in the parental genome.

The rate of recombination varies across the genome but, as a general rule, the probability of a recombination occurring between two mutations increases as the physical distance increases. Thus, two adjacent mutations with a small chromosomal distance between them are less likely to be segregated in different chromosomes than two mutations further apart. The concept of co-inherited variant alleles or statistical correlation between pairs or a set of alleles is termed linkage disequilibrium (Lewontin 1988). The relationship between recombination and linkage disequilibrium (LD) is demonstrated in Figure 1.1.

Linkage disequilibrium can be viewed as a deviation from the random association of alleles, or linkage equilibrium (LE). For two variants v_i, v_j with alleles $\{A, a\}$, $\{B, b\}$ and frequencies $\{f_A, f_a\}$, $\{f_B, f_b\}$ respectively, random association among the alleles would suggest the haplotype frequencies in Table 1.1. Popular measures of LD include the pairwise D, D', r (Hill and Robertson 1968; Lewontin 1988) and multi-loci informativeness measures (Lam, Tarpine, and Istrail 2011).

1.1.4 Genome-Wide Association Studies

A genome-wide association study (GWAS) is a large-scale approach of associating genetic variants with observable outcomes in a population. GWAS proceed by identifying a number of individuals

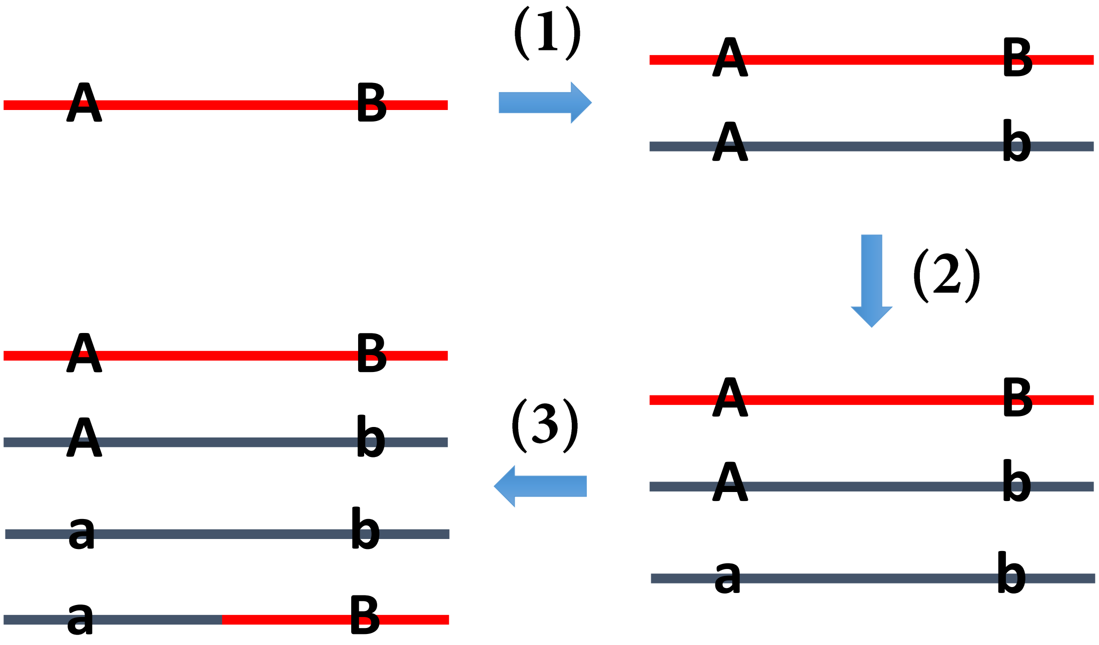


Figure 1.1: Recombination disrupts linkage disequilibrium. Originally, one ancestral non-mutated haplotype exists in the population. (1) A mutation is introduced increasing the number of haplotypes in the population to two. (2) When a distinct site is mutated a third haplotype is generated but the a allele completely determines the b allele so the sites remain in relatively strong linkage disequilibrium. The fourth haplotype may be generated from a the low probability event of an inheritable recurrent mutation or (3) a recombination joining the a and B alleles from the blue and red chromosomes.

Haplotype	LE Frequency	Observed Frequency	LD
AB	f_Af_B	f_{AB}	$f_{AB} - f_Af_B$
Ab	f_Af_b	f_{Ab}	$f_Af_b - f_{Ab}$
aB	f_af_B	f_{aB}	$f_af_B - f_{aB}$
ab	f_af_b	f_{ab}	$f_{ab} - f_af_b$

Table 1.1: Linkage disequilibrium (LD) viewed as deviation from linkage equilibrium (LE). $D \approx 0$ suggests linkage equilibrium while $D \neq 0$ indicates some level of linkage disequilibrium.

carrying a disease or trait and comparing these individuals to those that do not or are not known to carry the disease/trait. Both sets of individuals are then genotyped for a large number of SNP genetic variants, which are then tested for association to the disease/trait. GWAS have been able to successfully identify a very large number of polymorphisms associated to disease (Consortium 2007; Styrkarsdottir et al. 2008) and the amount of SNP data from these studies is growing rapidly. Studies using tens of thousands of individuals are becoming commonplace and are increasingly the standard in the association of genetic variants to disease (Gudbjartsson et al. 2008; Rivadeneira et al. 2009; Styrkarsdottir et al. 2008). These studies generally proceed by pooling together large amounts of genome-wide data from multiple studies, for a combined total of tens of thousands of individuals in a single meta-analysis study. It can be expected that if the number of individuals being genotyped continues to grow, hundreds of thousands, if not millions, of individuals will soon be studied for association to a single disease or trait.

1.2 Data

In this dissertation, we describe algorithms that operate on a variety of molecular data. In all applications we are interested in the haplotype sequences of alleles which are experimentally determined or computationally inferred from genotyping or sequencing technologies. Because experimental methods for determining haplotypes are infeasible to apply in most settings, we focus on acquisition of sequence and genotype data.

1.2.1 Sequencing

The development of the polymerase chain reaction (PCR), DNA microarray, and genome sequencing technologies have provided a foundation for the genomics era to thrive. PCR enables the exponential replication of DNA generating enormous samples of genomic material for various experimental techniques. DNA microarrays contain millions of single stranded DNA probes which hybridize to input DNA. The level of hybridization can be measured and can be applied to detect multiple types of variation or the expression of genes. Genome sequencing technologies can determine the sequence of bases in a genome. The cost of generating a raw megabase of DNA sequence has been decreasing faster than Moore's law since the early 2000's (KA. 2009).

1.2.2 Genotypes

The two dominant experimental techniques for determining genotype sequences in humans are genome sequencing and SNP microarrays.

Whole genome shotgun sequencing was employed to assembly the first human genome (Venter et al. 2001). In this landmark study, researchers fragmented the DNA randomly into contiguous pieces which could be sequenced using Sanger sequencing. Variants were inferred by comparing the consensus genome assembly sequence with the aligned sequence read depth and quality data. With the introduction of Illumina, 454, Pacific Biosciences, and other high-throughput sequencing technologies (Koboldt et al. 2013; Mardis 2013) the cost of sequencing has plummeted, but the workflow for calling variation from sequencing remains similar (Li et al. 2009; McKenna et al. 2010).

Genotyping with sequence data enables discovery of *de novo* SNPs and more complex structural mutations.

If the SNP alleles and adjacent sequences are known, then SNP arrays are an appropriate option for genotyping large population samples. SNP arrays detect and interpret hybridization signals from allele-specific oligonucleotide probes and target sequences. The hybridization signal can be interpreted as either homozygous for either allele or heterozygous in parallel for millions of SNPs. This technology is widely applicable to studies adopting the hypothesis that common variation is the dominant genetic contributors to heritable disease, which is common in the population (applies in most GWAS).

1.3 Haplotype reconstruction problems

High-throughput DNA sequencing technologies are producing increasingly abundant and long sequence reads. Third generation technologies promise to output even longer reads (up to a few kb) with increasingly long insert sizes. While the latter promises to alleviate many of the difficulties associated with high-throughput sequencing pipelines, both technologies suffer from producing haplotype phase ambiguous sequence reads. Determining the haplotype phase of an individual is computationally challenging and experimentally expensive; but haplotype phase information is crucial in bioinformatics workflows (Tewhey et al. 2011) including genetic association studies and identifying components of the missing heritability problem (e.g., phase-dependent interactions like compound heterozygosity (Krawitz et al. 2010; Pierson et al. 2012)), the reconstruction of phylogenies and pedigrees, genomic imputation (Marchini and Howie 2010), linkage disequilibrium and SNP tagging (Tarpine, Lam, and Istrail 2011).

Two categories of computational methods exist for determining haplotypes: haplotype phasing and haplotype assembly.

1.3.1 Haplotype Phasing

Large population-based studies of variation employ DNA microarrays to produce genotype information for a set of individuals. Given the genotypes of a sample of individuals from a population, *haplotype phasing* attempts to infer the haplotypes of the sample using haplotype sharing information within the sample. In the related problem of genotype imputation, a phased reference panel is used to infer missing markers and haplotype phase of the sample (Marchini and Howie 2010). Methods for haplotype phasing and imputation are based on computational (Halldórsson et al. 2004) and statistical inference techniques (Browning and Browning 2011b), but both use the fact that closely spaced markers tend to be in linkage disequilibrium and smaller haplotypes blocks are often shared in a population of seemingly unrelated individuals.

If a diploid genotype contains i heterozygous variants, then the space of haplotype pairs consistent with the genotype is 2^i. Fortunately, although an exponential number of haplotype pairs are possible, very few exist in the population. Haplotype phasing uses haplotype sharing in a population and the co-inheritance of closely linked alleles to infer the haplotype solutions for a set of genotypes.

1.3.2 Haplotype Assembly

Standard genome sequencing workflows produce contiguous DNA segments of an unknown chromosomal origin. *De-novo* assemblies for genomes with two sets of chromosomes (*diploid*) or more (*polyploid*) produce consensus sequences in which the relative haplotype phase between variants is undetermined. The set of sequencing reads can be mapped to the phase-ambiguous reference genome and the diploid chromosome origin can be determined but, without knowledge of the haplotype sequences, reads cannot be mapped to the particular haploid chromosome sequence. As a result, reference based genome assembly algorithms also produce unphased assemblies. However, sequence reads are derived from a single haploid fragment and thus provide valuable phase information when they contain two or more variants. *Haplotype assembly* – sometimes referred to as single individual haplotyping (Rizzi et al. 2002) – builds haplotypes for a single individual from a set of sequence reads (Schwartz 2010). The *haplotype assembly problem* aims to compute the haplotype sequences for each chromosome given a set of aligned sequence reads to the genome and variant information. After mapping the reads on a reference genome, reads are translated into haplotype *fragments* containing only the polymorphic single nucleotide polymorphism (SNP) sites. A fragment *covers* a SNP if the corresponding sequence read contains an allele for that SNP. Because DNA sequence reads originate from a haploid chromosome, the alleles spanned by a read are assumed to exist on the same haplotype. Haplotype assembly algorithms operate on either a *SNP-fragment matrix* containing a row for each fragment and columns for SNPs or an associated graph that models the relationship between fragments or their SNP alleles.

1.3.3 Identical-by-descent haplotype tract inference

The patterns of mutation and recombination along the genome produce a block-like structure where variants within blocks are in high LD while variants across blocks are in LE. Figure 1.3 shows the

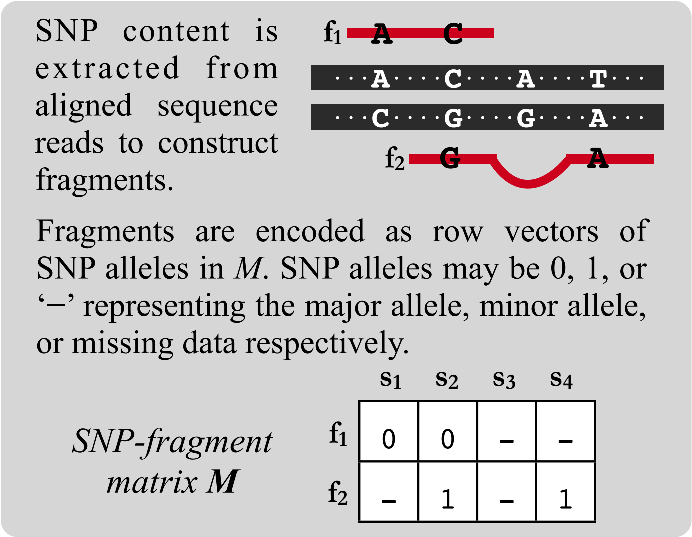

SNP content is extracted from aligned sequence reads to construct fragments.
f1 A C
· · · A · · · C · · · A · · · T · · ·
· · · C · · · G · · · G · · · A · · ·
f2 G A
Fragments are encoded as row vectors of SNP alleles in M. SNP alleles may be 0, 1, or '–' representing the major allele, minor allele, or missing data respectively.
SNP-fragment matrix M
s1 s2 s3 s4
f1 0 0 – –
f2 – 1 – 1

Figure 1.2: Construction of the input to the haplotype assembly problem.

structure of LD in a 200kbp region around BRCA2 for two populations. We refer to the unique haplotypes within these blocks as haplotype tracts. A haplotype tract can be unique or shared among a set of individuals and is defined by its start and stop positions in a set of haplotype. A haplotype tract in two or more individuals is identical-by-descent (IBD) if the sequence of alleles are identical and the tract is inherited from a common ancestor.

In general, IBD haplotype tracts can range from the very recent (e.g. parent and child) to the distant (e.g. two tracts 30 meioses apart). The expectation of both the amount of sharing and the percentage of shared ancestry is a function of the number of meioses to the most recent common ancestor. If two individuals are m^{th} degree cousins (separated by $2m + 2$ meioses), the expected percentage of genome in IBD haplotype tracts is $\frac{1}{2^{2m}}$. However, due to the patterns of recombination, the lengths of IBD tracts are approximately exponentially distributed and m^{th} degree cousins are expected to share a region of average size $\frac{200}{2n+2}$ centiMorgans in length (on average, 1 centiMorgan represents a distance of 750kbp in the human genome (Lodish 2008)). In other words, even though distantly related individuals may not share much of their genomes IBD, the haplotype tracts that are shared IBD are expected to be large. Algorithms for inferring IBD tracts exploit these properties of IBD to compute extended regions of sharing in haplotype and genotype data.

1.4 Algorithmic challenges

In this dissertation, we focus on combinatorial and statistical problems arising in haplotype reconstruction and identical-by-descent tract inference. Specifically, we study computing genomic

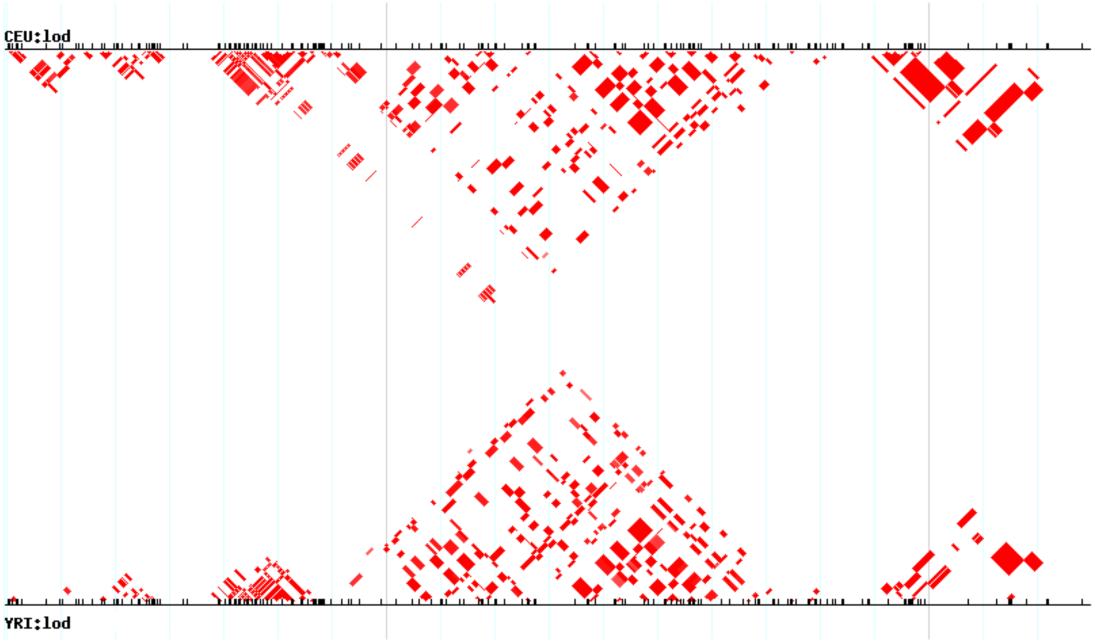


Figure 1.3: This plot shows the HapMap CEU (top) and YRI (bottom) populations log odds (lod) pairwise LD in a 200kbp region around BRCA2 ($lod > 10$) and generated using the HapMap genome browser (Thorisson et al. 2005). The variants are along the x-axis and the color of the cells that intersect from the diagonals of each variant represent the LD. The deeper the red color the more LD.

deletions from genotype data in the context of autism, haplotype phasing of large populations using IBD haplotype tracts, haplotype assembly of diploid, polyploid and cancer genomes, and IBD tract inference in haplotypes. To aid the reader, we briefly re-introduce key concepts at the start of each algorithmic part.

In Part I, we focus on algorithms for inferring haplotypes from genotypes. Chapter 2 focuses on phasing individuals using IBD tracts inferred from genotype data. If a set of individuals share a haplotype tract IBD, then every variant with at least one individual who is homozygous can be phased. The concept is related to the work of Kong et al. (2008) in which the combination of comprehensive pedigree information and extensive IBD sharing enabled the phasing of a large proportion of the 35,528 Icelanders genotyped. We ask a similar question: how many individuals in the United States population must be genotyped to haplotype phase the majority of SNPs using IBD without explicit knowledge of pedigree information?

The main focus of Chapter 3 is phasing deletion haplotypes in large GWAS-sized cohorts. We place a particular focus on autism and similar disorders of cognitive development. The connection between genomic deletions and autism is well documented, but the vast majority of these associations are for large and *de novo* deletions. Deletions inherited from a single parent, generally considered to be a healthy control in most study designs, have garnered less attention. We develop the DELISHUS algorithm that exploits the structure of recurrent deletions to infer inherited deletions which may impact developing brains sensitive to changes in protein concentration.

The overall theme of Part II is algorithms for inferring haplotype from sequence reads. Chapter 4 describes the well-studied problem of haplotype assembly of diploid genomes. For diploid genomes, we can make simplifying assumptions on the structure of intermediate solutions, namely that the two haplotypes are compliment of each other and there exists exactly two possible phasings between any two variants. Chapter 5 describes algorithmic extensions of diploid haplotype assembly to polyploid and cancer genomes. Assumptions made in the diploid case no longer hold, and thus we develop a statistical framework for phasing pairs of SNPs and model the resolution of conflicts with the k-disjoint paths problem which we can solve exactly for our graphs which have a specific structure.

Part III, Chapter 6 focuses on the problem of computing identical-by-descent haplotype tracts. We describe our exact linear-time algorithm, termed Tractatus, for computing all IBD multi-shared tracts in a set of haplotypes. The problem becomes more difficult when sequencing errors and mutations occurring after divergence from the recent common ancestor are allowed. The remainder of the chapter describes extensions for these cases and also inferring runs of homozygosity in genotype data.

The dissertation concludes with Part IV Chapters 7 and 8 which concentrate on a discussion of future work, open problems, and a summary of contributions.

Part I

Haplotype Phasing

Chapter 2

Haplotype Phasing Algorithms

To reach their full potential, the future direction of genetic association studies are mainly twofold: the testing of more individuals using genome-wide association arrays and the resequencing of a small number of individuals with the goal of detecting more types of genetic variations, both rare SNPs and structural variation (Siva 2008). Testing multiple individuals for the same variants using standard genome-wide association arrays is becoming increasingly common and can be done at a cost of approximately $100 per individual. In the next couple of years it is plausible that several million individuals in the US population will have been genotyped. In contrast, whole genome resequencing is currently in its infancy. A few people have had their genome resequenced and the cost of sequencing a single individual is still estimated in the hundreds of thousands of dollars. However, whole genome sequencing is preferable for association studies as it allows for the detection of all genomic variation and not only SNP variation.

Due to the fact whole genome SNP arrays are becoming increasingly abundant and whole genome resequencing is still quite expensive, the question has been raised whether it would suffice to whole genome sequence a small number of individuals and then impute other genotypes using SNP arrays and the shared inheritance of these two sets of individuals. It has been shown – in the Icelandic population with a rich pedigree structure known – that this could be done most efficiently using the haplotypes shared by descent between the individuals that are genotyped and those that have been resequenced (Kong et al. 2008). Haplotype sharing by descent occurs most frequently between closely related individuals, but also occurs with low probability between individuals that are more distantly related. In small, closely related populations, as in the Icelandic population, only a moderately sized sample size is therefore needed in order for each individual to have, with high probability, an individual that is closely related to them. In larger and more genetically diverse populations, such as the US population, a larger sample size will be needed for there to be a significant probability of an individual sharing a haplotype by descent within the population. We say that an individual is "Clark phaseable" with respect to a population sample if the sample contains another individual that shares a haplotype with this individual by descent. In this paper we explore what the required sample size is so that most individuals within the sample are Clark phaseable, when the sample is drawn from a large heterogeneous population, such as the US population.

Current technologies, suitable for large-scale polymorphism screening, only yield the genotype information at each SNP site. The actual haplotypes in the typed region can only be obtained at a considerably high experimental cost or computationally by haplotype phasing. Due to the importance of haplotype information for inferring population history and for disease associations, the development of algorithms for detecting haplotypes from genotype data has been an active research area for several years (Clark 1990; Halldórsson et al. 2004; Kong et al. 2008; Scheet and Stephens 2006; Sharan, Halldórsson, and Istrail 2006; Stephens, Smith, and Donnelly 2001). However, algorithms for determining haplotype phase are still in their infancy after about 15 years of development. Of particular worry is the fact that the learning rate of the algorithms, i.e. the rate that the algorithms are able to infer more correct haplotypes, grows quite slowly with the number of individuals being genotyped.

In this chapter we present an algorithm for the phasing of a large number of individuals. We show that the algorithm will get an almost perfect solution if the number of individuals being genotyped is large enough and the correctness of the algorithm grows with the number of individuals being genotyped. We will consider the problem of haplotype phasing from long shared genomic regions (that we call tracts). Long shared tracts are unlikely unless the haplotypes are identical-by-descent (IBD), in contrast to short shared tracts which may be identical by state (IBS). We show how we can use these long shared tracts for haplotype phasing.

2.1 Long Range Phasing and Haplotype Tracts

The haplotype phasing problem asks to computationally determine the set of haplotypes given a set of individual's genotypes. We define a *haplotype tract* (or *tract* for short) denoted $[i, j]$ as a sequence of SNPs that is shared between at least two individuals starting at the same position i in all individuals and ending at the same position j in all individuals. We show that if we have a long enough tract then the probability that the sharing is IBD is close to 1. Multiple sharing of long tracts further increases the probability that the sharing corresponds to the true phasing.

2.1.1 Probability of Observing a Long Tract

We show that as the length of the tract increases the probability that the tract is shared IBD increases. Let t be some shared tract between two individual's haplotypes and l be the length of that shared tract. We can then approximate the probability this shared tract is identical by state (IBS) $p_{IBS}(l)$. Let $f_{M,i}$ be the major allele frequency of the SNP in position i in the shared tract t. Assuming the Infinite Sites model and each locus is independent,

$$p_{IBS}(l) = \prod_{i=1}^{l} ((f_{M,i})(f_{M,i}) + (1 - f_{M,i})(1 - f_{M,i}))$$

We can approximate $p_{IBS}(l)$ by noticing $f_{M,i} * f_{M,i}$ dominates $(1 - f_{M,i})(1 - f_{M,i})$ as $f_{M,i} \to 1$, $p_{IBS}(l) \approx \prod_{i=1}^{l}(f_{M,i})^2$. Let f_{avg} be $\frac{1}{l} f_{M,i}\ \forall i \in t$. Then $p_{IBS}(l) \approx (f_{avg})^{2l}$. Given $f_{M,i}$ is some high frequency, say 95%, then a sharing of 100 consecutive alleles is very unlikely, $p_{IBS}(100) \approx 0.95^{200} = 10^{-5}$. For very large datasets we will need to select the length of the tract being considered to be large enough so that the probability that the sharing is identical by state is small.

The probability two individuals separated by $2(k+1)$ meiosis (kth-degree cousins) share a locus IBD is 2^{-2k} (Kong et al. 2008). As k increases, the probability kth-degree cousins share a particular locus IBD decreases exponentially. However, if two individuals share a locus IBD then they are expected to share about $\frac{200}{2k+2}$ cM (Kong et al. 2008). Relating $P(IBD)$ to length of tract l,

$$P(IBD|\textit{sharing of length } l) = \frac{2^{-2n}}{2^{-2n} + \left((f_{M,i})^{2l} + (1 - f_{M,i})^{2l}\right)}$$

which is shown in Fig. 2.1. Figure 2.1 shows the probability of IBD haplotype sharing given a tract of length l. We developed our phasing algorithm based on genotype sharing which exhibits a similar trend as Fig. 2.1, but shifted to the right (that is, we require more SNPs to commit to an IBD relationship).

2.2 The Clark Phase-able Sample Size Problem

Given the large tract sharing, we can construct the *Clark consistency graph* having individuals as vertices and an edge between two individuals if they share a tract (Sharan, Halldórsson, and Istrail 2006). Figure 2.2 shows the Clark consistency graph for different *minimum significant tract lengths*

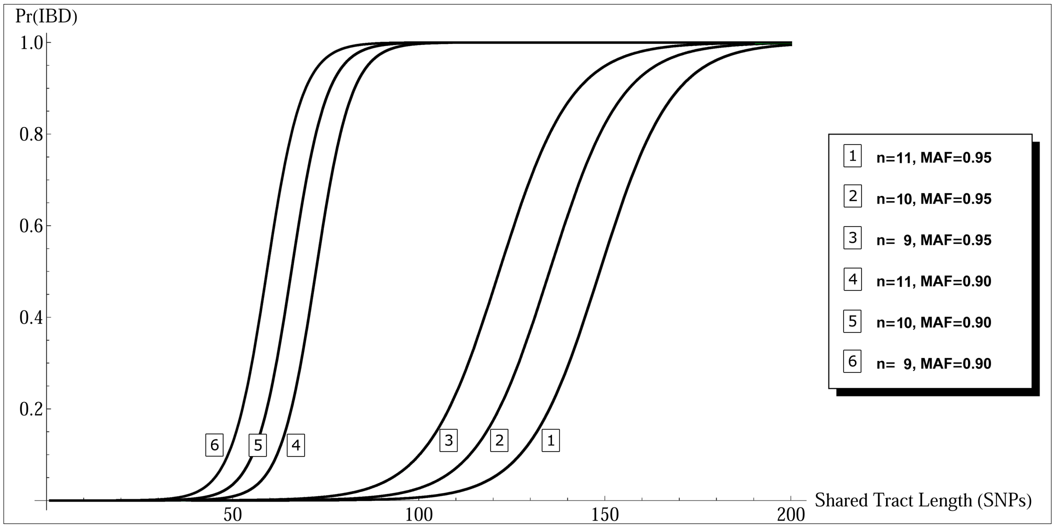


Figure 2.1: Probability of IBD as a function of shared tract length (measured in SNPs) and plotted for several n and major allele frequencies (MAF). Lower values for the MAF or n require less SNPs in a tract to commit to an IBD relationship.

(or window sizes) in the MS dataset. At what minimum significant tract lengths will the graph become dense enough so that phasing can be done properly? What percentage of the population needs to be genotyped so that the Clark consistency graph becomes essentially a single connected component? We call this "The Clark sample estimate: the size for which the Clark consistency graph is connected."

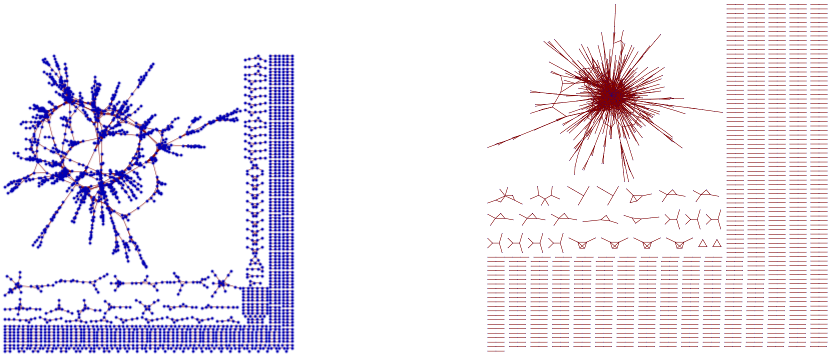


Figure 2.2: Left: The Clark consistency graph for SNP region [1400,1600). A large fraction of individuals share consistent haplotypes of length 200 suggesting many are IBD. Right: The Clark consistency graph for a smaller window size of 180 base pairs.

We computed the average number of edges in the haplotype consistency graph as a function of window size to get a sense of when the Clark consistency graph of the MS data becomes connected. Based on Fig. A.0.1 and $P(IBD)$ we can propose an algorithmic problem formulation from the

Clark consistency graph. Preferably we would like to solve either Problem 3 or 4.

Problem 1. *Remove the minimum number of the edges from the Clark consistency graph so that the resulting graph gives a consistent phasing of the haplotypes.*

Problem 2. *Maximize the joint probability of all the haplotypes given the observed haplotype sharing.*

We believe that both of these problem formulations are NP-hard and instead propose to solve these problems using a heuristic. Our benchmarking on simulated data shows that this heuristic works quite well.

2.2.1 Phasing the Individuals That Are Part of the Largest Component

We now proceed with an iterative algorithm working on the connected components in the Clark haplotype consistency graph. First we construct the graph according to some minimum length of haplotype consistency (Fig. A.0.1 and $P(IBD)$ aid in defining this length). We iterate through each site of each individual to find the tracts. After finding a site with some long shared region, we look at its neighbors in the connected component and apply a voting scheme to decide what the value is for each heterozygous allele. After each individual has been processed we iterate with having resolved sites in the original matrix.

Observation 1. *If the Clark consistency graph is fully connected in a window, all individuals can be phased at sites where there is at least one homozygote.*

Therefore, phasing individuals in a connected component of the graph should be easy, but in practice there will be some inconsistencies for a number of reasons. If a node in the Clark consistency graph has a high degree then the phasing of that node will be ambiguous if its neighbors are not consistent. At some times this may be due to genotyping error and at times this may be due to identical by state sharing to either one or both of an individuals haplotypes. The identical by state sharing may be a result of the haplotype having undergone recombination, possibly a part of the haplotype is shared identical-by-descent and a part is identical by state.

Our alphabet for genotype data is $\Sigma = \{0, 1, 2, 3\}$. 0s and 1s represent the homozygote for the two alleles of a SNP. A 2 represents a heterozygous site and a 3 represents missing data. Given a set of n-long genotype strings $G = \{g_1, g_2, \ldots, g_{|G|}\}$ where $g_i \in \Sigma^n$, we represent this in a matrix

M with $m = 2\,|G|$ rows and n columns:

$$M = \begin{bmatrix} M_{1,1} & M_{1,2} & \cdots & M_{1,n} \\ M_{2,1} & M_{2,2} & \cdots & M_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ M_{m,1} & M_{m,2} & \cdots & M_{m,n} \end{bmatrix}$$

Each genotype g_i is represented by the two rows $2i-1$ and $2i$. Initially, $M_{2i-1,j} = M_{2i,j} = g_i[j]$.

We define allele consistency to be:

$$c(a,\, b) = \begin{cases} 1 & \text{if } a = b \text{ or } a \in \{2,\, 3\} \text{ or } b \in \{2,\, 3\} \\ 0 & \textit{otherwise} \end{cases}$$

Rows r and s of M are consistent along a tract $[i,\, j]$ (i.e. have a shared tract) is written

$$C_{[i,\, j]}(r,\, s) = \prod_{k \in [i,\, j]} c\left(M_{r,k},\, M_{s,k}\right)$$

The length of a tract is written $|[i,\, j]| = j - i + 1$.

A shared tract $[i,\, j]$ between rows r and s is *maximal shared tract* if it cannot be extended to the left or right; i.e., $i = 1$ or $c(M_{r,i-1},\, M_{s,i-1}) = 0$ and $j = n$ or $c(M_{r,j+1},\, M_{s,j+1}) = 0$. The maximal shared tract between rows r and s at position i is written $S_i^{r,s}$. It is unique. Note that if $S_i^{r,s} = [j,\, k]$ then $\forall_{l \in [j,\, k]} S_l^{r,s} = S_i^{r,s}$.

2.2.2 Tract Finding and Phasing Algorithm

Given that there are some loci for which individuals share IBD and that these sharings are expected to be large, we developed an algorithm to detect and use these sharings to resolve the phase at heterozygous sites. Each site is resolved by determining if there are any other individuals that likely share a haplotype by descent. SNPs that do not have their phase determined during any given iteration will be processed in succeeding iterations. If there are enough long IBD loci, this algorithm should unambiguously determine the phase of each individual.

We start by phasing the trios using Mendelian laws of inheritance. This replaces many of the heterozygote sites (whenever at least one member of a family is homozygous) and even a few of the sites having missing data (i.e., when the parents are both homozygous and the child's genotype is missing).

To phase using long shared tracts, we start by fixing a minimum significant tract length L. We run several iterations, each of which generate a modified matrix M' from M, which is then used as the basis for the next iteration.

First, we set $M' := M$.

For each row r we examine position i. If $M_{r,i} \in \{0, 1\}$ then we move to the next i. Otherwise $M_{r,i} \in \{2, 3\}$, and we count "votes" for whether the actual allele is a 0 or 1.

$$V_0^r = |\{s \mid s \neq r \text{ and } |S_i^{r,s}| \geq L \text{ and } M_{s,i} = 0\}|$$

V_1^r is defined analogously (the difference being the condition $M_{s,i} = 1$). If $V_0^r > V_1^r$ then we set $M'_{r,i} := 0$. Similarly for $V_1^r > V_0^r$. If $V_0^r = V_1^r$ then we do nothing.

When $M_{r,i} = 2$, we make sure the complementary haplotypes are given different alleles by setting the values of both haplotypes simultaneously. This does not cause a dependency on which haplotype is visited first because we have extra information we can take advantage of. We count votes for the complementary haplotype and treat them oppositely. That is, votes for the complementary haplotype having a 1 can be treated as votes for the current haplotype having a 0 (and vice versa). So letting r' be the row index for the complementary haplotype, we actually compare $V_0^r + V_1^{r'}$ and $V_1^r + V_0^{r'}$. This is helpful when SNPs near position i (which therefore will fall within shared tracts involving i) have already been phased (by trio pre-phasing or previous iterations). It also helps in making the best decision when both haplotypes receive a majority of votes for the same allele, e.g., both have a majority of votes for 0. In this case, taking into account votes for the two haplotypes simultaneously will result in whichever has *more* votes getting assigned the actual value 0. If they each receive the exact same number of votes, then no allele will be assigned. This also avoids the dependency on the order in which the haplotypes are visited; the outcome is the same since votes for both are taken into account.

In this manner, M' is calculated at each position. If $M' = M$ (i.e. no changes were made) then the algorithm terminates. Otherwise, $M := M'$ (M is replaced by M') and another iteration is run.

2.2.3 Phasing the Individuals That Are Not a Part of the Largest Component

Individuals that are part of small connected components will have a number of ambiguous sites once they have been phased using the edges in their connected component. For these individuals, we compute a minimum number of recombinations and mutations from their haplotypes to others that have better phasing (belong to larger components). We then assign these haplotypes phase based on minimizing the number of mutations plus recombinations in a similar manner as the approach of Minichiello and Durbin (2006).

Alternatively this could be done in a sampling framework, where we sample the haplotype with a probability that is a function of the number of mutations and recombinations.

2.2.4 Results

We compared the correctness and learning rate of our algorithm against BEAGLE (Browning and Browning 2009) using a simulated dataset. Using the Hudson Simulator (Hudson 2002), we generated 3000 haplotypes each consisting of 3434 SNPs from chromosomes of length 10^5. We estimated a population size of 10^6 with a neutral mutation rate of 10^{-9}. To generate genotypes, we randomly sampled from the distribution of simulated haplotypes with replacement such that each haplotype was sampled on average 2, 3, and 4 times. We applied our algorithm and BEAGLE to the simulated data after combining haplotypes to create parent-offspring trio data (inspired by our analysis of the MS dataset). Both algorithms effectively phase the simulated dataset largely due to the initial trio phasing (Table 1). Our algorithm learns the true phasing at an increasing rate as the expectation of haplotypes sampled increases. The most clear example of this trend is in the Brown Long Range Phasing miscall rate. By weighing edges proportional to the length of sharing IBD rather than a fixed set of votes per edge, we achieve more accurate phasings.

	Population 1	Population 2	Population 3
BEAGLE miscall rate	0.0685%	0.0160%	0.00951%
Brown Long Range Phasing miscall rate	0.0501%	0.0148%	0.00503%
BEAGLE Error-free phasings	4467	6819	8898
Brown Long Range Phasing Error-free phasings	4459	6840	8923
Total haplotypes	4524	6870	8940

Table 2.1: We created three populations using a base pool of 3000 simulated haplotypes using the Hudson simulator. Populations 1, 2, and 3 were created by sampling each haplotype according to a geometric distribution with expectation 2, 3, and 4 respectively. Haplotypes were then randomly paired to create genotypes. The miscall rate is the ratio of 2's miscalled to total 2's (after trio phasing). Error-free phasings denote the number of haplotype phasings with zero miscalled 2's.

Chapter 3

Deletion Haplotypes and Autism

3.1 Introduction

The understanding of the genetic determinants of complex disease is undergoing a paradigm shift. Genetic heterogeneity of rare mutations with severe effects is more commonly being viewed as a major component of disease (McClellan and King 2010). Phenotypic heterogeneity – a large collection of individually rare or personal conditions – is associated with a higher genetic heterogeneity than previously assumed. This heterogeneity spectrum can be summarized as follows: (i) individually rare mutations collectively explain a large portion of complex disease; (ii) a single gene may contain many severe but rare mutations in unrelated individuals; (iii) the same mutation may lead to different clinical conditions in different individuals; (iv) mutations in different genes in the same pathways or related broader pathways may lead to same disorder or disorder family (McClellan and King 2010).

3.1.1 Genetic heterogeneity in autism

Autism spectrum disorders (ASD) are an excellent example of where research is active in identifying matches between the phenotypic and genomic heterogeneities (Bruining et al. 2010). A considerable portion of autism appears to be correlated with rare point mutations, deletions, duplications and larger chromosomal abnormalities including a disproportionately high rate of *de novo* large ($>$ 100 kb) deletions and duplications (Morrow 2010). Rare severe mutations in multiple genes important in brain development such as NRXN1, CNTN4, CNTNAP2, NLGN4, DPP10 and SHANK3 have been identified in patients with ASD (Ching et al. 2010; Glessner et al. 2009; Guilmatre et al.

2009; McClellan and King 2010; Sebat et al. 2007; Walsh, Morrow, and Rubenstein 2008). Furthermore, large *recurrent* structural mutations in genomic "hotspots", such as in chromosomal regions 1q21.1, 15q11-q13, 16p11.2 and 22q11.21, have been shown to be associated with autism and other psychiatric diseases (Mefford and Eichler 2009; Morrow 2010; Sanders et al. 2011).

Due to the size and growth rate of the human population, nearly all viable single nucleotide polymorphisms (SNPs) are likely present in some individual; however, most point mutations are rare and occur in low frequencies (a single individual or family). The large majority of such mutations have no functional significance and persist by chance in the absence of selective pressures. In contrast, mutations with significant deleterious effects on fertility (e.g. in some cases of severe autism) are less frequently transmitted to subsequent generations. It follows that severe mutations are disproportionately *de novo* and individually rare (McClellan and King 2010).

3.1.2 Deletion polymorphism in autism

A number of experimental and computational methods exist that can efficiently infer large and rare deletions. Deletions of this type have exhibited a significant role in many diseases particularly in autism where recent studies of simplex families suggest 7%-10% of autistic children have a variety of large *de novo* deletions (Weiss et al. 2008). Examples of deletions in autism include highly penetrant chromosomal deletions in regions that affect many genes (e.g. 22q11.2) and large deletions which implicate few genes (e.g. DIA1 or NRXN1) (Morrow 2010; Morrow et al. 2008). The detection of such variants has also been used successfully in finding deletions associated with schizophrenia (Stefansson et al. 2008). While thousands of deletions have been cataloged with various platforms (Fiegler et al. 2006; Khaja et al. 2006; Mills et al. 2006; Stefansson et al. 2008) and deposited into the Toronto Database of Genomic Variants (Iafrate et al. 2004), the vast majority are large and rare partly due to the lack of a reliable methodology for the detection of small deletions.

In the context of genetic heterogeneity, compound heterozygosity and other phase-dependent interactions between small deletion variants have been shown to play a role in complex disease (Hague et al. 2003). Furthermore, deletion variants may also be involved in loss of heterozygosity and uniparental disomy events, both of which may be genetic determinants in the development of disease (Stefansson et al. 2008). Each of these examples may include smaller deletion polymorphisms which are commonly overlooked by GWAS as they are not directly probed by SNP arrays and difficult to infer from high-throughput sequence data. However, three main categories of computational methods for inferring small deletions have been developed each associated with their own strengths and weaknesses.

3.2 Methods for identifying deletion polymorphism

3.2.1 Intensity-based

Intensity-based methods may be employed on SNP arrays or custom designed fine-tiling arrays (Wang et al. 2007; Zerr et al. 2010). Because probe intensities are noisy, both SNP and fine-tiling arrays require many probes to span the deletion for accurate measurement. Intensities from SNP arrays can extend to genome-wide data but have difficulties inferring small deletions due to the wide spacing of tag SNPs. Fine-tiling arrays provide a higher resolution for detecting small deletions but are not in widespread use and are prohibitively expensive to implement for genome-wide data.

3.2.2 Sequence-based

Sequence-based algorithms first map sequence reads to a reference chromosome and then use coverage estimates and mapping statistics to identify deletions (Medvedev, Stanciu, and Brudno 2009; Mills et al. 2011). While regions of sparse read mappings may indicate the presence of a deletion, these methods suffer from high false positive rates originating from regions that cannot be sequenced or mapped with reads and inherent biases in the choice and assembly quality of the reference genome. Additionally, as the sampling from high-throughput sequencers is not always random across the genome, the problem of inferring deletions is conflated with the problem of detecting sampling bias, particularly for hemizygous deletions.

3.2.3 Pedigree-based

The final category of algorithms is based on deletion inference from genotype data with a familial structure. These *SNP-based* methods use genotype data to probe for specific genomic inheritance events that suggest inherited or *de novo* deletion polymorphisms. The key insight lies within inheritance patterns where an individual should be heterozygous for a SNP allele according to the laws of Mendelian inheritance, but is not. The deletion inference method employed here, as well as previously published methods (Conrad et al. 2006; McCarroll et al. 2005), relies on the fact that the SNP calling algorithm for SNP arrays and sequence data cannot distinguish between an individual who is homozygous for some allele a and an individual who has a deletion haplotype and the allele a (Fig. 3.1). Hemizygous deletions can then be inferred by finding such genotypic events throughout the data and analyzing their relationships to each other.

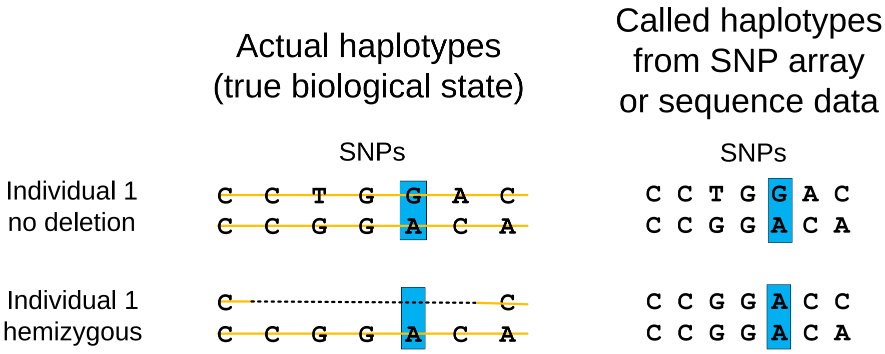


Figure 3.1: Alleles in the genomic interval of a hemizygous deletion are interpreted as homozygous by modern technologies. For example, individual 1 is correctly called heterozygous at the blue SNP position in the absence of a deletion but, if individual 1 is hemizygous, then each SNP will be called homozygous throughout the span of the deletion. This is true for SNP array (the intensities of only one probe is processed) and high-throughput sequencing technologies (sequence reads are sampled from a single chromosome).

Previously developed SNP-based methods were applied to the SNP array HapMap data (International HapMap Consortium 2003) containing a considerably fewer number of individuals than current GWAS data (albeit with more SNPs). These methods do not consider multiple individuals and thus have difficulties inferring recurrent deletions that may be associated with disease in association study data. However, a major benefit of SNP-based algorithms is that they extend to genome-wide data and are not restricted to operate on SNP arrays; on the contrary, they have higher power to infer deletions from SNP calls on high-throughput sequencing data. Another considerable benefit of these approaches is that they are largely orthogonal to deletion inference from intensity-based and sequence-based methods and can hence be used in conjunction with those methods to control type I and type II error.

3.3 Prior work on genome-wide deletion maps

Several algorithms exist capable of producing genome-wide deletion maps. McCarroll et al. (2005) developed a combinatorial clustering approach to identify sets of aberrant genotype inheritance patterns for dense genome-wide HapMap data. Conrad et al. (2006) first classifies SNP genotypes into several categories of Mendelian inheritance. They then iterate over all individuals separately and search for several sites that provide strong evidence of a deletion near each other. Both of these algorithms consider a single individual during deletion inference which is effective at finding large deletions. However, these algorithms are underpowered when considering data containing small recurrent deletions. Corona et al. (2007) developed an algorithm aiming to support recurrent deletion calling by estimating haplotype frequencies assuming the presence or absence of a deletion in a window. This algorithm, however, phases the data first and the Mendelian inconsistencies caused by genomic deletions create difficulties for haplotype phasing algorithms. In fact, haplotype phasing algorithms generally convert all Mendelian inconsistencies to missing data prior to phasing thereby removing the deletion signal from the data. In Halldórsson et al. (2011) we presented an algorithm that called deletions based on a maximum clique finding heuristic algorithm. Although the run-time of this algorithm was acceptable for GWAS data, we found it was missing deletion calls in genomic regions of complex deletion signature. All of these methods employ heuristics and can miss small deletions that may be conserved among a few individuals in the sample.

Aside from algorithms that exclusively use SNP data, a number of different technologies have been used to determine deletions and other copy number variations (CNVs) throughout the human genome. Conrad et al. (2009) used tiling arrays to identify 8888 (7075 unique) CNVs. Park et al. (2010) employed a combination of a tiling array and resequencing to determine CNVs in an Asian population. Levy et al. (2007) identified a number of CNVs from the sequencing of a single individual. The 1000 Genomes Project has worked on identifying CNVs from the sequencing of a subset of one thousand individuals (Siva 2008). There have also been SNP arrays developed to specifically target CNVs (Halldórsson and Gudbjartsson 2011). These methods represent orthogonal analyses and can be used alongside SNP-based methods to infer deletions.

In this dissertation, we present a SNP-based algorithmic framework for genome-wide hemizygous deletion inference termed DELISHUS (**del**etions **i**n **s**hared **h**aplotypes **u**sing **S**NPs). We model the input SNP data using graph theory and implement efficient and exact algorithms to call genomic deletions based on biological conservation of a pattern of Mendelian inconsistency. Because our algorithms consider all individuals in the sample simultaneously, they achieve significantly lower false positive rates and higher power when compared to previously published algorithms. By slightly modifying the model, we also present an algorithm for detecting *de novo* deletions. After deletions are called, we employ a similar graph theoretic approach for computing the critical regions of recurrent deletions in polynomial time algorithm. We also present a human genome deletion map of the Autism Genetic Resource Exchange (AGRE) GWAS data (Supplemental Figure 1). Our algorithmic strategy is based on a combination of (1) using deletion conservation across many individuals to benefit from recurrent deletions in the population; (2) modeling the input with graph theory and bounding the number deletion calls by a polynomial; (3) implementing an exact backtracking algorithm which completes its computation on a GWAS sized dataset in a few minutes due to a sparsity condition in the data. These three stringent requirements provide a rigorous basis for extracting genomic deletions of all sizes from the abundant SNP data available from high-throughput sequencing and array technologies.

3.4 Definitions and terminology

The input to our algorithm is an $m \times n$ genotype matrix M. The rows of M correspond to sets of related individuals and we assume that for every individual i there exists at least one other individual j such that i and j share a haplotype. In practice, M frequently consists of parent-child pairs or parents-child trios from a family-based association study design. The columns of M correspond to SNP calls for the m individuals. The genotype data are commonly obtained with SNP arrays but are increasingly acquired from whole-exome or whole-genome sequence data that provide SNP calls at a high resolution; consequently, this allows the detection of smaller or less frequent deletions.

Mendelian inheritance patterns in M can be divided into three major categories (Fig. 3.2). If an inheritance pattern can be explained only by the introduction of a deletion haplotype or a SNP call error, then we call it *evidence of deletion.* If the pattern can be explained by introducing a deletion haplotype or SNP call error but follows the laws of normal Mendelian inheritance, then we call it *consistent with a deletion.* Finally, if the pattern cannot be explained by introducing an inherited deletion haplotype then we call it *no deletion.*

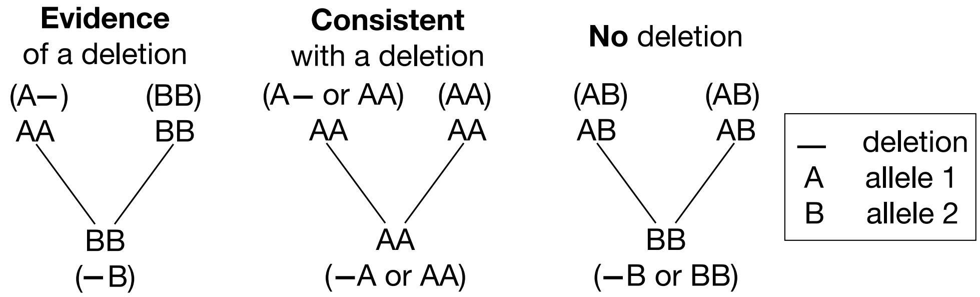


Figure 3.2: Each trio inheritance pattern can be classified into three categories under the interpretation of inherited deletions. The evidence of deletion pattern provides evidence for the presence of an inherited deletion. The no deletion pattern provides evidence for the absence of a deletion. The consistent with a deletion pattern does not provide strong evidence for the presence or absence of a deletion.

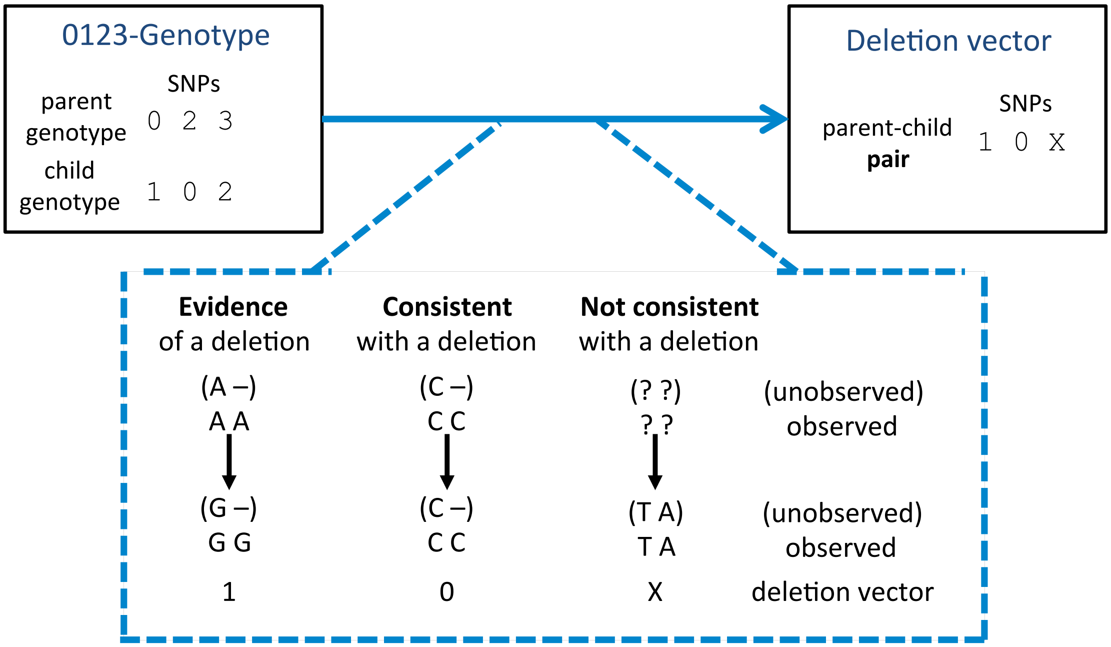


Figure 3.3: Parent-child pairs of encoded genotypes are converted to deletion vectors according to Mendelian inheritance patterns that show evidence of a deletion, are consistent with a deletion, or are not consistent with a deletion. Informally, evidence of deletion sites can only be explained by introduction of a sequencing error or deletion polymorphism. Consistent with a deletion sites can be explained by a deletion or normal Mendelian inheritance. Not consistent with a deletion sites cannot be explained with a deletion.

3.5 Identification of inherited deletions

We assume, for ease of exposition, M consists of trio data (in general, individuals in M can be any type of parent-child designation). DELISHUS first converts M to a new matrix M' with $\frac{m}{3}$ rows and n columns. Each row of M' corresponds to a trio and each column corresponds to a trio-SNP inheritance pattern. Let the value of the (i, j) cell be denoted $M'_{i,j}$. Then $M'_{i,j} \in \{0, 1, X\}$ where

- $M'_{i,j} = 1$ if the i^{th} trio exhibits an evidence of deletion inheritance pattern at SNP j.
- $M'_{i,j} = 0$ if the i^{th} trio exhibits a consistent with a deletion inheritance pattern at SNP j.
- $M'_{i,j} = X$ if the i^{th} trio exhibits a no deletion inheritance pattern at SNP j.

The rows of M' are termed *deletion vectors*. Figure A.0.2 gives an in-depth overview of the genome to deletion vector relationship. Figure 3.3 shows the translation from genotypes to deletion vectors.

DELISHUS then constructs a graph $G(V, E)$ based on M'. A node is introduced for each evidence of deletion site and an edge between two nodes signifies that both nodes can be explained by the same deletion; formally, let $v_{i,j}$ denote the vertex associated with row i and column j, then $v_{i,j} \in V$

if $M'_{i,j} = 1$ and $(v_{i,j}, v_{k,l}) \in E$ if the ranges $[M'_{i,j}, M'_{i,l}]$ and $[M'_{k,l}, M'_{k,j}]$ contain no X. In this graph, two nodes that are connected can be explained by the same deletion polymorphism and are termed *compatible.* Therefore, dense subgraphs of G correspond to genomic regions that are likely to contain inherited deletions. However, this picture is complicated by the fact that deletions may occur in a region of the genome independently and at slightly different intervals.

3.5.1 Minimum number of errors

We present an exponential algorithm and a greedy heuristic for computing putative deletions. Both algorithms begin by parsing M and removing SNPs in which the Mendelian error rate is above 5% to remove artifacts from genotyping. We then calculate the deletion vector for each trio in the dataset which corresponds to using the table defined in Fig. 3.2 (Right) to translate each SNP site. This new matrix is denoted $N^{\left(\frac{\lfloor m \rfloor}{3} \times n\right)}$. To identify the genotyping errors and putative deletions, we define two operations on the evidence of deletion sites of N: error correction call and deletion haplotype call. An error correction call will categorize an evidence of deletion as a genotyping error effectively removing it from any particular deletion haplotype. A deletion haplotype call will identify a putative deletion as an inherited deletion haplotype. We infer inherited deletion haplotypes using the objective function

$$min_N \; (k_1 * (\text{genotype error corrections calls}) + k_2 * (\text{deletion haplotypes calls}))$$

where k_1 and k_2 are weighing factors. k_1 and k_2 can be simple constant factors or a more complex distribution. For example, setting k_1 to 2 and k_2 to 7, we will prefer calling a putative deletion with at least 4 pairwise compatible evidence of deletion sites an inherited deletion. The parameters must be tuned to the input data. In the case of the Multiple Sclerosis dataset, the matrix N contains small overlapping putative deletions and over 95% of N is non-putative deletions, that is, N is very sparse.

We start by giving an exact exponential algorithm which minimizes the objective function. Let x_i denote a set of overlapping putative deletions.

1. For sparse N we can reduce the minimization function from min_N to $min_{x_1..x_s}$ where $x_1..x_s \in N$ and $\{x_1..x_s\} \subseteq N$.

2. Since any particular putative deletion is defined by the evidence of deletion sites, we can enumerate all feasible non-empty sets of evidence of deletion sites for all x_i.

Computing this for all putative deletions demands work proportional to $\sum_{i=1}^{s} B(e_i)$ where e_i is the number of evidence of deletion sites in x_i and B is the Bell number. In practice, we found that e_i is bounded by a small constant but this complexity is still unreasonable for most e_i.

For practical purposes, we've developed a greedy heuristic algorithm for cases where the exact exponential algorithm is infeasible (Fig. 3.4).

1. For each $x_i \in N$, the algorithm selects the component with the maximum *trio sharing*, that is, the possibly overlapping putative deletions that include the most evidence of deletion sites. Because every two evidence of deletion sites in an inherited deletion must be pairwise compatible, this component is a clique.

2. To find the maximum clique, we construct an overlap graph $G(V, E)$ where $h_i \in V$ if h_i is an evidence of deletion in a putative deletion in this interval and $(h_i, h_j) \in E$ if h_i and h_j are compatible.

3. We find maximum cliques using a greedy approach that iterates over a queue containing the compatible vertices, selecting the highest degree node v_m and adding it to the potential clique set if and only there is an edge between v_m and each vertex in the clique.

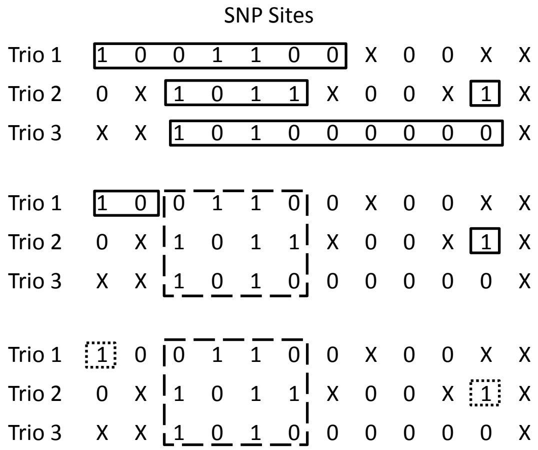
4. At the end of this process, the algorithm calls the site(s) a deletion haplotype or genotyping error according to the objective function, clears the set, and continues until all vertices in the queue are processed.

Experimental Results on Simulated Data

We tested the algorithm using the same simulated dataset used to test our phasing algorithm. To simulate and score an error-prone GWAS dataset containing a deletion, we define six parameters, two metrics, and generate only one deletion in the genotype matrix (Table 2). We randomly select a set of trios and an interval in the simulated haplotype matrix to contain the generated deletion. After the site is selected, we place evidence of deletion sites on the SNPs according to some probability (assumed independent for each SNP in the interval).

We observe promising results with our deletion model. Our algorithm is sensitive to inherited deletions that are very short but shared among many individuals and also sensitive to inherited deletions that are longer and shared by few people.

In general, the algorithm is accurate when the coefficient of deletion call and genotype error call are tuned well (Table 3 – parameter sets 1-3). For a dataset with low genotyping error rate



SNP Sites

Trio 1 1 0 0 1 1 0 0 X 0 0 X X
Trio 2 0 X 1 0 1 1 X 0 0 X 1 X
Trio 3 X X 1 0 1 0 0 0 0 0 0 X

Trio 1 1 0 0 1 1 0 0 X 0 0 X X
Trio 2 0 X 1 0 1 1 X 0 0 X 1 X
Trio 3 X X 1 0 1 0 0 0 0 0 0 X

Trio 1 1 0 0 1 1 0 0 X 0 0 X X
Trio 2 0 X 1 0 1 1 X 0 0 X 1 X
Trio 3 X X 1 0 1 0 0 0 0 0 0 X

Figure 3.4: A visual depiction of the greedy algorithm for finding putative deletions (consistencies with particular parents are omitted for simplicity). The solid rectangles denote trio SNP sites which have not been called yet. The dashed rectangle denotes a called inherited deletion haplotype. A dotted rectangle denotes a genotype error call. First, the algorithm finds the submatrix (a clique in G(V,E)) with the maximum trio sharing: SNP sites 3-6. Using the objective function, the algorithm either calls the set of SNPs an inherited deletion or a set of genotyping errors (in this case the former). The intervals are updated by removing vertices and edges from the overlap graph and the algorithm continues. Both remaining subgraphs consisting of SNP sites 1 and 11 are both cliques of size one. These will most likely be called genotyping errors.

Param Set	Site Error Prob.	Interval Length	Trios in Deletion	Prob. of ED	Coeff. of Deletion	True Positive	False Positive	Runs
1	0.0001	5	5	0.75	11	1000	0	1000
2	0.0001	2	5	1	11	1000	0	1000
3	0.0001	9	3	0.75	11	1000	0	1000
4	0.0001	7	3	0.50	15	58	0	100
5	0.00333	9	3	0.75	15	100	38888	100

Table 3.1: We tested our deletion inference algorithm using the six tunable parameters as defined in Table B.0.1. Each configuration was run with a coefficient of genotyping error of 2. Evidence of deletion is denoted ED.

(∼0.0001 site error probability), the coefficient of deletion call can be set low; if it is set too high, a true inherited deletion may be incorrectly called a genotyping error, possibly missing an evidence of deletion (Table 3 – parameter set 4). A similar caveat pertains to datasets with significant genotyping error rates (for instance, the MS dataset). A coefficient of deletion call that is too low can yield false positives (Table 3 – parameter set 5). Finding appropriate tuning mechanisms for the two coefficients to maximize algorithm specificity and sensitivity will be the subject of future work.

3.5.2 Exact algorithm

Each vertex in G may be a member of many different dense subgraphs and thus we formulate the problem of identifying deletions as follows:

Formulation 1. *For each connected component $d \in G$ and for each set of vertices that form a maximal clique C in d, report C as deleted if $|C| \geq k$ where k is some threshold of evidence. Report a subset of vertices in C as genotyping errors if they are not members of at least one deletion.*

In the absence of genotyping or sequencing errors, each evidence of deletion site would indicate a hemizygous deletion. In real data, random errors create false positives and the threshold k is tuned to lift predictions above the noise level by enforcing a minimum number of evidence of deletion sites to commit to a deletion. In particular, the value for k is guided by false positive rate and power analysis experiments specifically tuned for a specific dataset. Formulation 1 computes all maximal cliques which, in G, correspond to rectangular areas of M' whose evidence of deletion sites reinforce each other. It takes exponential time to compute and output all maximal cliques in a general graph, however, G has a special structure that allows us to achieve polynomial-time algorithms.

Lemma 1. *G contains at most $\binom{n+1}{2}$ maximal cliques.*

Proof. Let C be a set of vertices forming a maximal clique in G. Let the interval of C be I_C as defined by the span of SNPs from the leftmost evidence of deletion site of C (denoted l) to the rightmost evidence of deletion site of C (denoted r). We say C induces the interval of SNPs I_C.

Because C is maximal, there cannot exist a vertex $v \notin C$ such that v is compatible with every vertex of C, thus I_C cannot be extended. Furthermore, a maximal clique distinct from C but inducing I_C cannot exist because each of its vertices must be compatible in the interval $[l, r]$ which is in violation of the maximality of C. It follows that no maximal clique other than C can induce I_C; thus, each maximal clique uniquely defines an interval. Since $\binom{n+1}{2}$ distinct intervals exist for any given M', the statement follows.

■

Figure 3.5 gives an illustration of Lemma 1 on an example M' and G.

Because of Lemma 1, G has a polynomial number of maximal cliques. As the n of a larger chromosome can be several hundreds of thousands, this may still be prohibitively large. A more precise bound can be computed by observing that we only consider columns with at least one 1. Let n_1 be the number of columns containing at least one 1, therefore the number of maximal cliques is at most $\binom{n_1+1}{2}$. But, if non-overlapping sections of the matrix exist, we consider connected components

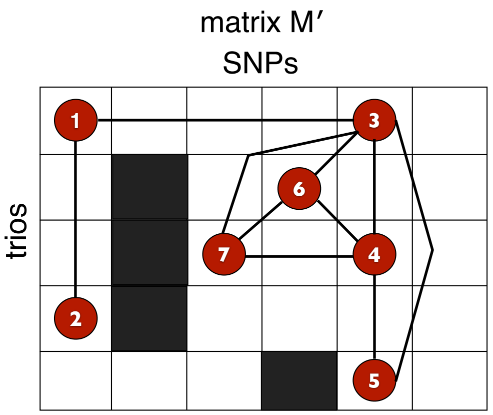


Figure 3.5: The outline of the matrix M' is given with the red vertices corresponding to evidence of deletion sites in G. Four maximal cliques are formed namely, {1,2},{1,3},{3,4,5} and {3,4,6,7}. Each maximal clique induces an interval which is the shortest such interval associated to the vertex set.

separately; let d_i be the i^{th} connected component of the set of all components D and n_{d_i} be the number of columns with at least one 1 in the SNPs of d_i.

$$\textit{number of maximal cliques} \leq \sum_{i=1}^{|D|} \binom{n_{d_i}+1}{2}$$

We call the matrix M' sparse if the number of connected components is large. A sparse M' allows for trivial parallelization of deletion inference on distinct connected components and efficient computations due to the component sizes being small. Table 3.2 shows that the probability of evidence of deletion sites is low while the probability of a no deletion site is high for the HapMap and AGRE data. This suggests that M' contains few deletion intervals compared to non-deleted intervals and thus M' is sparse and D is large. This follows the intuition that the emergence of deletion polymorphisms are typically infrequent events.

Data	Evidence of deletion	No deletion
HapMap P1	5.89×10^{-4}	0.30
HapMap P2+3	2.78×10^{-4}	0.18
AGRE autism	1.21×10^{-4}	0.41

Table 3.2: The probabilities of an evidence of deletion site and a no deletion site for HapMap and autism GWAS data suggests M' is sparse.

Tsukiyama et al. (1977) presented an output sensitive algorithm that computes all maximal cliques of a component d with edges e in time $O(de)$ per clique generated.

Corollary 1. *Computing all genomic deletions of M' using Formulation 1 can be done in polynomial time.*

In practice, however, the Bron-Kerbosch algorithm for maximal clique computation has proven to be more efficient. The Bron-Kerbosch algorithm is a recursive backtracking algorithm that computes all maximal cliques in an undirected graph but is not guaranteed to run in polynomial time. Although the Bron-Kerbosch algorithm is not an output-sensitive algorithm, it is still widely considered the fastest maximal clique finding algorithm (Cazals and Karande 2008; Harley 2004). Also, through empirical observations, the components of G are chordal with high probability. When a component of G is chordal, we can compute all maximal cliques even faster by simply generating a perfect elimination ordering.

With complex genetic heterogeneity (e.g. compound heterozygosity of small deletions), it is likely most informative to compute all possible configurations of deletions. Each maximal clique can be tested for association to disease if the data has a special structure. For example, the AGRE autism dataset includes families with a mixture of children diagnosed with autism and children without the disorder treated as healthy controls. DELISHUS computes the deletion transmission rates of parents to children with autism and parents to children whom are healthy; these deletion calls and transmission rates can be used to prioritize variants based on a number of statistical tests. This extra phenotypic information helps resolve situations where multiple deletion configurations are possible in the data (Fig. 3.6) and guides the deletion calls towards disease relevancy.

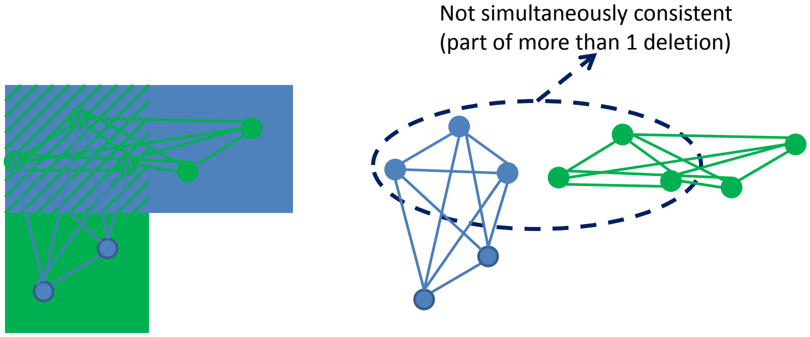


Figure 3.6: M' is shown on the left with a superimposition of evidence of deletion vertices and edge connections. On the right, two maximal cliques are shown that share a subset of evidence of deletion sites. If the threshold $k \leq 5$, DELISHUS would report both cliques as potential deletions.

3.6 Results

Formulation 1 also enables the resolution of complex genomic deletion "hot-spot" regions. These regions (e.g. 22q11.21) pose the difficult problem of sorting through many possible configurations of deletions. DELISHUS can identify and process every deletion separately to resolve these complexity regions. Using this formulation, we called inherited deletions from the AGRE autism GWAS data and produced a high level deletion map of autism (Figure A.0.3). Table 3.3 demonstrates that DELISHUS is capable of efficiently resolving these regions for genome-wide data.

Data	Runtime (s)	Memory (GB)
HapMap P1 CEU	71.5	< 1
HapMap P2+3 CEU	91	< 1
AGRE autism	139.8	1.6

Table 3.3: We ran DELISHUS using Formulation 1 on HapMap P1, P2+3, and the AGRE autism data. The HapMap P1 CEU data consists of 90 genotypes with about 1 million SNPs. The HapMap P2+3 CEU data consists of 174 genotypes with about 4 million SNPs. The AGRE data includes 4327 genotypes with about 500k SNPs. We show DELISHUS scales to current GWAS sized data by presenting the runtime and memory requirements for the AGRE autism data. We ran DELISHUS on each chromosome in parallel on a cluster of 23 nodes. The numbers reported are the maximum requirements for a single machine in the computing cluster.

However, if evidence of deletion sites must be committed to exactly zero or one deletion, we can iteratively remove the largest clique of all maximal cliques in the component. More precisely, if the cardinality of a maximal clique is $\geq k$, we call the associated intervals deleted and remove the corresponding vertices from the graph. Statistical models that score deletions based on other quantities, such as deletion length or allele frequencies, may be used to provide a different ordering for the maximal clique processing. For example, if deletion length were the most important statistic, the green clique in Fig. 3.6 would be preferable to the blue clique. This procedure is iterated until each evidence of deletion site has been called as part of a deletion or a SNP calling error.

3.6.1 Assessing the false positive rate

Our algorithm uses enrichment of compatible evidence of deletion sites from many individuals to infer deletions. While inherited deletions are certainly a cause for evidence of deletion sites, these sites may also arise from genotyping or sequencing errors. To assess the false positive rate occurring from random error, we computed the distribution of evidence, consistent, and no deletion sites across three datasets: HapMap Phase 1 CEU, HapMap Phase 2+3 CEU and the AGRE autism data. We simulated a chromosome of length 25000 SNPs with 30, 58, and 2500 parent-child trios for the HapMap Phase 1, HapMap Phase 2+3, and AGRE autism data respectively. The inheritance patterns are drawn independently at random according to the distribution defined by each dataset. We ran this simulation at different thresholds for 1000 iterations. These computations are conservative because the evidence of deletion probabilities were computed from the entirety of the HapMap data including sites that may arise from both SNP calling errors and true genomic deletions.

The false positive rate depends on the density of the SNP array, the sample size of trios, and the probabilities of Mendelian inheritance patterns. In the smaller HapMap data, DELISHUS produces very few false positives at a threshold of 3. In the larger AGRE autism data, DELISHUS requires a threshold of 5 to produce similar false positive rates. In contrast, when DELISHUS is tuned to reproduce the results of Conrad et al. (2006) by considering each individual independently (identified as the single individual algorithm), a threshold of 2 and 3 yields similar false positive rates for both the HapMap and autism data. Table 3.4 summarizes these computations.

T	D P1	D P2+3	D AGRE	SI P1	SI AGRE
2	8.528	10.356	1214.063	0.701	1.854
3	0.076	0.135	141.13	0.001	0.001
4	0	0.001	11.274	0	0
5	0	0	0.632	0	0
6	0	0	0.028	0	0
7	0	0	0	0	0

Table 3.4: We simulated 25000 independent and identically distributed trio inheritance patterns according to the distribution observed in the data. The HapMap P1 CEU, P2+3 CEU, and AGRE autism data were simulated with 30, 58, and 2500 trios respectively. We inferred deletions using different thresholds (T) for DELISHUS (D) and the single individual (SI) algorithms. The statistic calculated for the false positive rate is the average amount of deletions detected in 1000 iterations for the HapMap Phase 1 (P1), Phase 2+3 (P2), and AGRE autism GWAS data.

It is difficult to simulate false positives that may arise from technical artifacts, SNPs that are poorly genotyped, or SNPs that are undersampled from sequence reads. If such a SNP passes quality control, we may detect the error by observing the distribution of Mendelian errors. Mendelian errors can be placed into two categories: those that show evidence of a deletion and those that do not. We assume there is no bias toward producing genotyping errors in either category. Even though evidence of deletion Mendelian errors are more probable, we would still expect to find non-evidence of deletion Mendelian errors for poorly genotyped SNPs. For these reasons, we may filter out SNP sites with many non-evidence of deletion Mendelian errors to reduce false positive rates from systematic errors. Conservative approaches may further filter deletions that feature only one SNP containing evidence of deletion sites regardless of the Mendelian error distribution.

3.6.2 Estimating statistical power

The power to correctly infer deletions is a function of three variables: (1) the number of probes, distance between probes, or size of the deletion, (2) the frequency of the deletion in the population, and (3) the allele frequencies. To estimate the power for predicting deletions we use the HapMap Phase 1 CEU, Phase2+3 CEU, and AGRE autism data; this selection fixes the allele frequencies. When compute the size of the deletion in base pairs, we select a genomic position at random and extend this interval for the defined size of the deletion. Therefore, it is possible for smaller deletions to be missed by the data completely if no SNPs exist within the deleted interval. We can also compute the size of a deletion in SNPs for which we randomly select a SNP and extend the deletion interval appropriately. In this case, there is always at least 1 SNP in the interval of the deletion. We varied the sizes of the deletions between 1 bp and 1 Mb or 1 and 20 SNPs and randomly selected 3 individuals in the HapMap data and 5 individuals in the AGRE autism data to harbor the deletions. To simulate the deletion, the genotypes of the child and a randomly selected parent were altered to indicate an inherited deletion. That is, the alleles of the child and selected parent were changed to homozygous for the non transmitted allele in the span of the deletion. A deletion is said to be detected if the algorithm correctly reports a deletion for that specific trio. For example, if DELISHUS detects 3 individuals having a deletion within the simulated deleted region in the AGRE autism data, it will have detected $^3/_5$ of the deletion.

We tested the power of the DELISHUS algorithm to detect inherited deletions within simulated intervals of various sizes in the HapMap P2+3 CEU data (Fig. 3.7 Top). In general, algorithms that infer deletions from SNP data have reduced power to infer deletions if only one parent is genotyped. This is also true of X chromosome deletions compared to the autosomes; the SNP calls for deleted haplotypes are less predictable and usually result in missing data. However, it is still feasible to call X chromosome deletions passed from mother to daughter. Due to the density of the data, our algorithm can robustly detect small deletions in the autosomes and performs fairly well on the X chromosome.

We then compare the power of the DELISHUS algorithm and the single individual algorithm for the HapMap P1 CEU data (Fig. 3.7 Bottom). This data is roughly one-quarter as dense but useful for comparison of smaller sample sizes; it is also the same data used by Conrad et al. (2006). There is a clear trade-off between false positive rates and algorithmic power to detect deletions. However, when tuning the algorithms to achieve similar false positive rates, the DELISHUS algorithm clearly outperforms the single individual algorithm due, in part, to leveraging the genomic information of the entire sample during inference.

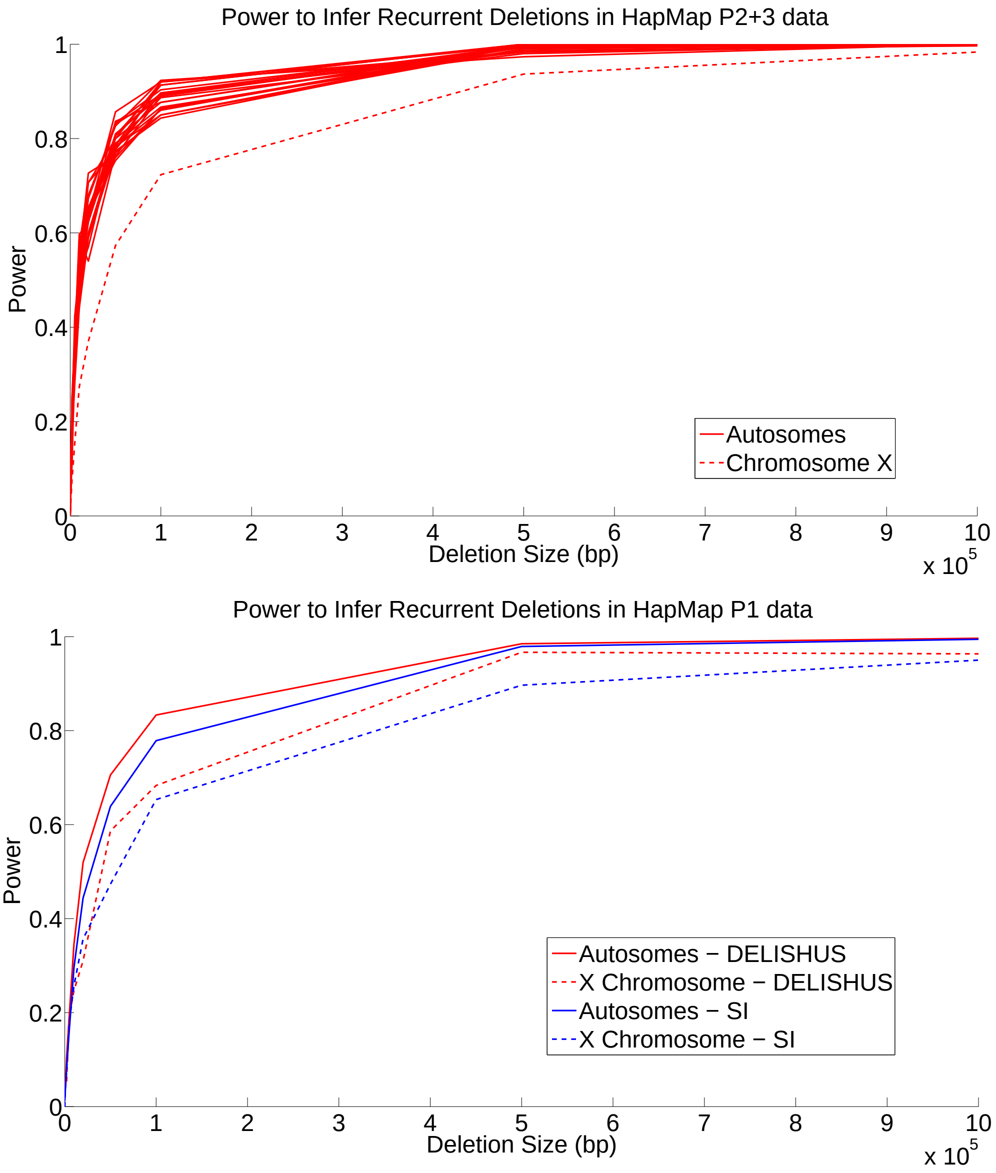


Figure 3.7: Top: The power to infer deletions in the HapMap Phase 2+3 CEU data as a function of the number of base pairs in the deletion. Bottom: We compare the power of the DELISHUS and single individual algorithms on HapMap Phase 1 CEU data. We average the power over all autosomes as they produced a similar curve. There is less power to predict deletions on chromosome X due to the male having only a single X chromosome. This power calculation was repeated 100 times for each autosome and then averaged. In both figures, the threshold of the DELISHUS algorithm was set to 3 and calibrated using the false positive rate calculations of the previous section. Also a total of three individuals were selected at random to harbor the genomic deletion.

Current association studies feature about as many SNPs as the HapMap data but many more individuals. Considering this, we applied the DELISHUS and single individual algorithms to the

AGRE autism data (Fig. 3.8 Top). Five trios were selected at random (from the set of about 2500 trios) and a random interval was deleted. Using conservative thresholds, the DELISHUS algorithm is much more sensitive than the single individual algorithm. DELISHUS excels at inferring recurrent small deletions but the power of the two algorithms eventually converges as the deleted genomic interval increases. This proposition is highlighted in Fig. 3.8 Bottom where we inspect small deletions at a high resolution. The trend for the X chromosome is similar to the autosomes and is omitted.

Power to infer deletions is also a function of deletion frequency. After increasing the frequency of the deletion in the sample from 0.2% to 1%, the power of the DELISHUS algorithm increases significantly and notably for smaller deletions (Fig. 3.9).

3.7 Identification of *de novo* deletions

Recent studies have highlighted the importance of protein altering *de novo* mutations for neural developmental disorders like autism (O'Roak et al. 2011). Inferring *de novo* deletions in genotype data is more difficult due to the parent having a lower frequency of homozygous SNPs over the interval of the child's deletion. For instance, the no deletion pattern in Fig. 3.2 could be hiding an undetectable *de novo* deletion. Figure 3.10 shows the inheritance patterns for inherited and *de novo* deletions for a pair of individuals sharing a haplotype. The most obvious relationship between the two types of deletions is that there is a much higher probability of consistent with a deletion patterns when inferring *de novo* deletions. This causes G to become more connected and, in regions of deletion complexity, may cause DELISHUS to run in superpolynomial time. However, Lemma 1 still applies, thus this problem remains theoretically polynomial and empirical evidence suggests our algorithms are still efficient.

Table 3.5 shows false positive rates for the DELISHUS *de novo* deletion inference algorithm on the AGRE autism data. We do not observe a significant increase in the false positive rate because the probability of a no deletion site is only reduced slightly. If the probability of a no deletion site is high enough and the threshold is set to a large enough value, random genotyping errors cannot form enough compatible evidence of deletion sites to be called a deletion.

We have found many examples of *de novo* deletions in the autism AGRE dataset. Figure 3.11 shows the two different interpretations of M' using Fig. 3.10. Due to data usage rules, we have substituted the gene name. It is certainly the case that one larger *de novo* deletion is more likely than 3 smaller inherited deletions. In this case the *de novo* deletion becomes connected and not many

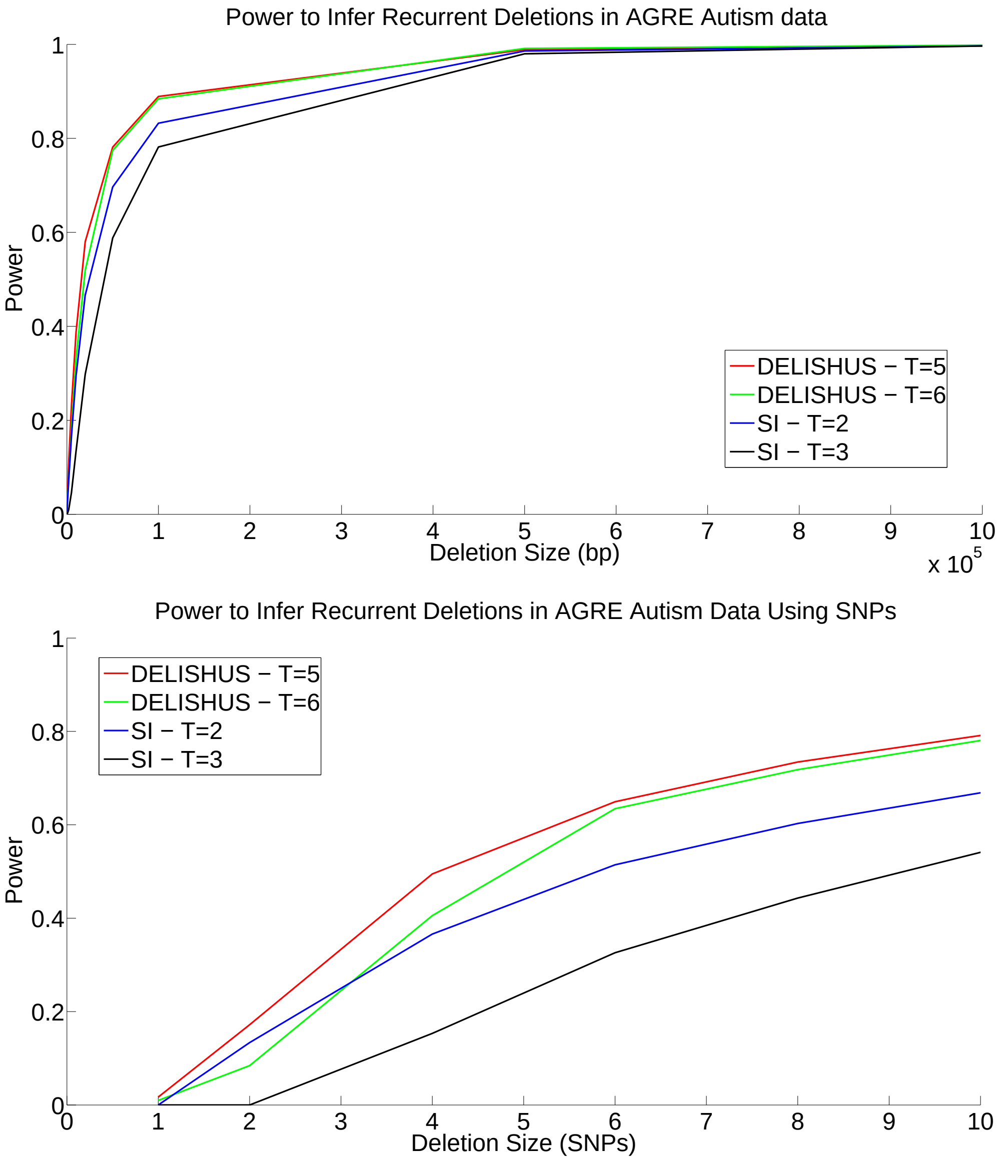


Figure 3.8: The power of the DELISHUS and single individual algorithms to infer inherited deletions in the AGRE autism autosomal data using (Top) a view of large deletions defined by basepairs and (Bottom) a higher resolution view for small deletions defined by SNPs. In both cases, a total of five individuals were chosen at random to harbor the deletion.

other SNPs become consistent with a deletion. In practice we do observe this same phenomenon which most likely occurs because the probability of no deletion is still sufficiently large.

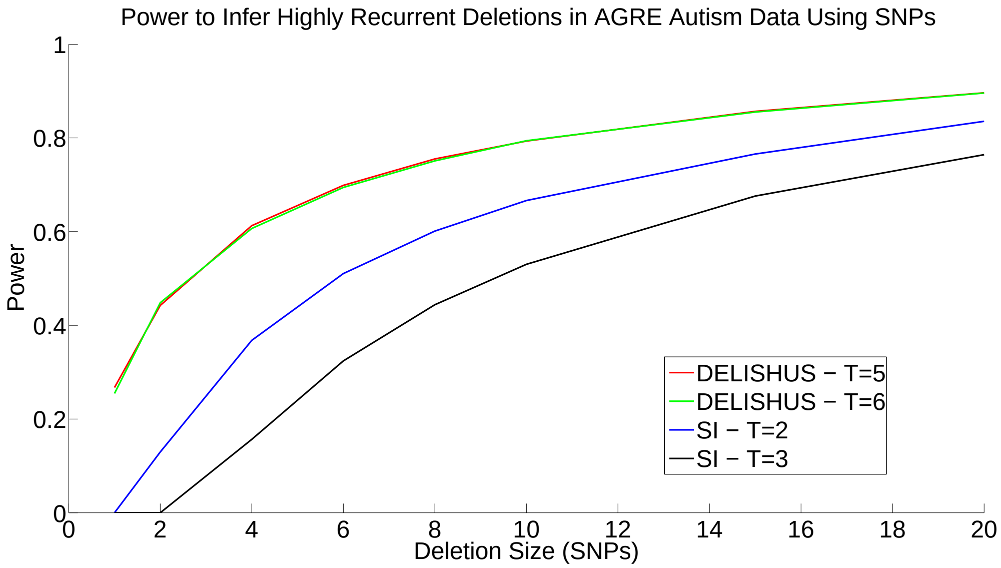
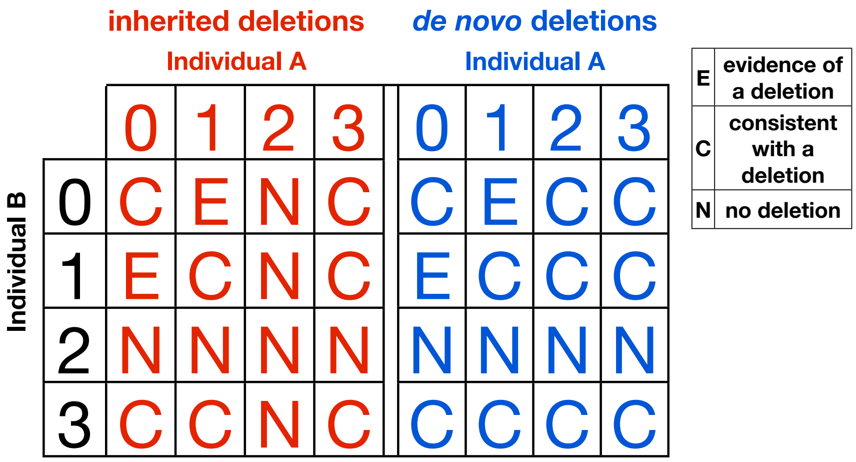


Figure 3.9: The power of the DELISHUS and single individual algorithms to infer highly recurrent small inherited deletions with a frequency of 1% (or 25 people) in the AGRE autism data.



	inherited deletions				*de novo* deletions			
	Individual A				Individual A			
Individual B	0	1	2	3	0	1	2	3
0	C	E	N	C	C	E	C	C
1	E	C	N	C	E	C	C	C
2	N	N	N	N	N	N	N	N
3	C	C	N	C	C	C	C	C

E	evidence of a deletion
C	consistent with a deletion
N	no deletion

Figure 3.10: Categories of inheritance between a pair of individuals sharing a haplotype for inherited and *de novo* (in individual B) deletions. To represent all possible inheritance patterns, we encode an individual's SNP as 0 or 1 for the homozygote, 2 for the heterozygote, and 3 for missing data. Unlike inherited deletions, if individual A is a heterozygote, individual B may still harbor a *de novo* deletion.

3.8 Identification of the critical regions of recurrent deletions

Deletions in autism and other neurological disorders are often recurrent (Stefansson et al. 2008; Weiss et al. 2008), with multiple deletions occurring in the same region of distinct individuals independently. Recurrently deleted regions often present a complex deletion signature with many deletions

T	D AGRE
5	0.94
6	0.06
7	0.002

Table 3.5: We simulated 25000 trio inheritance patterns for 2500 trios using parameters from the AGRE autism data. We inferred deletions using different thresholds (T) for the DELISHUS (D) *de novo* algorithm. The statistic calculated for the false positive rate is the average amount of deletions detected in 500 iterations.

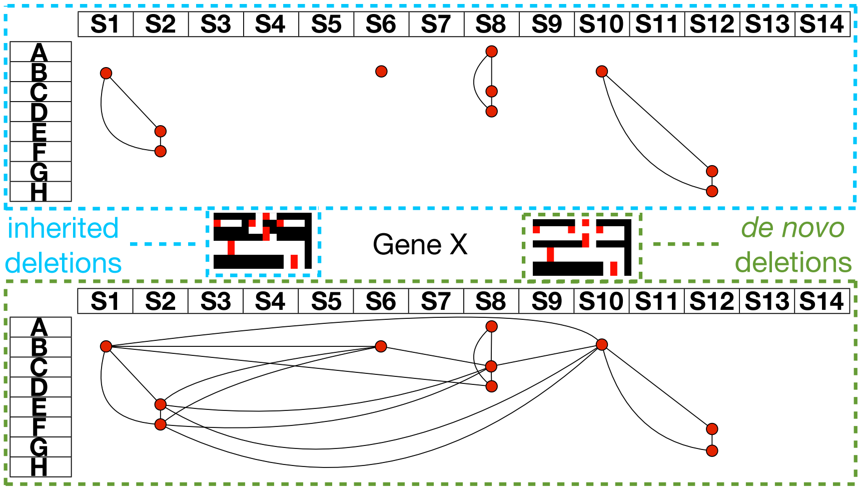


Figure 3.11: We show the graph G superimposed on M' with the trio rows denoted A-H and the SNPs denoted S1-S14 for inherited and *de novo* deletion interpretations. For inherited deletions, Gene X displays three small 3-cliques each conferring little evidence of being a true deletion. When interpreting this data for *de novo* deletions, the second trio shows evidence for one larger *de novo* deletion. In G, we see that the second trio now becomes a hub for connections to trios C through F. The outlined black, red, and white maps are deletion heat maps representing M'. Regions of 1's and 0's are represented by red and white respectively. Regions of X's and 0's are represented by black.

existing at slightly different intervals. While many configurations of deletions exist, interpretation of these regions is often formulated in a parsimonious manner. *Critical regions* capture this sense of parsimony and are defined as a region of large overlap for a subset of deletions. Critical regions are often used when attempting to connect a set of associated recurrent deletions to underlying biological mechanisms.

Because many critical regions may exist in the data, it is often useful to prioritize critical regions by generating a ranking. Formulation 2 demonstrates one method for prioritization using critical region size.

Formulation 2. *Report all recurrently deleted regions shared by at least k' deletions as significant critical regions.*

To solve this formulation, we construct a graph $G'(V', E')$ on the set of recurrent deletions. We introduce a vertex $v \in V'$ for each deletion and an edge $(v_i, v_j) \in E'$ if v_i and v_j share a SNP index. As the deletions are intervals on the chromosome we can make the following observation.

Observation 2. *$G'(V', E')$ is an interval graph and hence chordal.*

Each maximal clique now corresponds to a critical region and its size corresponds to the number of deletions participating in the critical region. Therefore, an algorithm for Formulation 2 first computes $G'(V', E')$ from the output of DELISHUS for inherited or *de novo* deletion. Because $G'(V', E')$ is chordal, all critical regions are computed using perfect elimination orderings to generate maximal clique components in guaranteed polynomial time. Critical regions with the number of deletions $\geq k'$ are then ranked according to some metric (e.g. size).

3.8.1 Validation of deletions

Deletion calls may be validated with several types of experimental and computational methods. A select subset of deletions inferred in the autism GWAS data are scheduled to undergo experimental validation in Dr. Morrow's laboratory using qPCR and custom-designed fine-tiling arrays. We validated our HapMap P1 deletion calls by comparing inferred inherited deletions to the deletions found by Conrad et al. (2006) and testing for a significant overlap. Conrad et al. (2006) developed a method that calls a region deleted if two or more evidence of deletion sites exist within close proximity to each other. From the set of computationally inferred deletion calls in the HapMap P1 data, they apply additional filtering steps and commit to 543 deletions (data extracted from the Database of Genomic Variants). From our analysis of the HapMap P1 data, we were able to produce a total of 1844 deletions covering all 543 deletions of Conrad et al. (2006).

We have shown previously that this type of analysis yields few false positives per chromosome (0.701 on average, Table 3.4). However, recurrent genomic deletions may be shared by descent or appear more frequently in specific genomic regions. In the both cases, DELISHUS uses information of the entire sample to call genomic deletions which explains, in part, the increased number of deletion calls.

Part II

Haplotype Assembly

Chapter 4

Diploid Genomes

4.1 Introduction

A considerable amount of theory and algorithms have been developed for the haplotype assembly problem (Halldórsson et al. 2004; Schwartz 2010). One approach is to restrict the input to convert an NP-hard optimization into a computationally feasible problem. For example, some authors have considered restricting the input to sequences of small read length or without mate pairs (termed gapless fragments) (Bafna et al. 2005; He et al. 2010; Lancia et al. 2001; Li et al. 2006; Rizzi et al. 2002). These models, however, are often unrealistic for current high-throughput and future third generation sequence data. Moreover, gapless fragment models are particularly problematic as paired-end sequencing is required to cover SNP alleles that are spaced at distances longer than the sequencing technology's read length. Other combinatorial and statistical algorithms have been developed for general data that relax the optimality constraint (Bansal et al. 2008; DePristo et al. 2011; He et al. 2010; Panconesi and Sozio 2004). For example, HapCut, which was used to assemble Craig Venter's diploid genome, computes maximum cuts on a graph modeled from the fragment matrix to iteratively improve their phasing solution (Bansal and Bafna 2008). Several of these methods were developed when Sanger was the abundant form of sequencing and thus it is unclear whether they can handle massive data on the scale of the 1000 genomes project and beyond. We will test this hypothesis for two leading haplotype assembly algorithms: the Genome Analysis ToolKit's read-backed phasing algorithm (DePristo et al. 2011) and HapCut (Bansal and Bafna 2008). For a survey of these approaches see Schwartz (2010).

4.1.1 Definitions

Let a fragment f be a sequence read with the non-polymorphic bases removed such that only SNPs remain. Fragments may be either a single contiguous region of DNA or contain any number of gaps between contiguous regions (for example, one gap between two contiguous regions in paired-end sequencing). Each SNP must be heterozygous and each row must cover at least two SNPs to be able to extract useful haplotype phase information from sequence reads. A SNP allele is encoded as 0 or 1 corresponding to the major or minor allele. The k^{th} base of the i^{th} fragment is referred to as $f_{i,k}$. If f_i does not include the base k in the sequence read (within the gap of a paired-read, for instance) then $f_{i,k} =$ '$-$'. Let M be the $m \times n$ SNP-fragment matrix with m rows corresponding to the m fragments and n columns corresponding to n SNPs. Two fragments f_i and f_j are in *fragment conflict* if

$$\exists k | f_{i,k} \neq f_{j,k} \wedge f_{i,k} \neq \text{'}-\text{'} \wedge f_{j,k} \neq \text{'}-\text{'} \tag{4.1}$$

Informally, fragment conflict represents two fragments that include the same SNP but differ in the allele.

The input to the haplotype assembly problem is a matrix M whose rows correspond to aligned read fragments and columns correspond to SNPs (Figure 1.2). The quality of M's construction depends on the parameters of the sequencing workflow and the accuracy of the read alignment algorithms. Misaligned read fragments can introduce erroneous base calls or sampling biases so the careful alignment of sequence reads is necessary for high quality haplotype assemblies. Without read alignment or sequencing errors, the haplotype assembly problem can be solved in time linear in the size of M by partitioning the fragments in two sets whereby no fragments internal to a set share a SNP and differ in the allele called. When errors are present, error correction may be modeled by: removing a fragment (row), removing a SNP (column), or flipping the matrix entry defined by a particular fragment and SNP (from 0 to 1 or vice versa). The goal is to convert M into a state such that the fragments (rows of M) can be distributed into two sets corresponding to the two haplotypes. All fragments in a set must agree on the allele at each SNP site and this is accomplished using the minimum number of:

1. Minimum error correction (MEC): SNP allele flips (0 to 1 or vice-versa)

2. Minimum SNP removal (MSR): SNP (columns of M) removals

3. Minimum fragment removal (MFR): fragment (rows of M) removals

4.1.2 Graph models

Two fundamental graph models associated to the SNP-fragment matrix M were introduced by Lancia et al. (2001) called the *fragment conflict graph* and the *SNP conflict graph*. The *fragment conflict graph*, $G_F(M) = (V_F, E_F)$, is defined as follows: the vertices are fragments, $f_i \in V_F$, $\forall i$ and the edges are $\{f_i, f_j\} \in E_F$ if f_i and f_j conflict $\forall i, j$. For an error-free M, each connected component in $G_F(M)$ has a bipartition and thus the vertices can be divided into two conflict-free disjoint subsets; the subsets define a haplotype phasing for the SNPs associated with the connected component (Figure 4.1).

The *SNP conflict graph*, $G_S(M) = (V_S, E_S)$, is defined as follows: the vertices are SNPs, $s_i \in V_S$, $\forall i$ and the edges $\{s_i, s_j\} \in E_S$ if s_i and s_j exhibit more than two haplotypes $\forall i, j$. If s_i and s_j exhibit three or four haplotypes, then some read covering s_i and s_j contains at least one error because only two haplotypes are possible for a diploid organism. Methods like HASH and HapCut employ different graph models where SNPs correspond to vertices and fragment information is encoded in the edges (Bansal and Bafna 2008; Bansal et al. 2008). HASH and HapCut keep a reference to the current phasing of the data and each edge is weighted proportional to the number of fragments that cover the adjacent SNPs and agree with the reference phasing.

4.1.3 Graph problem formulations

Every M induces a particular G_F, G_S, and G_C, and error correction models on these graphs yield different formulations of the haplotype assembly problem. The previously defined MEC, MSR, and MFR optimizations (along with minimum edge removal) can be defined in terms of the fundamental graph models.

1./2. Minimum edge/fragment removal (MER/MFR): Remove the minimum number of edges/vertices from the fragment conflict graph $G_F(M)$ such that the resulting graph is bipartite.

3. Minimum SNP removal (MSR): Remove the minimum number of vertices from the SNP conflict graph $G_S(M)$ such that no two vertices are adjacent.

4. Minimum error correction (MEC): Correct the minimum number of errors in fragments of M (by switching the allele from 0 to 1 or vice versa) such that the induced matrix M' is resolvable into two distinct haplotypes.

We note that in the MER formulation, although G_F may be completely resolvable, the resulting haplotypes may not be completely free of conflicts. A consensus SNP is commonly chosen at the

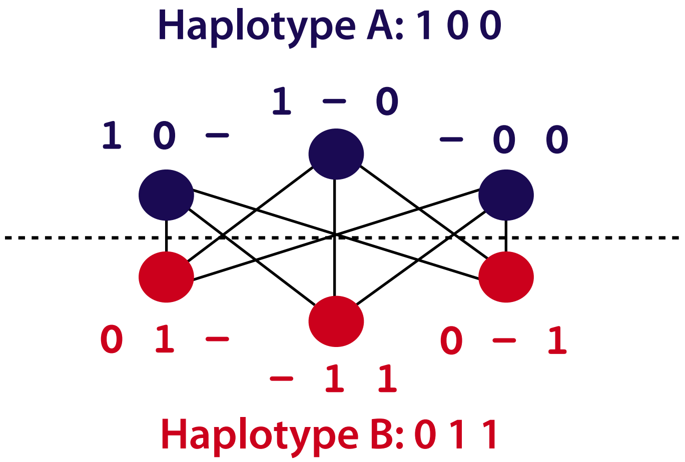


Figure 4.1: We consider three fragments sampled from both Haplotype A and Haplotype B. Fragments are represented as vertices and edges connect fragments in conflict. The dotted line represents the bipartition that separates the fragments of Haplotype A and Haplotype B.

construction of the haplotypes.

4.1.4 Prior work

Lancia et al. (2001) and Rizzi et al. (2002) provide a theoretical foundation for the MFR and MSR optimizations and describe the fundamental SNP and fragment conflict graph structures. The first widely available haplotype assembly software package was presented in Panconesi and Sozio (2004) in which the authors describe the Fast Hare algorithm which optimizes the "Min Element Removal" problem. Bansal et al. (2008) describes a Markov chain model with Metropolis updating rules to sample a set of likely haplotypes under the MEC optimization. In a follow-up, the authors presen t a much faster algorithm on a related graph model that relates maximum cuts to SNP allele flips (in the MEC model) (Bansal and Bafna 2008). Still other authors have suggested reductions to the well-known maximum satisfiability problem (He et al. 2010; Mousavi et al. 2011). The Levy et al. (2007) algorithm is a well-known heuristic that was used to haplotype assemble the HuRef genome; it assigns fragments to haplotypes in a greedy fashion and iteratively refines the solution by comparing the set of fragments to the assembled haplotypes using majority rule phasings. In a recent survey, Geraci (2010) describes the Levy et al. (2007) algorithm as, arguably, the best performing algorithm tested. The Genome Analysis ToolKit (McKenna et al. 2010) is a well-known software package which includes a haplotype assembly method that builds a Bayesian framework for the reads and attempts to infer the haplotypes with highest probability.

When the input is restricted to gapless fragments, i.e. each fragment covers a contiguous set of SNPs, MFR and MSR can be solved efficiently. However, when considering sequence reads with an arbitrary length between an arbitrary number of contiguous blocks of SNPs, MFR and MSR are NP-hard (Lancia et al. 2001). MER is NP-hard for general input (Lippert et al. 2002) and MEC is NP-hard even for gapless instances (Lippert et al. 2002; Zhao et al. 2005).

4.2 HapCompass

4.2.1 A new model: Compass graphs

Our algorithms operate on a new undirected weighted graph associated to the SNP-fragment matrix M (similar to the SNP conflict and HapCut graphs), called the *compass graph*, $G_C(M) = (V_C, E_C, w)$, defined as follows: (1) the vertices are SNPs, $s_i \in V_C$; (2) the edges are $\{s_i, s_j\} \in E_C$ if at least one fragment covers both s_i and s_j; (3) each edge $\{s_i, s_j\}$ has an associated integer weight $w(s_i, s_j)$. The weight function w is defined by the fragments. Because there exists exactly two phasings between any two heterozygous SNPs for a diploid genome, let us denote the two possible phasings as $\substack{00\\11}$ when the haplotype 00 is paired with the haplotype 11 and similarly denote $\substack{01\\10}$ the other phasing. Our weight function w for a pair of SNPs simply counts the difference between the number of $\substack{00\\11}$ phasings and the number of $\substack{01\\10}$ phasings as defined by the fragments. Formally, let F be the set of all fragments covering two SNPs s_i and s_j. The weight $w(s_i, s_j)$ is defined as follows:

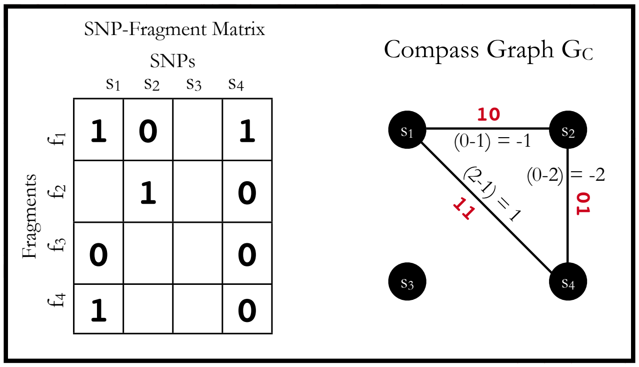
$$\sum_{f_k \in F} \Big[1\Big((f_{k,i} = 1 \wedge f_{k,j} = 1) \vee (f_{k,i} = 0 \wedge f_{k,j} = 0)\Big) \\ - 1\Big((f_{k,i} = 1 \wedge f_{k,j} = 0) \vee (f_{k,i} = 1 \wedge f_{k,j} = 0)\Big)\Big]$$

where $1(b) = 1$ for b true and $1(b) = 0$ for b false. We note that a subgraph of a compass graph is also a compass graph.

The compass graph G_C encodes information derived from the fragment set regarding the phasings of SNPs in its edge weights. For example, fragments covering three SNPs would provide phasing information for all the $\binom{3}{2}$ edges defined by the fragment in G_C. The collected evidence for an edge may have conflicting information, that is, some fragments may provide evidence for a $\substack{00\\11}$ phasing while other fragments suggest a $\substack{01\\10}$ phasing. An edge with weight of zero occurs when evidence for both phasings between the pair of SNPs is equal and thus both phasings are considered. An edge with a non-zero weight is called *decisive*. A decisive edge in E_C defines the phasing between its two

SNPs which is given by the sign of its weight i.e., majority rule phasing.

Figure 4.2 illustrates the relationship between M and its compass graph $G_C(M)$.



	s1	s2	s3	s4
f1	1	0		1
f2		1		0
f3	0			0
f4	1			0

Figure 4.2: Construction of the compass graph from SNP-fragment matrix M. The SNP-fragment matrix M (left) contains four fragments and four SNPs. Each SNP's pairwise phasing relationship defined by the fragments is represented on the edges of the compass graph (right). The majority rule phasing for one of the haplotypes is shown in red on the compass graph edges.

4.2.2 Minimum weighted edge removal

The *minimum weighted edge removal (MWER)* optimization problem is defined for a compass graph G_C. Let $L \subset E_C$ be a subset of edges in G_C and let G'_C be the resulting graph created from removing L from E_C. MWER aims to compute an L such that the following conditions are satisfied: (1) $\sum_{\{s_i,s_j\}\in L} |w(s_i,s_j)|$ is minimal (cost of removed edges is minimal); (2) all edges in G'_C are decisive; (3) choosing a phasing for each edge in G'_C by majority rule gives a unique phasing for G'_C. We call a subgraph of a compass graph that meets conditions (1-3) a *happy graph.*

The MWER problem for G_C aims at constructing the phased haplotypes that are most witnessed by pairwise phasing information contained in the fragments. Removed edges model the tolerance of some conflicting evidence. The final phasing for the retained edges is obtained as a consequence of the global unique phasing of the resulting happy graph.

4.2.3 Properties of the compass graph

We can extend unique pairwise phasings of decisive edges of G_C to unique phasings of paths. In other words, the phasing is transitive among the SNPs along a path. For example, the $(s_1, s_2), (s_2, s_4)$ path in Figure 4.2 corresponds to the concatenation of the $\genfrac{}{}{0pt}{}{01}{10}$ phasing with the $\genfrac{}{}{0pt}{}{01}{10}$ phasing, yielding the $\genfrac{}{}{0pt}{}{010}{101}$ phasing. An edge of G_C is said to be positive (negative) if its weight is positive (negative).

Lemma 2. *There is a unique phasing between two SNPs s_i and s_j if and only if for any two simple edge-disjoint paths p and q in G_C between s_i and s_j, the number of negative edges of p plus the number of negative edges of q is even, and p and q include no 0-weight edges.*

Proof. If there is a unique phasing between two SNPs s_i and s_j then they must be connected in G_C. If there is one path between s_i and s_j then the phasing is unique because this one path induces the only phasing between the two SNPs. If there is $t > 1$ paths between s_i and s_j then there exists a total of $\binom{t}{2}$ pairs of paths. Let p and q be any two paths in the traversal from s_i to s_j. We say that p and q have k and l edges with negative weight respectively. If k and l are both odd, the phasing induced between s_i and s_j by both paths is $\substack{10\\01}$. Likewise, if k and l are both even, the phasing induced between s_i and s_j by both paths is $\substack{00\\11}$. If k is odd and l is even, p defines the phasing as $\substack{10\\01}$ and q defines the phasing as $\substack{00\\11}$ (and vice versa in the case of l odd and k even). So if all paths between s_i and s_j produce a total negative edge traversal count that is even, the induced phasings cannot conflict. Likewise, if at least one pair of paths produce a total negative edge traversal count that is odd then at least one pair of paths disagree on the phasings of s_i and s_j. Also, if there is a unique phasing between two SNPs, no paths include a 0-weight edge by definition. The other direction follows similarly.

■

Definition 1. *A compass graph is **happy** if it has a unique phasing, that is, for every pair of SNPs the phasing is unique.*

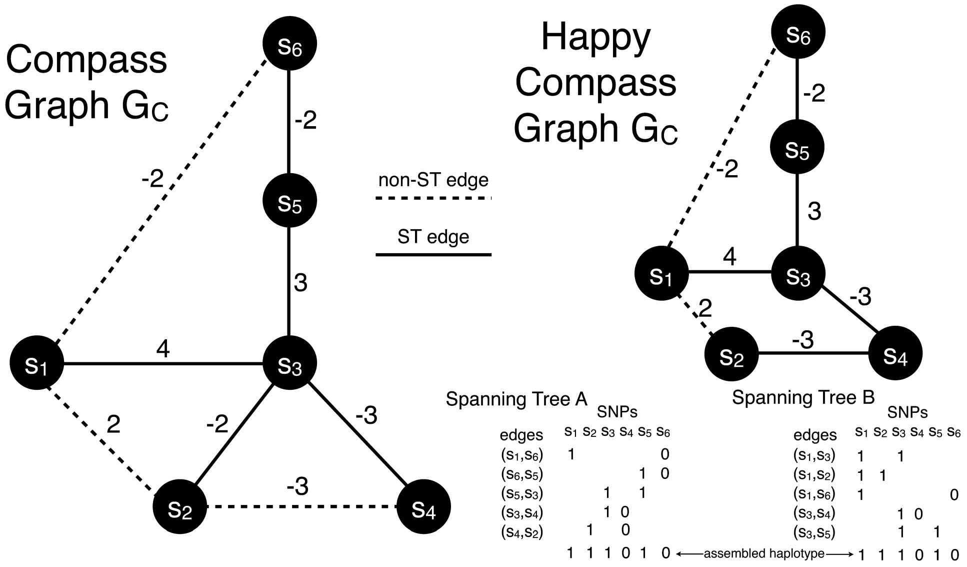
Definition 2. *A **conflicting cycle** in G_C is a simple cycle that contains an odd number of negative edges, at least one 0-weight edge or both. A non-conflicting cycle, is called a **concordant cycle** and contains an even number of negative edges and no 0-weight edges.*

Corollary 2. *A compass graph is happy iff it has no conflicting cycles.*

In general an edge may be a member of many conflicting or concordant cycles. A spanning tree of G_C is a connected, undirected subgraph that contains no cycles. There is a unique path between every two vertices in a spanning tree.

Theorem 1. *Every spanning tree of a compass graph is a happy graph. Every spanning tree of a happy compass graph has the same unique phasing as the compass graph.*

Figure 4.3 gives an example of computing a happy compass graph from G_C one edge removal step. Two spanning trees are shown in the happy G_C which correspond to the same phasing.



Spanning Tree A

edges	s1	s2	s3	s4	s5	s6
(s1,s6)	1					0
(s6,s5)					1	0
(s5,s3)			1		1	
(s3,s4)			1	0		
(s4,s2)		1		0		
	1	1	1	0	1	0

Spanning Tree B

edges	s1	s2	s3	s4	s5	s6
(s1,s3)	1		1			
(s1,s2)	1	1				
(s1,s6)	1					0
(s3,s4)			1	0		
(s3,s5)			1		1	
	1	1	1	0	1	0

Figure 4.3: A compass graph G_C is shown on the left with two conflicting cycles. One edge removal (s_2, s_3) makes G_C happy by removing two conflicting cycles in one step. All spanning trees (ST) of the happy G_C correspond to the same phasing but only two are shown in the lower right corner.

4.2.4 Cycle Basis Algorithm

We present two algorithms for the minimum weighted edge removal problem on compass graphs. Our algorithms are based on optimizations involving constructing cycle bases of connected undirected weighted subgraphs of G_C. The main idea is to consider all simple cycles in an undirected graph obtained from a cycle basis. In short, we first compute a cycle basis for G_C. An efficient algorithm for generating a cycle basis first constructs a spanning tree T of G_C and defines an arbitrary root. Then, for every non-tree edge $e \in G_C$ but $e \notin T$, we form the cycle of e plus the paths from the adjacent SNPs of e to their least common ancestor. We add the cycles created by this operation on non-tree edges to the cycle basis. This *spanning tree cycle basis* has cardinality $|E_C| - (|V_C| - 1)$.

Algorithm 1

1. Remove all 0-weight edges from G_C. *The removal of edges with 0-weight does not affect the MWER score and can therefore be removed.*

2. Construct a maximum (or near maximum) spanning tree T. *A maximum weight spanning tree basis may be preferable, but computing such a basis is NP-hard (Deo, Prabhu, and Krishnamoorthy 1982).*

3. The spanning tree cycle basis is computed in respect to T and cycles are marked as either conflicting or concordant. Iterate (4-6) until G_C is happy:

4. Select a conflicting cycle at random and remove the edge e with weight closest to 0; this represents the edge with the least amount of evidence for phasing its SNPs. *The removal of* e *can either remove a tree or non-tree edge of* T.

5. If e is a non-tree edge then T is obviously still a valid spanning tree. If e is a tree edge then we add the non-tree edge e_{nt} into the spanning tree T. *After this step we clearly still have a spanning tree as any path that previously passed through the removed edge* e *can now pass through the added edge* e_{nt}.

6. If e was a tree edge, compute a new cycle basis in respect to $T \cup e_{nt}$. *The addition of the non-tree edge into the spanning tree* T *might introduce conflicts in existing concordant cycles in which case we add these cycles to the set of conflicting cycles. However, in the worst case, the algorithm will continue to remove edges until* G_C *is a tree which is a valid phasing thus the algorithm terminates.*

7. Output the phasing corresponding to any spanning tree of G_C. Report the number of weighted edges corrected as the score of this phasing (or report the weight of all remaining edges in G_C).

Let the $|E_C| = m$, $|V_C| = n$ and the number of non-tree edges $|E_C| - |T| = m - n + 1$.

Lemma 3. *Algorithm 1 runs in* $O(m(m-n+1)^2 + (m-n+1)(m \log n))$ *time.*

Proof. The removal of 0-weight edges in step (1) can be done in $O(m)$ time. Step (2) involves computing a (near) maximum spanning tree which can be done in $O(m \log n)$ time. For step (3) we keep pointers at each vertex pointing to the “parent” node in respect to an arbitrary root vertex. The algorithm never traverse an edge more than $m - n + 1$ times. So this step takes no longer than $O(m(m-n+1))$. Again, step (4) takes no longer than $m(m-n+1)$ time for processing all simple cycles in respect to T. Step (5) processes one cycle, so, if the cycle being considered is c, then this operation takes at most $|c|$ time. Step (6) is dominated by $O(m \log n)$. For step (7) the algorithm parses through each edge of G_C thus this step takes no more than $O(m)$ time. Because we iterate through steps (4-6) at most $(m-n+1)$ times and $m(m-n+1) >> |c|$, the algorithmic complexity is $O(m(m-n+1)^2 + (m-n+1)(m \log n))$. ■

Algorithm 1 is quite simple and, in practice, we use a more complex algorithm that exploits the relationship between $MWER$ and set cover.

Algorithm 2

- We follow steps (1-3) but replace (4) with a step that removes a set of highly conflicting edges. In the $MWER$ set cover formulation each edge of G_C is a set and each conflicting simple cycle is an element. The simple cycle elements belong to the edge set if the edge is part of that cycle. We then formulate the problem of resolving the conflicting cycles as finding the set of edges (sets) of minimum weight such that they cover all of the conflicting simple cycles (elements). For an example, see Figure 4.4.

- Each conflicting simple cycle will have at least one edge removed, and, removing one or more edges from a conflicting cycle creates a tree which, due to Lemma 2, is non-conflicting. This, of course, would be too computationally expensive to formulate for the entire graph so we use this step on a subset of cycles. This subset is found by selecting the edge that is a member of the most conflicting cycles (this can easily be logged at the computation of the cycle basis).

- After removing a set of edges, we reconnect T. During the removal of each edge, we find the non-tree edge whose absolute value of the weight is the largest and add it back into T after all edges are removed.

- Step (6-7) is computed as before. Because the $MWER$ score is influenced by the order in which cycles are processed as well as the initial maximum spanning tree, steps (1-7) are iterated many times and the lowest score is reported as the solution.

Lemma 4. *At the end of each step, G_S is connected.*

Proof. If only one cycle was corrected at a time then the non-tree edge selected for inclusion into T provides a new path for vertices previously using the removed edge. If more than one cycle was corrected by the removal of one edge, then paths previously taking the removed edge can now take any non-tree edge associated with the set of cycles. ■

Lemma 4 is critically important because it ensures we do not needlessly separate components and create haplotype phase uncertainty.

The primary differences between Algorithms 1 and 2 is the local optimization step where Algorithm 2 removes multiple edges using the set cover formulation; this formulation models a sense of parsimony in that we prefer the removal of edges that resolve multiple conflicting cycles at once.

Lemma 5. *If the edge e is shared by k conflicting cycles then the removal of e resolves the k conflicting cycles.*

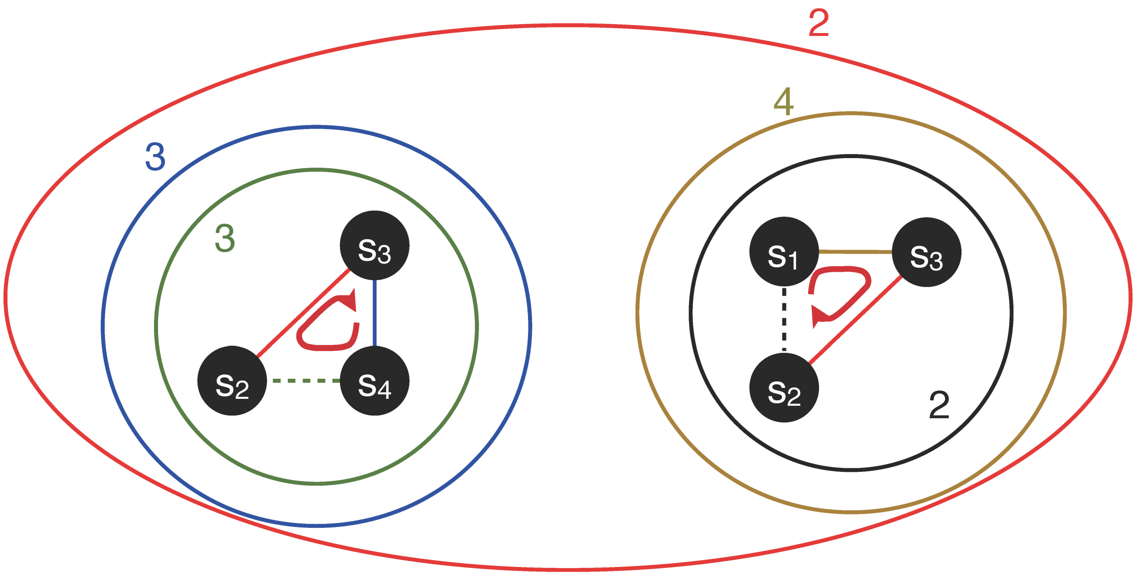


Figure 4.4: When deciding which edges to remove, HapCompass considers a set of cycles simultaneously. In this example, we consider the $\{s_1, s_3, s_2\}$ and $\{s_4, s_3, s_2\}$ cycles from Figure 4.3. Both cycles are conflicting so we must resolve them. We formulate a set cover with the cycles as the elements and the edges as the covering set. The minimum set cover is the red set, which corresponds to removing the edge (s_2, s_3). The removal of (s_2, s_3) resolves both conflicting cycles.

Proof. G_C has had all edges with 0-weight removed thus each conflicting cycle has an odd number of negative edges. Let c_i and c_j be any two of the k conflicting cycles with negative edge counts of n_i and n_j. If e is positive then c_i and c_j form a cycle whose negative edge count is $n_i + n_j$. If e is negative then c_i and c_j form a cycle whose negative edge count is $(n_i - 1) + (n_j - 1)$. In both cases (odd+odd and even+even) a cycle is produced containing an even number of negative weighted cycles. ■

An illustration of Lemma 5 is shown in Figure 4.3. There are two caveats to Lemma 5 that are due to the complex relationship between sets: (1) the removal of an edge will resolve conflicting cycles but may change concordant cycles into conflicting and (2) the removal of successive edge after the first may revert previously resolved conflicting cycles. These issues arise from the set cover formulation which simply optimizes the sum of the weighted sets and does not consider complex interactions between sets.

There are several ways to address these caveats. We may consider other properties of the edges in our minimum weighted set cover formulation. The weight on an edge e corresponds to the confidence in the pairwise phasing between the two adjacent SNPs of e. Another measure of confidence for e in G_C is the number of conflicting and concordant cycles e is a member of. The weight in the minimum weighted set cover formulation can then be computed as a combination of the edge weight and conflicting/concordant cycle membership. Because the number of conflicting or concordant cycles an edge is a member of may change with the selection of the first covering set, this minimum weighted set cover is solved iteratively. However, in practice, we specifically address (1) by breaking edge-weight ties with the number of conflicting cycles minus the number of concordant cycles and (2) by not considering shared edges from any of the resolved conflicting cycles in future removal steps of the same iteration.

Theorem 2. *Algorithm 2 is polynomial and terminates with G_C a happy graph, i.e., having exactly one phasing.*

Proof. Algorithm 2 retains Algorithm 1's complexity with additional computation in step (4). The greedy approximation algorithm for set cover, however, can be computed in linear time in the size of the sets so Algorithm 2 is clearly polynomial if it terminates. Lemma 5 allows the resolution of many conflicting cycles at each local optimization step but may also change existing concordant cycles to conflicting. However, because the graph is connected at the end of each step (Lemma 4) and we correct $|E_C| - (|V_C| - 1)$ edges in the worst case, the algorithm clearly terminates. We also have the property that the final happy graph corresponds to a valid phasing because of Lemma 4. ■

4.2.5 Generalizing the model

The generalized HapCompass model described in this work supports multiple optimizations on compass graphs, joint haplotype assembly of individuals sharing a haplotype tract IBD, and haplotype assembly of polyploid organisms. To support these algorithmic extensions, we examine key concepts of the HapCompass model and describe their generalizations.

The core of the HapCompass framework constructs the compass graph G_C, a spanning-tree cycle basis of G_C, and then corrects conflicting cycles. One such method for correcting conflicting cycles was presented in Aguiar and Istrail (2012) where edge weights are used to compute a set of edges whose removal would eliminate conflicting cycles (the MWER optimization). In principle, other methods may be used to remove edges, or, entirely new optimizations may be employed, for example, MEC. Specifically, we implement an algorithm for the MEC optimization on compass graphs. However, before an implementation of an MEC algorithm on compass graphs can be realized, the HapCompass framework must be generalized to allow for corrections to fragments.

4.2.6 Edge weights

The HapCompass framework proposed in Aguiar and Istrail (2012) defines edge weights as the difference between the number of reads indicating the $\substack{00\\11}$ and $\substack{01\\10}$ phasings. The generalized model includes a vector for edge e, v_e, consisting of four integers corresponding to the four possible haplotypes between two SNPs: 00, 01, 10, 11. A function, $f(e)$, maps the vector to a meaningful value interpreted by the HapCompass algorithm. For example, in the MWER HapCompass algorithm, $f_{MWER}(e) = v_e[0] + v_e[3] - v_e[1] - v_e[2]$ where $v_e[i]$ is the count of the phasings 00, 01, 10, 11 for $i = 0, 1, 2, 3$ respectively.

4.2.7 An MEC HapCompass optimization

The MEC optimization on G_C aims to flip the minimum number of alleles such that all of the cycles are non-conflicting. The MEC algorithm proceeds by building a spanning tree cycle basis of the compass graph. The following steps are repeated until each edge is non-conflicting. (1) For each edge e in the set of conflicting cycles: let v_1 and v_2 be the two vertices adjacent to e. (2) If $f_{MWER}(e) < 0$, we check the fragments that include both v_1 and v_2, and temporarily flip the fragment alleles of v_1 (v_2 in following iteration) to indicate $\substack{00\\11}$ phasings. The other alleles in the fragments cause edges adjacent to v_1 (v_2) to change weight as well. We record the number of conflicting cycles resolved and created by checking each cycle in the cycle basis including an edge that was modified by the flipping of a fragment allele. (3) The case of $f_{MWER}(e) > 0$ is handled analogously with the exception of flipping the alleles to indicate $\substack{10\\01}$ phasings. (4) Let the number of conflicting cycles resolved by processing e be $c_{e,r}$ and the number of conflicting cycles created be $c_{e,c}$. If $max_{\forall e}(c_{e,r} - c_{e,c}) \leq 0$, then there does not exist a favorable switching of fragment alleles and an edge is removed following the MWER algorithm. Otherwise, the fragment changes giving $max_{\forall e}(c_{e,r} - c_{e,c}) \leq 0$ are introduced in G_C. (5) When all cycles are non-conflicting, we output the phasing defined by any spanning tree.

The primary data structure change in G_C introduces a mapping of edges to fragments. The primary addition to the HapCompass framework is a definition of optimization function to remove conflicting cycles from G_C.

4.2.8 Identical-by-descent tracts and haplotype assembly

Thus far, the HapCompass framework has only been defined for a single diploid individual. The generalization of haplotype assembly to multiple genomes must be selective for which individuals to assemble jointly. For example, if two individuals do not share a haplotype by descent, one individual's set of reads does not provide any information for the other. However, when two individuals do share a haplotype by descent, the shared haplotype provides phasing information across homozygous sites as long as one individual remains heterozygous (Figure 4.5). Regions of homozygosity in an individual, which would otherwise disconnect SNPs and partition haplotype solutions, can be phased together as long as the jointly assembled genotype has heterozygous SNPs within the interval.

Multiple genotypes

The problem of joint assembly of two individuals who share a haplotype IBD (hereafter referred to as a pair) is different from jointly assembling two individuals who do not share a haplotype. In the compass graph, two unrelated genotypes have the effect that both individuals can be heterozygous but have completely different phasings. However, if they share a haplotype, a transition from a doubly heterozygous SNP to another doubly heterozygous SNP forces exactly two phasings, namely $^{00}_{11}$ or $^{01}_{10}$ (for example, SNP transitions (1,2) and (4,5) in Figure 4.5). For the doubly heterozygous to singly heterozygous transitions, we may have exactly three of the four possible 2-SNP haplotypes. In Figure 4.5, the child's genotype is 22122 and in order to phase this block using the child's data alone, we require a read to cover at least one of the first two SNPs and at least one of the last two SNPs, which may be impossible depending on the distance between the SNPs and sequence read insert length. However, if we assemble the parent with the child, we can use the shared haplotype to decode the parent's phase across SNPs 2, 3, and 4 to be $^{000}_{111}$. Because they share a haplotype, the 111 haplotype *must* be the shared haplotype and it can be inferred that the child's phased haplotypes are $^{01110}_{10101}$.

Joint haplotype assembly in HapCompass is thus encoded as follows. Each edge now has two sets of vectors corresponding to the 2-SNP haplotype transitions of the parent and child. For a doubly heterozygote to doubly heterozygote transition, the weight function can be computed as before using the coverage from both individuals (because there are exactly two disjoint phasings). For a singly heterozygote to doubly heterozygote transition (or vice-versa), the weight function can solely use the heterozygous-heterozygous transmission data from a single individual.

4.2.9 HapCompass-ILP: A combinatorial optimization approach

The minimum weighted edge removal problem aims to compute a set of edges of G_C such that the weight is minimum and there are no conflicting cycles. HapCompass is a fast algorithm optimizing

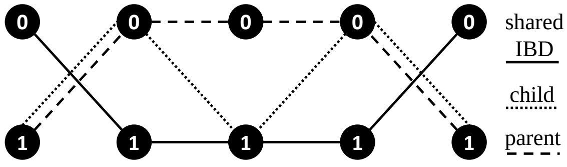


Figure 4.5: A graph of the haplotype transitions defined by the majority rule phasings of a compass graph. SNPs 1, 2, 3, 4, and 5 (left to right) are shown with both alleles (vertices) and edge transitions are encoded by a specific type of line depending on whether the haplotype is shared IBD or unique to the child or parent. The genotype of the parent and child are 22222 and 22122 respectively (where the 2 corresponds to the heterozygote and 0 and 1 correspond to homozygous for the major and minor alleles respectively).

MWER that appears to compute a solution close to the optimal. We propose to develop a mixed integer program to solve the MWER optimization optimally. Combinatorial optimization approaches for haplotype assembly are not without precedent. Lippert et al. (2002) and Wang et al. (2005) developed branch and bound algorithms for optimal inference under the MFR and MEC optimizations respectively. Guaranteed optimality will enable us to test (1) how close the HapCompass solution is to optimal and (2) how close the optimal solution is to the actual true haplotypes.

Let $x_1, ..., x_n$ be integer indicator variables on the set of all edges in G_C and $w_1, ..., w_n$ the absolute value of their edge weights. For all conflicting simple cycles $c_1, ..., c_o$ of G_C define a constraint such that the sum of all edges in c_i are less than or equal to the size of $c_i - 1$. Then the MWER ILP can be expressed as

$$\begin{aligned} \text{minimize } & \sum_{i=1}^{n} w_i x_i \\ \text{subject to } & \sum_{x \in c} x \leq |c| - 1,\, c \in C \\ & x \text{ integer} \end{aligned}$$

In general, there may be exponentially many cycles in a graph and thus an exponential number of constraints so this approach is only useful for small compass graphs.

4.3 Results

4.3.1 Theoretical

We first present results on the complexity of the MWER optimization, and related minimum weighted vertex removal (MWVR) problems on the compass graph G_C. These results motivate the usage of our heuristics for the diploid and polyploid algorithms. Let $L \subset V_C$ be a subset of vertices in G_C and let G'_C be the resulting graph created from removing L from V_C. The MWVR optimization aims to compute an L such that the following conditions are satisfied:

- (1) $\sum_{\{s_i\} \in L} |w(s_i)|$ is minimal where $w(s_i)$ is the weight of the i^{th} SNP (cost of removed vertices is minimal);
- (2) all edges in G'_C are decisive (each edge has a majority rule phasing);
- (3) choosing a phasing for each edge in G'_C by majority rule gives a unique phasing for G'_C.

We omit the straightforward proofs that the MWVR and MWER problems are in NP. It remains to be shown that known NP-hard problems can be reduced to MWVR and MWER.

We restate the conflict graph generality lemma from Lippert et al. (2002).

Lemma 6. *Let $G = (V, E)$ be an arbitrary graph. Then there exists a SNP-fragment matrix M such that $G_F(M) = G$.*

Proof. Introduce a fragment f_i for each vertex $v_i \in V$. For every two adjacent vertices $\{v_i, v_j\} \in E$, introduce a new SNP column s_k in M where $f_{i,k} = 0$ and $f_{j,k} = 1$. ■

Let M be the SNP-fragment matrix constructed from Lemma 6, G_F the corresponding fragment conflict graph of M, and G_C the compass graph of M.

Lemma 7. *Every simple cycle of odd length in G_F produces exactly one conflicting simple cycle in G_C.*

Proof. Let $\{(f_1, f_2), (f_2, f_3), ..., (f_{k-1}, f_k), (f_k, f_1)\}$ be the edges of a simple cycle c_s in G_F of length k fragments (vertices). We can partition the fragments into two sets such that each set corresponds to the haplotypes of the individual. If k is even, then we can partition the even fragments $(f_2, ..., f_k)$ and odd fragments $(f_1, ..., f_{k-1})$ into two sets such that each set does not contain internal fragment conflicts. Likewise, if k is odd, then no such partition exists because f_k conflicts with f_1 and f_{k-1}. The function that takes a cycle in G_C and computes the number of $^{01}_{10}$ (negative) edges is denoted *neg*(). We claim that for k even, $neg(c_s)$ is even and for k odd $neg(c_s)$ is odd. For this proof we consider any length $k-1$ subset of vertices in c and without loss of generality we assume this subset is $v_1, ..., v_{k-1}$. Consider any two adjacent fragments in this cycle f_i and f_j such that $i < j$ and they share the k^{th} SNP. As we iterate through fragments of the cycle, we call the allele that will be paired with the next fragment the *active* allele. If $(s_k, s_{k+1}) < 0$ then $f_{j,k+1} = f_{i,k}$, that is, the active allele that will pair with f_{j+1} is the same allele as $f_{i,k}$. However, if $(s_k, s_{k+1}) > 0$ then $f_{j,k+1} \neq f_{i,k}$, and the active allele that will pair with f_{j+1} will be the opposite allele as $f_{i,k}$. Thus negative edges in G_C do not change the active allele while positive edges in G_C flip the active allele from 0 to 1 (or vice-versa).

Case (1): k even. The $v_1, ..., v_{k-1}$ subset either has an even or odd number of negative pairwise phase relationships. Case 1.a: Even number of negative pairwise phase relationships; odd number of positive pairwise phase relationships. The active allele of v_{k-1} is the same as the active allele of v_1 therefore v_k must be induce a positive pairwise phase relationship. Case 1.b: Odd number of negative pairwise phase relationships; even number of positive pairwise phase relationships. The active allele of v_{k-1} is different from the active allele of v_1 therefore v_k must be induce a negative pairwise phase relationship. In both cases 1.a and 1.b the total number of negative edges is even.

Case(2): k is odd. Case 2.a: Even number of negative pairwise phase relationships; even number of positive pairwise phase relationships. The active allele of v_{k-1} is different from the active allele of v_1 therefore v_k must be induce a negative pairwise phase relationship. Case 2.b: Odd number of negative pairwise phase relationships; odd number of positive pairwise phase relationships. The active allele of v_{k-1} is the same as the active allele of v_1 therefore v_k must be induce a positive pairwise phase relationship. In both cases 2.a and 2.b the total number of negative edges is odd. ■

Lemma 8. *Every conflicting simple cycle in G_C includes exactly one odd length simple cycle in G_F.*

Proof. We now interpret conflicting cycles in G_C as a set of vertices of G_C which define a set of edges in G_F. ■

Because of the previous lemma, every conflicting cycle in G_C can be resolved by removing an edge of G_F which corresponds to removing a vertex in G_C.

Corollary 3. *There exists no conflicting cycles in G_C if and only if there are no cycles of odd length in G_F.*

Lemma 9. *Given an M produced from Lemma 6, the compass graph $G_C(M)$ is the line graph of $G_F(M)$ with weights of G_C as defined by the phasing relationships of the fragments of M.*

Proof. The SNPs (columns) of M contain exactly two alleles from two fragments that conflict. Therefore, in G_F, each SNP uniquely defines an edge and in G_C each SNP uniquely defines a vertex. All that remains is to show that every two adjacent edges in G_F produce an edge in G_C. Consider a SNP s whose conflicts involve fragments f_i and f_j. The edge defined by s in G_F is adjacent to edges defined by the other conflicts of f_i and f_j. The vertex s in G_C is defined exactly as the pairwise phasing relationships as defined by the SNP s and other SNP alleles in fragments f_i and f_j which in turn define the adjacencies in G_F. ∎

Because G_C is the line graph of G_F, if k simple cycles in G_C share an edge then k simple cycles in G_F share a vertex.

Theorem 3. *MWVR is NP-hard.*

Proof. The reduction is from the problem of removing the minimum number of edges of a graph to make it bipartite. Let G be an arbitrary graph and M the SNP-fragment matrix as defined in Lemma 1 which encodes the fragment conflict graph $G_F = G$. G_F may contain a number of cycles of odd length which produce conflicting cycles in the compass graph G_C by Lemma 2. Each vertex in G_C corresponds to an edge in G_F by Lemma 1. The vertex set solution to the MVR optimization L yields the minimum number of vertices required to remove all of the conflicting cycles in G_C. Because a graph is bipartite if and only if it contains no odd length cycles and G_C is the line graph of G_F, the removal of these vertices corresponds to removal of edges; the minimum of which makes G_F bipartite. ∎

Theorem 4. *MWER is NP-hard.*

Proof. The reduction is from the problem of removing the minimum number of edges of a graph to make it bipartite. Let G be an arbitrary graph and M the SNP-fragment matrix as defined in Lemma 6. We modify $G_F(M)$ by adding two additional degree 2 vertices to each edge, effectively converting each edge to a length 3 path. Cycles of odd (even) length retain their odd (even) length thus odd length cycles still create conflicting cycles in G_C. All vertices of degree k produce cliques of size k in G_C which do not correspond to any cycles in $G_F(M)$. Therefore, we label all edges of clique vertices produced from a single vertex with weight ∞. All paths of G_F will be encoded with two edges of G_C; both of which cannot be removed in an optimal solution to MWER. Given a solution to the MWER optimization, we can determine the minimum number of edges in G_F to make it bipartite. ∎

4.3.2 Experimental

The direct comparison of algorithms for which the same problem optimization is used (e.g. MEC, MFR, MSR) is straightforward. The algorithm that computes the minimum number of errors to correct is clearly the winner, for example. However, before haplotype assembly algorithms that optimize different formulations can be compared, care must be taken to develop a metric that best captures the more accurate solution.

Evaluation criteria for haplotype assembly

Before we consider new evaluation metrics that capture the quality of the haplotype assembly, we address the haplotype switch error metric that has been used previously when the ground truth is known. The haplotype switch error metric is defined as the number of switches in haplotype orientation required to reproduce the correct phasing (Lin et al. 2002). It was originally developed for the haplotype phasing problem and was among the metrics used in the Marchini et al. (2006) phasing benchmark. Switch error is generally more favorable than pure edit distances for haplotypes because it more accurately models phase relationship between adjacent SNPs.

This metric was originally developed for haplotype phasing algorithms which operate on the genotype data of many individuals simultaneously. Haplotype sharing and linkage disequilibrium are very important quantities for haplotype phasing algorithms as the relationship among adjacent SNPs allows methods to infer likely haplotypes in the data. In this manner, the switch error metric accurately captures the close range relationship between adjacent SNP phase. However, haplotype assembly algorithms operate on much different data and assumptions. Phase relationships are inferred often from long distance mate pair reads. The switch error metric does not accurately capture these relationships. Furthermore, if two haplotype assemblies do not produce the same amount of blocks of haplotypes or otherwise do not agree on where to commit to a particular phasing, then the switch error becomes biased towards those algorithms that phase less SNPs.

Instead, we suggest using a new metric inspired by genome assembly that captures how well the haplotype assembly represents the input fragments and can be applied regardless of knowing the true haplotypes. One of the most meaningful statistics for genome assembly is how many sequence reads successfully map back to the assembly. We can also slightly modify this metric to ask the question: "How many of the phase relationships represented by the read fragments are represented in the assembly?" This *fragment mapping phase relationship* (FMPR) metric summarizes how well the haplotype assembly represents the input data.

Let the set of all fragments be F and f_i the i^{th} fragment of F. We denote the k^{th} SNP of f_i as $f_{i,k}$. The haplotypes produced from an algorithm are denoted h_1 and h_2 and the allele of h_1 at position k is denoted $h_{1,k}$ ($h_{2,k}$ is defined similarly). Then the fragment mapping phase relationship metric can be described as

$$\sum_{f_i \in F} \sum_{f_{i,j}, f_{i,k} \in f_i | j \neq k} min\,(1(f_{i,j}, f_{i,k}, h_1), 1(f_{i,j}, f_{i,k}, h_2))$$

where 1() is a function that takes two SNP alleles and a haplotype and determines whether the phase relationship between the two alleles exists in the haplotype; formally, $1(f_{i,j}, f_{i,k}, h_1) = 1$ if $(f_{i,j} \neq \text{'–'} \wedge f_{i,k} \neq \text{'–'}) \wedge (f_{i,j} \neq h_{1,j} \wedge f_{i,k} \neq h_{1,k})$ and $1(f_{i,j}, f_{i,k}, h_1) = 0$ otherwise. This metric is computed by counting all of the pairwise phase relationships defined by the input set of fragments that do not exist in the solution. One fortunate side effect of this metric is that an algorithm that produces smaller blocks will be penalized. For instance, if an algorithm produces a haplotype assembly for five disjoint blocks when fragments exist in the data that connect every SNP in one large block, the switch error metric will not penalize the unknown phase between blocks. However, the fragment mapping metric will capture the phase ambiguity error that exists between disjoint blocks if the input fragments do indeed suggest they should be connected. We also define the Boolean fragment mapping (BFM) metric which counts the percentage of fragments that map to the resolved haplotypes with at least one error. The third evaluation criteria we use is the minimum error correction (MEC) measure which counts the number of allele flips in the fragments required to produce the phased haplotype assembly solution. In all previously described measures, lower values are desired. These metrics are similar to read mapping metrics in genome and transcriptome assembly, where good quality assemblies will allow for many reads to map back to them.

A pilot study

We first evaluate the number of reads that must supplement the current high coverage 1000 genomes data (The 1000 Genomes Project Consortium 2010) for the NA12878 CEU individual in order to achieve a complete haplotype assembly of chromosome 22. To do this, we supplemented the 454, Illumina, and SOLiD sequence data with simulated Illumina reads. The starting point of each simulated read was generated at random from the set of bases that were sampled by real sequence reads. Illumina-sized reads were simulated using varying distributions for insert size. Figure 4.6 shows a least squares fitted curve to the largest component (or block) sizes for various coverages in chromosome 22. Disconnected components of G_C must be phased separately and the haplotype phase between them is ambiguous; therefore, the largest component size gives an indication of the connectedness of G_C and the size of the maximum achievable phased haplotype.

We then evaluated each algorithm using the aforementioned metrics for the Illumina, SOLiD,

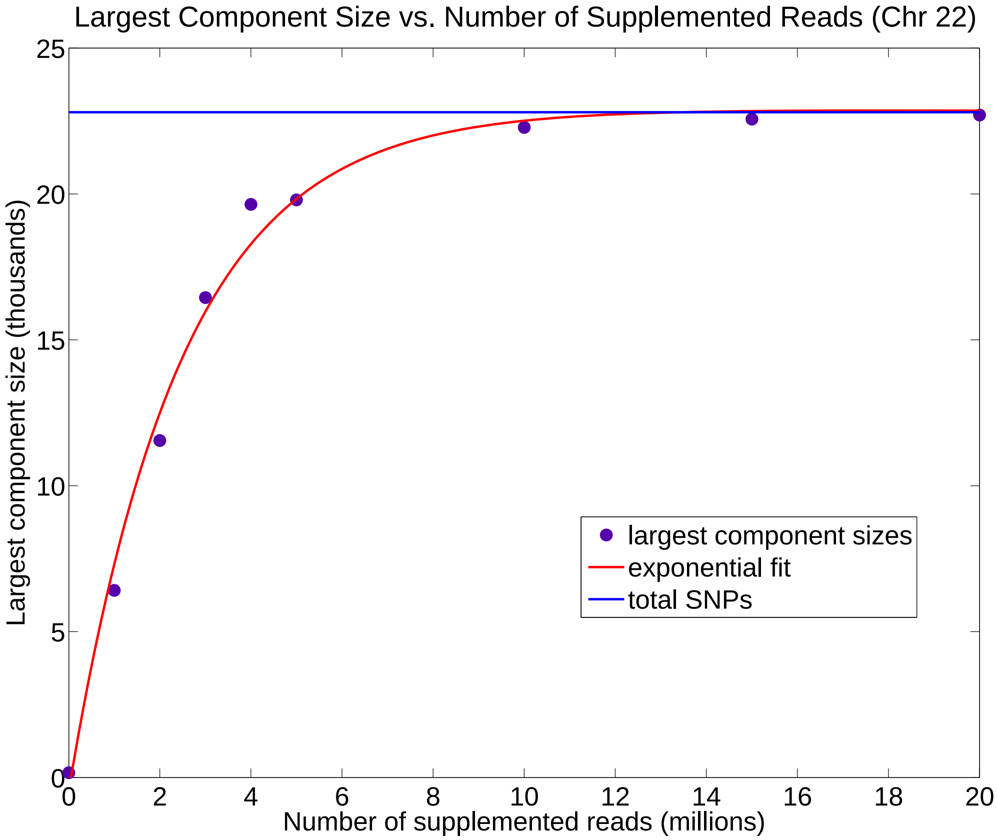


Figure 4.6: In this simulation study, reads of size 100bp were simulated on chromosome 22 of the 1000 genomes CEU individual NA12878. Mate pair lengths were sampled at random from one of four normal distribution with means [10kb, 50kb, 100kb, 250kb] and standard deviations [1kb, 5kb, 10kb, 25kb]. With these parameters for sequencing, we require about 10 million reads to connect most of the SNPs of G_C.

block size	no. fragments	HapCut FMPR (BFM %)	GATK FMPR (BFM %)	HC FMPR (BFM %)
51	477	223 (13.8)	60 (8.4)	**23 (1.9)**
53	581	265 (11)	30 (2.9)	**25 (2.4)**
53	551	71 (7.1)	23 (2.9)	**9 (1.3)**
58	626	209 (11)	**12 (1.4)**	**12 (1.4)**
60	645	199 (10.1)	54 (3.9)	**43 (3.1)**
60	467	28 (4.7)	18 (3)	**4 (0.86)**
62	393	24 (4.5)	14 (3.1)	**6 (1.5)**
62	528	126 (10.6)	16 (2.5)	**8 (1.3)**
63	770	45 (3.8)	24 (2.2)	**19 (1.7)**
66	602	91 (5.6)	31 (3.7)	**11 (1.5)**
66	718	452 (14.6)	47 (3.3)	**28 (2.1)**
79	877	245 (10.1)	26 (2.1)	**8 (0.8)**
102	949	212 (8.7)	48 (2.7)	**37 (1.9)**
166	1914	207 (5.9)	83 (2.7)	**44 (1.5)**
Total FMPR	-	2397	486	277

Table 4.1: HapCut, GATK, and HapCompass (HC) were evaluated according to the fragment mapping phase relationship and Boolean fragment mapping metrics for 1000 genomes data chromosome 22 of individual NA12878. The *block size* is the number of SNPs in the component of G_C and *no. fragments* denotes how many read fragments were used for assembly. Bold cells denote the algorithm with the best score.

and 454 reads generated for the CEU individual NA12878 in the 1000 genomes data (Table 4.1). Because each sequencing technology produces reads with similar insert sizes, the real data block sizes are small. For these block sizes, HapCompass produces the best results with GATK also producing very accurate haplotype assemblies.

Simulated data

Limitations in current sequencing technologies restrict the number of SNPs one can hope to phase from the sequence reads. Many factors influence the connectedness of G_C but the most influential factor is the mean sizes and variance of the inserts used to generate the paired reads (Halldorsson, Aguiar, and Istrail 2011). This is less of a concern for whole-exome data where haplotype assemblies can be constructed rather easily with high coverage. However, in order to test the algorithms on their capability to provide genome-wide haplotype assemblies in terms of both accuracy and time efficiency, we simulated two datasets of 10 million 100 bp reads and varied the error parameter. The 10 million reads parameter is guided by the data generated for Figure 4.6 and will vary depending on read length, insert size distributions, coverage and genome allele structure (e.g. runs of homozygosity that are longer than the insert size will disconnect components of G_C).

block size	no. fragments	HapCut FMPR (BFM %)	GATK FMPR (BFM %)	HC FMPR (BFM %)
580	2268	355 (13.8)	703 (28.2)	**284 (11.4)**
1331	4023	647 (14.5)	1236 (28.7)	**441 (10.1)**
1598	6545	1182 (15.5)	2011 (27.6)	**1033 (13.9)**
1835	6962	1212 (14.8)	2235 (28.8)	**1089 (13.8)**
3193	15036	3416 (17.7)	5237 (30)	**2746 (15.7)**
4153	17862	3642 (16.6)	- (-)	**2719 (13.2)**

Table 4.2: HapCut, GATK, and HapCompass (HC) were evaluated according to the fragment mapping phase relationship and Boolean fragment mapping metrics for 1000 genomes data chromosome 22 of individual NA12878 and 10 million simulated reads with error rate = 0.05 and read length = 100. A dash (-) mark denotes the algorithm did not finish using the allotted resources. Bold cells denote the algorithm with the best score.

Because the NA12878 individual is the child of a CEU trio who were also sequenced, we used the parents to phase most of the SNPs; a random phasing was selected for SNPs that were triply heterozygous. Using this method, we are able to construct a set of haplotypes to simulated reads from that are as close to the ground truth as possible with the available data. Our principle measurements of accuracy are FMPR and BFM. First we tested each algorithm on simulated data with moderately high error rates (0.05).

We can summarize the trends in Tables 4.1 and 4.2 by fitting a linear least squares regression line to the data (Figure 4.7).

It is clear from Figure 4.7 that HapCompass produces the best results. HapCut seems to produce better results than GATK on larger haplotype blocks (the reverse was true for the small haplotype blocks from real data). When considering serial execution, the processing times for HapCut and HapCompass were similar. For instance, for the simulated component of size 4177, 25 iterations of HapCut took 3.7 hours while 25 iterations of HapCompass took 4.8 hours. However, iterations of the HapCompass algorithm are independent and can be trivially parallelized. When this is the case, the solution with the smallest $MWER$ score is retained as the overall solution. HapCut and HapCompass both used less than 2 gigabytes of memory while GATK required a great deal more memory and processing time for similar sized components. Each algorithm was terminated if it required more than 12 hours of processing time or 8 gigabytes of heap space.

Even though switch error has an unclear interpretation on haplotype assembly data, we show that switch error produces the same algorithmic rankings. Switch error is as defined before but we incur a penalty of 1 for each haplotype block reported beyond the first. Because we compare each algorithm to a connected haplotype block – in the sense that there is a path between every SNP in G_C –

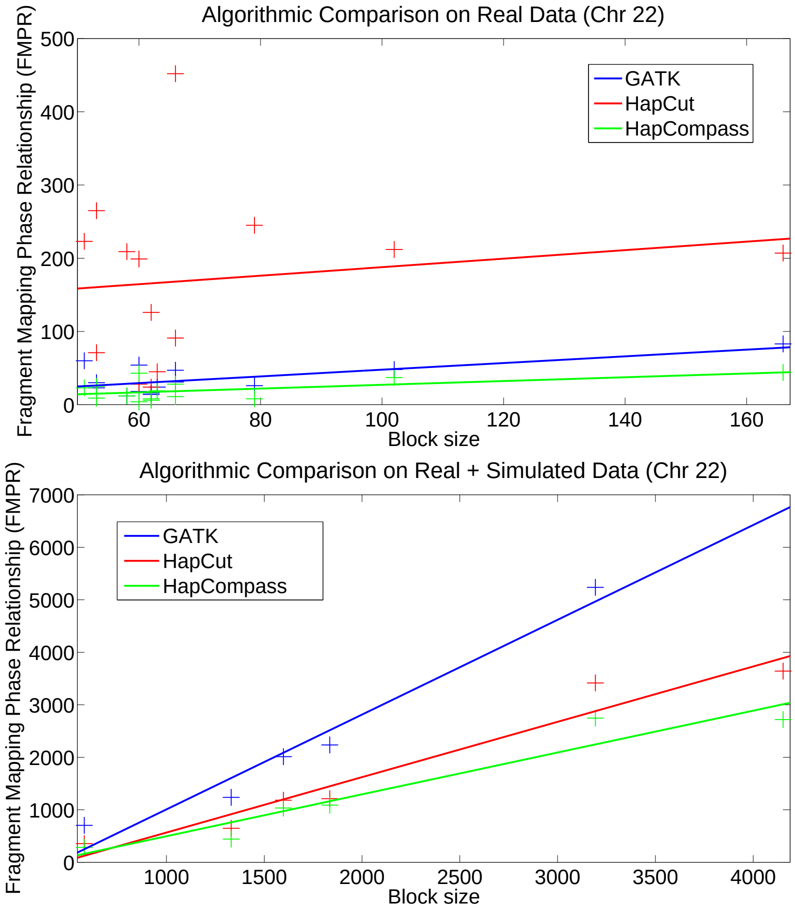


Figure 4.7: We fit a linear least squares regression line to the FMPR measurement for algorithms Genome Analysis ToolKit (GATK), HapCut, and HapCompass on chromosome 22 of the (Top) 1000 Genomes data for the NA12878 individual (Table 4.1) and (Bottom) 1000 genomes data for the NA12878 individual with 10 million simulated reads of length 100 with sequence base error rate of 0.05 (Table 4.2).

block size	no. fragments	HapCut SE	GATK SE	HC SE
580	2268	197	259	**148**
1331	4023	544	581	**329**
1598	6545	604	654	**474**
1835	6962	694	766	**563**
3193	15036	1092	1287	**859**
4153	17862	1630	-	**1007**

Table 4.3: Switch error (SE) measurements for HapCut, GATK, and HapCompass (HC) for the same data as Table 4.2. A dash (-) mark denotes the algorithm did not finish using the allotted resources. Bold cells denote the algorithm with the best score.

block size	no. fragments	HapCut FMPR (BFM %)	GATK FMPR (BFM %)	HC FMPR (BFM %)
578	2326	180 (6.6)	650 (25.5)	**46 (1.7)**
1852	7234	888 (10.7)	1896 (23.7)	**207 (2.5)**
4177	17953	2088 (9.3)	- (-)	**425 (2)**

Table 4.4: HapCut, GATK, and HapCompass (HC) were evaluated according to the fragment mapping phase relationship and Boolean fragment mapping metrics for 1000 genomes data chromosome 22 of individual NA12878 and 10 million simulated reads with error rate = 0.01 and read length = 100. A dash (-) mark denotes the algorithm did not finish using the allotted resources. Bold cells denote the algorithm with the best score.

reporting more than one phasing represents a switch error between phased components. The switch error metric gives the same relative ranking of algorithm performance (Table 4.3). We only show these results for completeness and do not recommend using switch error as the sole measurement of haplotype assembly algorithm accuracy.

We then reduced the error rate to evaluate the behavior of each algorithm on higher quality data. Table 4.4 again demonstrates that HapCompass remains significantly better than HapCut and GATK.

We also evaluate the HapCompass MEC and HapCompass IBD algorithms using 1000 Genomes Project (Siva 2008), Pacific Biosciences, and simulated data.

IBD haplotype assembly

Jointly assembling the haplotypes of related individuals has considerable benefits. The first benefit comes from the extra coverage on the shared haplotype which helps with differentiating true phasings from sequencing errors. However, the most notable advantage is being able to extend phasing past homozygous blocks. We compared the size of the phased haplotype blocks when assembling chromosome 22 of the NA12878 child in the 1000 Genomes Project data alone versus jointly with the mother. Figure 4.8 compares the maximum achievable haplotype block sizes of any single individual haplotype assembly algorithm to IBD haplotype assembly; it demonstrates that larger haplotype blocks are achievable by assembling two individuals with a shared haplotype together rather than separately.

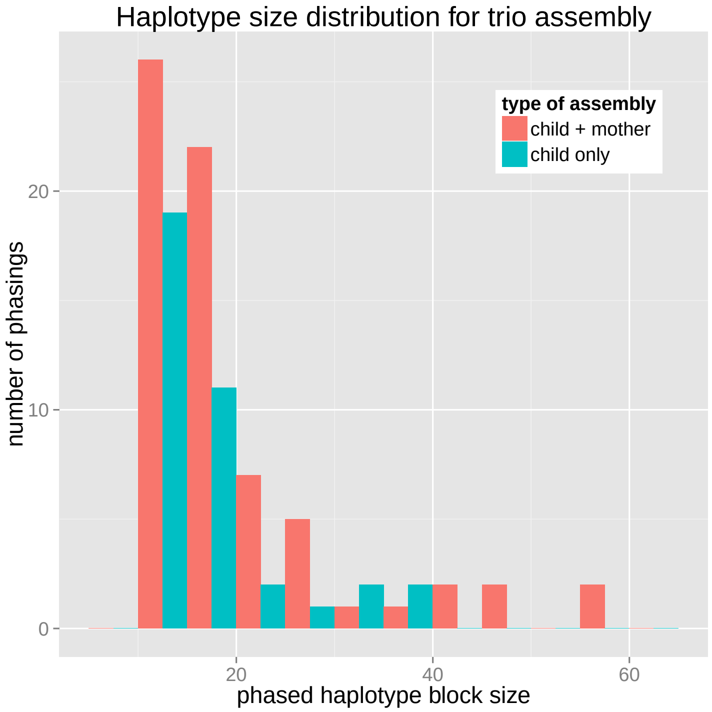


Figure 4.8: Comparison between haplotype assembling the child individually versus with a parent. The haplotype size is number of SNPs in the component of G_C which represents the maximum number of SNPs that may be phased together.

Pacific Biosciences Data

Single molecule sequencing has great potential to become a preferred method for haplotype assembly but current algorithmic techniques are untested on data with very high error rates. We downloaded the chromosome 20 data from individuals HG00321, HG00577, HG01101, NA18861, NA19313, NA19740, NA20296, and NA20800 (*Broad Institute HapMap Pacific Biosciences Data* 15 January 2013). Haplotype assembly solutions were produced by HapCompass, Levy et al. (2007), and HapCUT to obtain the results in Table 4.5 (run times can be found in Table 4.3.2). HapCompass outperforms the competition in terms of MEC using both optimizations. Interestingly, the Levy et al. (2007) algorithm is the most accurate in terms of FMPR and BFM. This is likely due to the Levy et al. (2007) algorithm processing entire read fragments each iteration while HapCompass focuses on correcting multiple fragments at adjacent SNPs. Because the Pacific Biosciences read lengths are long (several kb), more emphasis is placed on matching reads with large overlaps on the same haplotype. This result further suggests that it is important to consider the input data and the desired results when preparing data for a haplotype assembly experiment.

	HapCompass MWER	HapCompass MEC	Levy	HapCUT
FMPR	163799	169385	**153433**	169890
BFM	39827	40470	**38318**	41006
MEC	**48631**	49591	66299	50164

Table 4.5: The total FMPR, BFM, and MEC scores aggregated across individuals HG00321, HG00577, HG01101, NA18861, NA19313, NA19740, NA20296, and NA20800 in the Pacific Biosciences data.

	HapCompass MWER	HapCompass MEC	HapCUT	Levy
avg. time (s)	10	10.8	13.6	19.3
avg. memory (MB)	1251	1489	43.2	1049

Table 4.6: Average resource requirements for PacBio haplotype assembly runs. The HapCompass software is not optimized for minimal memory usage which is exemplified in the memory requirement results of the Levy et al. (2007) algorithm. This algorithm is implemented within the HapCompass software and should have a very small fingerprint but requires about a gigabyte of memory. Reducing the input fragment set into a secondary format prior to haplotype assembly (HapCUT does this) reduces our memory footprint by a factor of 10-100 times.

1000 Genomes Project Data

To further evaluate the HapCompass MEC implementation, we haplotype assembled the genome of 1000 Genomes Project NA12878 CEU child using our implementation of the Levy et al. (2007) method, HapCUT (v0.5), and the HapCompass MWER and MEC algorithms. Table 4.7 shows that the HapCompass MWER algorithm clearly performs best overall. Surprisingly, even though the MWER algorithm does not directly optimize the MEC measure, it produces the best haplotypes in respect to this measure for all but two chromosomes.

	MWER			**MEC**			**Levy**			**HapCUT**		
Chr	FMPR	BFM	MEC	FMPR	BFM	MEC	FMPR	BFM	MEC	FMPR	BFM	MEC
1	**3421**	**2348**	**2371**	3681	2519	2545	3619	2594	2632	3520	2423	2441
2	**4891**	**2930**	**2996**	5193	3081	3166	5154	3175	3273	5140	3022	3072
3	**3696**	**2394**	**2449**	4014	2585	2629	3823	2643	2703	3789	2476	2511
4	**4846**	**2710**	**2777**	5136	2906	2976	4891	2899	2974	4971	2805	2846
5	**3569**	**2245**	**2265**	3847	2428	2451	3851	2581	2606	3650	2290	2299
6	10425	**3603**	4032	10944	3846	4265	**9468**	3700	4075	10597	3630	**4030**
7	**3512**	**2138**	**2173**	3768	2288	2330	3677	2358	2407	3621	2214	2238
8	**2894**	**1864**	**1891**	3142	1999	2029	3048	2084	2118	2979	1947	1951
9	2844	**1551**	**1572**	3039	1667	1689	**2737**	1655	1687	2884	1580	1591
10	**2743**	**1857**	**1875**	2952	1981	2001	2838	2027	2048	2836	1932	1940
11	2662	**1634**	**1650**	2837	1727	1749	**2643**	1739	1778	2728	1694	1693
12	**2620**	**1627**	**1657**	2833	1784	1811	2786	1819	1856	2676	1678	1687
13	2503	**1461**	**1477**	2625	1554	1573	**2473**	1558	1576	2548	1490	1501
14	**1442**	**1020**	**1027**	1525	1070	1079	1512	1094	1102	1471	1045	1044
15	**1635**	**1085**	**1097**	1786	1168	1189	1757	1254	1272	1696	1133	1142
16	**2158**	**1308**	**1344**	2297	1410	1435	2198	1405	1458	2205	1333	1368
17	2797	1219	1320	3099	1354	1460	**2493**	1230	1305	2788	**1216**	**1299**
18	**1457**	**982**	**985**	1629	1088	1094	1563	1118	1130	1490	1013	1009
19	**1292**	**803**	**815**	1404	865	879	1369	901	918	1324	816	826
20	**1169**	**808**	**817**	1247	859	866	1279	924	939	1210	846	847
21	**871**	**545**	**558**	916	581	588	912	589	601	901	563	574
22	**681**	**446**	**449**	709	461	465	698	485	488	700	460	463
All	**64128**	**36578**	**37597**	68623	39221	40269	64789	39832	40946	65724	37606	38372

Table 4.7: Results of the NA12878 1000 Genomes Project 454 haplotype assemblies for chromosomes (chr) 1-22 and algorithms HapCompass MWER, HapCompass MEC, Levy et al. (2007), and HapCUT.

Chapter 5

Polyploid and Tumor Genomes

5.1 Introduction

5.1.1 Polyploidy

The research literature concerning polyploid haplotype assembly is essentially non-existent. The analysis of k-ploid genomes (k sets of chromosomes) has been hindered by the complexity of sequencing and assembling k chromosomes concurrently. With high-throughput sequencing technologies, genotype inference in polyploid organisms is manageable; sequence reads are mapped to a reference genome, and the relative quantities of alleles at a SNP can be inferred from sequence coverage. However, the basic assumption that there exists exactly two phasing between two SNPs no longer holds. We note that the polyploidy assembly problem is similar to a number of problems in other areas of haplotype reconstruction (when the number of haplotypes is known or unknown) such as modeling metagenomics (organism identification), HIV (viral quasispecies identification in the "metagenome" of patients), cancer (tumor and plasma), and epigenetics (regulatory region methylation reconstruction similar to "probabilistic haplotype" inference).

5.1.2 Cancer

Cancer is the world wide leading cause of death and the second leading cause of death in the United States. Despite the tremendous amount of effort and resources spent on cancer research, our knowledge of the disease pathology and treatments is limited and the outlooks for certain types of cancer are usually ominous. The commercialization of high-throughput sequencing platforms in the last decade has accelerated the growth of cancer genomics research dramatically. Since the first whole genome tumor sample was sequenced in 2008 (Ley, Mardis, et al. 2008), there have been hundreds of studies on numerous cancer types (Mardis 2012; Meyerson, Gabriel, and Getz 2010; Pleasance et al. 2010; The Cancer Genome Atlas 2012). One of the fundamental computational challenges common to many of these studies is to separate the true driver mutation signal from the biological noise (e.g. passenger mutations) and experimental noise (e.g. sequencing errors). While it is possible to map sequence reads from tumor samples to a reference genome and call genomic variants, it is exceedingly difficult to determine the parental chromosome of origin for each variant allele – that is, the variant's phase. But, *haplotypes* are important for elucidating genomic events critical to the understanding of cancer like gene fusions or driver mutations.

A theory for carcinogenesis formulated by Knudson in 1971 demonstrates the importance of haplotype phase in cancer (Knudson 1971). In the two-hit hypothesis, Knudson suggested that in order to cause cancer, at least two "hits" have to take place. The first "hit" is usually an inherited mutation, and the second "hit" is a somatic mutation in the same gene or a different gene in the same pathway occurring later in life and out of phase with the first mutation. The ability to compute the haplotype phase of the tumor genome would enable the discovery of such compound heterozygous relationships between variants and enhance our ability to identify driver mutations.

Tumor genomes have many similarities with polyploid genomes but present additional complexities that current methodologies do not model. Sequencing reads sampled from cancer patients exhibit a mixture of normal diploid cells and heavily rearranged, aneuploid cells. This introduces two major complexities into the haplotype assembly model: (1) heavily rearranged or translocated chromosomes will exhibit changes in copy number and (2) the heterogeneous nature of tumor samples requires reconstruction of more than two haplotypes each with a sample frequency which biases sequence read coverage.

5.1.3 Inferring variation

Before these complexities can be modeled, the spectrum of variation must be inferred. While early cancer research was focused on small variants such as single nucleotide variants (SNV) and indels (insertions and deletions) in a single gene or a small set of genes, advances in technology have enabled us to study large structural variants such as copy number aberrations (CNAs) and large chromosomal rearrangements in tumor genomes. Several recent studies on multiple tumor genomes have found the important role of these large structural variants in tumor development (Ding, Ellis, et al. 2010; Lee, Jiang, et al. 2010; Meyerson, Gabriel, and Getz 2010; The Cancer Genome Atlas 2012). In general, detection of cancer variation with sequencing data involves detecting those variants that are supported in the tumor genome but not found in the normal genome. The algorithms can be largely divided into three categories determined by the variant type they are trying to detect, i.e. small variants (SNVs and indels), CNAs and complex structural variants (translocations, duplications and inversions).

Strelka jointly models the normal sample as a mixture of germline variation with noise, and the tumor sample as a mixture of the normal sample with somatic mutations, in a Bayesian framework (Saunders et al. 2012). One advantage of Strelka is that it does local realignment on both tumor and normal reads together to avoid undercalls that lead to false positive results. VarScan 2 also uses the sequence reads from tumor and normal cells simultaneously, but uses a one tailed Fisher's exact test to determine whether the variants are somatic, normal, or loss of heterozygosity (LOH) (Koboldt et al. 2012). Control-FREEC not only uses the coverage information, but also the read count frequencies, to estimate copy numbers in tumor (Boeva et al. 2011). It also normalizes the tumor read depths by GC content and mappability and hence a normal genome is not required, although it could also be used for normalization.

Detection of large structural variations is often made possible by exploiting the properties of pair-end reads. For example, the insert sizes of reads that are mapped to both sides of a large deletion would appear to have much larger insert sizes than the rest of the population. CREST first looks for a cluster of soft-clipped reads that showed the evidence of a break point for a structural variant, then locates the other break point by scanning the location neighboring the paired read (Wang et al. 2011). However, the accuracy of these methods can be seriously affected when there is contamination in the samples. Contamination between individuals would lead to false positive results. Cibulskis *et al* developed a Bayesian model to estimate the level of cross-individual contamination in each sample (Cibulskis et al. 2011). There may also exist contamination within an individual: tumor tissue may be contaminated with normal DNA and vice versa. Both incorrect variant calling as well as sequence contamination represent potential sources of errors in the assembled haplotypes.

5.2 Modeling polyploid and tumor genomes

For any particular sample of tumor tissue or polyploid genome, there exists some integer k that represents the number of unique haplotypes to be assembled. In the case of an actively evolving tumor genome, this k may vary for independent samples of the same tumor. While sequence reads sampled from a tumor or polyploid genome may originate from any of the k haplotypes, we assume

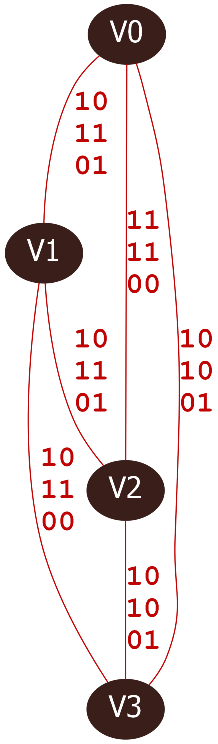


Figure 5.1: An example G_C for a tumor or polyploid sample with three unique haplotypes. Vertices are variants and edge show the haplotype phasing between pairs of variants.

sequence reads are sampled from haploid fragments. This property allows the building of phase relationships between alleles in sequence reads that contain two or more heterozygous variants (homozygous variants do not provide phase information for assembly). The input sequence reads and variants are modeled with two graph structures termed the compass graph, G_C, and chain graph, G_h.

Similar in the diploid case, let the compass graph $G_C(V_C, E_C)$ have $v \in V_C$ for each input variant and $(v_i, v_j) \in E_C$ if variants v_i and v_j are contained within a sequence read. The edges (v_i, v_j) are annotated with the weight (probability) and haplotype strings of the most likely haplotype phasing between the variants v_i and v_j (figure 5.1).

5.2.1 Uniqueness and disjoint phasings

One difficulty of polyploid and tumor haplotype assembly emerges from the non-disjointness of phasing solutions between SNPs. With the assumption that SNPs are biallelic, at least one haplotype will be shared by two or more phasings between two SNPs. In the diploid case, a read suggesting the 00 phasing could be interpreted as evidence for 11 on the other haplotype (uniqueness of phasing) and also evidence contradicting the $\substack{01\\10}$ phasing (disjointness of haplotypes in phasing solutions). In the tetraploid case, for example, if the genotype for each of 2 SNPs is $\{0, 0, 1, 1\}$ then there exists three possible haplotype phasings: $(00, 00, 11, 11)$, $(01, 01, 10, 10)$, $(00, 01, 10, 11)$.

In general, the number of haplotype phasings on an edge is a function of the ploidy of the organism and the alleles at each SNP. As in the diploid case, each SNP must have at least one of each allele or else the SNP is homozygous and sequence observations of an allele do not provide any phasing information. As a result, every 2-SNP haplotype includes either $\substack{00\\11}$ or $\substack{01\\10}$.

However, unlike in the diploid case, the extension from one edge in G_C to the next may not be deterministic. For example, in diploid assembly, if a reads suggest a $\substack{00\\11}$ phasing for SNPs 1 and 2, and a $\substack{00\\11}$ phasing for SNPs 2 and 3, the extension would give us a phasing of $\substack{000\\111}$. A conflicting cycle in G_C could then be generated if reads connecting SNPs 1 and 3 disagreed with this phasing. For the polyploid case, if the genotypes for each of SNP 1 and 2 are $(0, 0, 1)$, then both the $(00, 00, 11)$ phasing and $(00, 01, 10)$ phasing are valid. Assume that we can compute the phasings between SNPs 1 and 2 and SNPs 2 and 3 to be $(00, 00, 11)$; we can extend as we did in the diploid case to create the phasing $(000, 000, 111)$. Then, if a read suggests a 01 phasing between SNPs 1 and 3, we again generate a conflicting cycle. However, if the SNPs were phased using $(00, 01, 10)$ for SNPs 1,2 and 2,3, then either phasing $(000, 010, 101)$ or $(001, 010, 100)$ is possible. Both are completely valid phasings consistent with the genotype and read data but fragments connecting SNPs 1 and 3 *may* constrain the phasing solution to be unique.

5.2.2 Polyploid edge decidability

The polyploid and tumor HapCompass model retains the axiom that each edge is decidable; that is, each edge has a unique and computable phasing as defined by the reads. The compass graph and spanning tree cycle basis is built from the input genotypes and reads as before. The distribution of haplotype configurations between two SNPs are defined by the genotypes, and a singular configuration is computed using the available read data. The first approach attempts to assign reads into haplotype *bins* that represent the haplotype distribution for a valid phasing between two SNPs. Given a 2-SNP genotype, a *binning* is an assignment of reads to haplotypes. For example, if two SNPs both had two 0 alleles and two 1 alleles, there would exist three haplotype phasings: $(00, 00, 11, 11)$, $(01, 10, 11, 00)$, $(01, 10, 01, 10)$, each with 4 bins. The phasing $(00, 00, 11, 11)$, for instance, would contain two 00 bins and two 11 bins.

5.2.3 Binning algorithms

Greedy binning algorithm

Input: a maximum distance d between any two bins, a set of haplotype phasings P, and a set of reads R. Output: the haplotype phasing most supported by the reads.

1. For each haplotype phasing $p \in P$
2. For each haplotype bin $b \in p$, do steps (3-5).
3. Loop through steps (4-5) until all read fragments have been assigned.
4. Select a read $r \in R$ such that the edit distance between r and an available haplotype bin $h \in b$ is minimal.
5. Place r in the selected bin h and remove this read from the read set.
6. Report the haplotype phasing with the binning of minimum total edit distance as the optimal phasing.

We enforce that the difference of haplotypes in each bin must be at most d haplotypes to avoid always preferring diverse haplotype phasings (e.g. $(10, 10, 01, 01)$ vs. $(00, 11, 10, 01)$). This condition defines which haplotype bin is available during each iteration.

Probabilistic binning algorithm.

Alternatively, probabilities of each phasing given the set of reads can be computed and uncertainty can be accounted for when extending phase to adjacent edges. In particular, we wish to compute the likelihood of a phasing given the set of input sequence reads. Let p_i be the i^{th} phasing for two adjacent SNPs, P the set of all possible phasings for the two SNPs, r_j be the j^{th} read, and s_e the probability of a sequencing error. Then, the likelihood of a particular phasing p_p is

$$\begin{aligned} L(p_p|s_e, r_1, r_2, ..., r_n) &= \frac{P(r_1, r_2, ..., r_n|s_e, p_p)}{\sum_{i=1}^{|P|} P(r_1, r_2, ..., r_n|s_e, p_i)} \qquad (5.1) \\ &= \frac{P(r_1|s_e, p_p) \cdot P(r_2|s_e, p_p) \cdots P(r_n|s_e, p_p)}{\sum_{i=1}^{|P|} P(r_1, r_2, ..., r_n|s_e, p_i)} \end{aligned}$$

which may be computed using the assumption that sequence reads are independent. The probability of a read r_i given sequencing error s_e and phasing p can be computed by marginalizing over all possible haplotypes h sampled for phasing p:

$$\sum_{h \in b} p(h|s_e, p) \cdot p(r_i|s_e, h, p) \qquad (5.2)$$

Thus, the edge is decisive for the haplotype phasing with the maximum likelihood for all reads that span the two SNPs. The original diploid scoring scheme can be recreated with a manipulation of the unnormalized phasing likelihoods: $\sum_{i=1}^{n} P(r_i|s_e = 0, h =^{11}_{00}) - \sum_{i=1}^{n} P(r_i|s_e = 0, h =^{10}_{01})$.

This likelihood models haplotypes which are in equal proportion which is not necessarily the case in heterogeneous tumor samples. Thus the likelihood must be altered to accommodate the different frequencies of haplotypes we often observe in cancer samples. For example, a certain level of contamination from normal haplotypes is expected to be present in tumor sequence samples. Because the number of germline mutations is much larger than the number of somatic mutations and tumor haplotypes are derived from normal haplotypes, it is difficult to determine the origin of sequence reads by examining each read independently. We can model contamination by jointly assembling the k tumor haplotypes with two low frequency normal haplotypes. Therefore, the probability of a haplotype h with frequency f_h in the phased haplotypes of a pair of variants p can be expressed as $p(h|s_e, p) = \sum_{h \in p} f_h F(s_e, p, h)$ where F is a function that takes the sequencing error probability s_e, the set of all haplotypes for the two variant phasing p and the particular haplotype h and computes the probability of generating a read containing haplotype h.

For example, assume the three haplotypes 00, 00, and 11 exist between two variants and one of the 00 haplotypes was considered contamination at frequency 10%. If the other two haplotypes were in equal proportions, then

$$p(00|s_e, \{00, 00, 11\})F(s_e, \{00, 00, 11\}, 00) = (1 - s_e)^2 \cdot 0.1 + (1 - s_e)^2 \cdot 0.45 + (s_e)^2 \cdot 0.45 \qquad (5.3)$$

The number of unique phasings of an edge depends on the number of unique tumor haplotypes in the sample and the allele content of the variant pair. Let the number of 1 alleles for variants v_i and v_j be $1(v_i)$ and $1(v_j)$ respectively, and the number of unique tumor haplotypes be k. Then, the number of possible phasings of an edge is upper bounded by $\binom{k}{1(v_i)} \cdot \binom{k}{1(v_2)}$. This is a bound and not equality because a small number of these configurations are not allowed for heterozygous variants (for example, if $1(v_i) = k$).

5.2.4 Conflicting cycles and phase extensions

Both the greedy and probabilistic binning algorithms decide the haplotype phase of edges. In the diploid case, the extension of phasings from edges to paths was unambiguous because for each of the two phasings, exactly one haplotype begins with 0 (or 1) and exactly one haplotype ends with 0 (or 1). Therefore, the computation of phasings for paths and conflicting cycles was easily determined given the decided edges. In polyploid and tumor genomes, each SNP variant in G_C is still assumed to have only two possible alleles but each edge has three or more haplotypes. When extending phase from one edge to an adjacent edge, the haplotypes on different edges that share an allele can be used for extending phase. If this allele is present in k haplotypes, then there are $k!$ possible extensions.

5.2.5 Phase extension algorithm

We introduce the chain graph G_h which is defined on a path or cycle in G_C for a k-ploid genome or tumor sample with k haplotypes. Let $(e_1, e_2, ..., e_l) = p$ denote a path of edges in G_C of length l. Each edge e_i is phased (by the greedy or probabilistic method) and each haplotype in the phasing introduces a vertex in G_h at level i. Thus, G_h contains k vertices for each $e_i \in p$ and a total of $l \cdot k$ vertices in total. Two haplotype vertices are connected by an edge if and only if they share a SNP position and allele. Because haplotypes at adjacent levels uniquely share a SNP position in G_h, edges only exist between adjacent levels and a path through the chain graph corresponds to a joining (or extension) of haplotypes. Therefore, there is always a valid phasing for a G_h defined on a path of G_C.

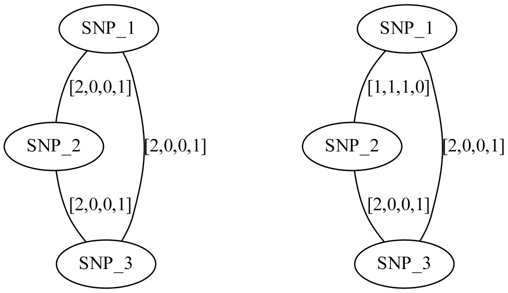


Figure 5.2: Compass graphs $G_{C,g}$, a non conflicting polyploid cycle (left), and $G_{C,c}$, a conflicting polyploid cycle (right). The vector on the edge corresponds to the haplotype counts for an edge in the format: [00,01,10,11]. In both compass graphs the haplotypes are 000, 000, and 111 while the reads in $G_{C,g}$ are 000, 000, and 111 and the reads in $G_{C,c}$ are $00-$, $01-$, $10-$, -00, -00, -11, $0-0$, $0-0$, and $1-1$

Cycles introduce complexity in G_h. G_h defined on a cycle retains the characteristics of the path chain graph, but also includes source and sink nodes: $s_1, ..., s_k$ and $t_1, ..., t_k$ respectively. Let $(e_1, e_2, ..., e_l, e_1) = p$ denote any path of edges in G_C of length l with the addition of the (e_l, e_1) edge. Source nodes are connected arbitrarily to haplotypes on level 1 but haplotypes on level l are only connected to sink nodes if the shared variant position agrees with the haplotype the source was connected to; for example, in Figure 5.3 Top, t2 is connected to both 00 haplotypes at level l because s2 is connected to a haplotype starting with 0. The sources and sinks represent the (e_l, e_1) edge and a path from s_i to t_i represents one valid haplotype. This intuition enables the formulation of the k vertex disjoint paths problem on chain graphs. If there exists k vertex disjoint, s_i to t_i paths for $i = 1, ..., k$, we have k valid phasings for the cycle; otherwise, the cycle is conflicting and there is no valid phasing.

To further build intuition, consider a conflicting cycle of G_C and G_h in the diploid case. A cycle was conflicting if the number of negative weighted edges in G_C was odd. Relating this to the chain graph G_h, an s_i node would be connected to a 0 (or 1) and each negative edge would flip the next bit. So, a conflicting cycle has an odd number of negative edges which translates into an odd number of bit flips resulting in no s_i to t_i path for $i = 1, 2$. Figure 5.2 gives an example of non-conflicting and conflicting cycles in polyploid compass graphs and Figure 5.3 their chain graphs.

G_h enables the (1) determination of conflicting cycles and (2) computation of the phased haplotypes for a path or cycle using disjoint paths. The k-disjoint paths problem is a well studied optimization in the field of discrete mathematics (Robertson and Seymour 1995). A polynomial-time solution is known to exist for the node disjoint paths problem when k is known as part of the input but these algorithms require manipulation of enormous constants rendering them difficult to

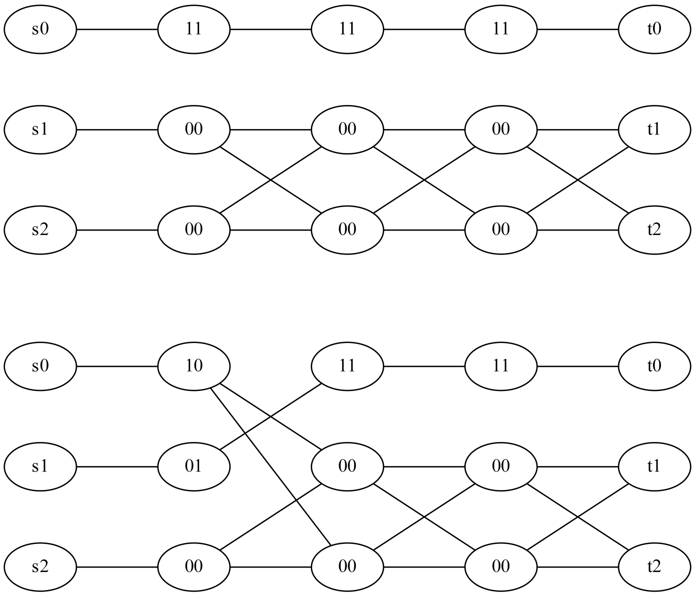


Figure 5.3: The chain graphs (top) $G_{h,g}$ and (bottom) $G_{h,c}$ corresponding to $G_{C,g}$ and $G_{C,c}$ respectively.

implement in practical settings (Kawarabayashi, Kobayashi, and Reed 2012; Robertson and Seymour 1995). Fortunately, the structure of G_h enables a much more efficient solution to the problem.

5.2.6 Disjoint $s_i t_i$ paths in the trellis graph G_h

We now present new results on the theoretical properties of this graph and extensions to phasing the entire compass graph. A *valid phasing of a path* of compass graph edges $e_{1,2}, ..., e_{s-1,s}$ is defined as k vertex-disjoint paths from level 1 to level s. A *valid phasing of a cycle* of compass graph edges $e_{1,2}, ..., e_{s,1}$ is defined as k vertex-disjoint paths from each source s_i to its corresponding sink t_i. There always exists at least one phasing for paths of G_C by definition of G_h; cycles may not exhibit a valid phasing.

Lemma 10. *There exists at least one valid phasing of k haplotypes for a cycle c if and only if there exists a valid matching between sink node annotation and chain graph nodes at each level of* G_c.

Proof. If: Adjacent edges share a variant and thus the number of x alleles at level i must equal the number of x alleles at level $i+1$ where x is any allele of the shared variant. If there is a matching at level i and $i+1$, then there must exist an edge between valid haplotype phase nodes because they share a common allele (adjacent levels). One can extend a valid haplotype phasing path from level i to $i+1$ using the edge generated by the shared allele. Only-if: Assume one level does not have a valid matching; then, either (1) at least two haplotypes share a phased haplotype node or (2) at least one phased haplotype nodes contain no sink node annotation. Case (1): multiple haplotype paths must share a phased haplotype node which breaks the vertex disjointness condition. Case (2): each level has exactly k nodes each of which must be taken once. If > 0 phased haplotype nodes contain no sink annotation, then by the pigeonhole principle at least one phased haplotype node must be shared by 2 or more haplotype paths. ■

We will use this property of G_h later in the computation of the tumor haplotype phasing.

All paths from s_i to t_i can be computed by a modified depth first search algorithm. A depth first search is started from each source s_i and the path from source to the current node is stored. Each node contains a list of integers initially empty. When the algorithm either encounters the sink node t_i, or a node already labeled with i, all nodes on the current path have i added to their list. After each source-sink pair is processed, each node contains a label i if there is an s_i to t_i path that includes the node. The runtime of this algorithm is $O(kve)$ where k is the ploidy, and v and e are the number of vertices and edges in G_h respectively.

After all nodes are labeled, we iterate through each level of G_h and create an auxiliary flow graph G_h^l where l is the level. G_h^l defines a bipartite graph where one set of vertices corresponds to the source haplotype paths which are connected to a set of vertices corresponding to the haplotypes of the phasing level l. A flow in G_h^l of total value k where each edge has capacity 1 corresponds to a maximum matching and thus a valid assignment of haplotype paths to haplotypes of the phasing at level l. This flow can be found in time linear in the size of the edge set of G_h^l. If every level of the chain graph has a valid bijection, then the cycle is non-conflicting and the path given by the matchings define a valid phasing. Figure 5.4 give an example of the auxiliary flow graphs for level 1 of the chain graphs defined in Figure 5.3.

5.2.7 Copy number aberrations and translocations in G_h

The chain graph and disjoint path framework accommodates modeling the types of variation typical of cancer genomes (Figure 5.2.7). Large copy number aberrations insert or remove large regions of genetic material. Genomic deletions can be modeled as a long edge connecting the variants flanking the deletion breakpoint. In this case, the model still expects the computation of k disjoint paths spanning the deletion. Large insertions of genetic material can be modeled as the addition of a temporary path in between or potentially overlapping vertices of G_h. The number of disjoint paths in this case changes to $k + 1$. Translocations may be modeled in G_h by a combination of a deletion

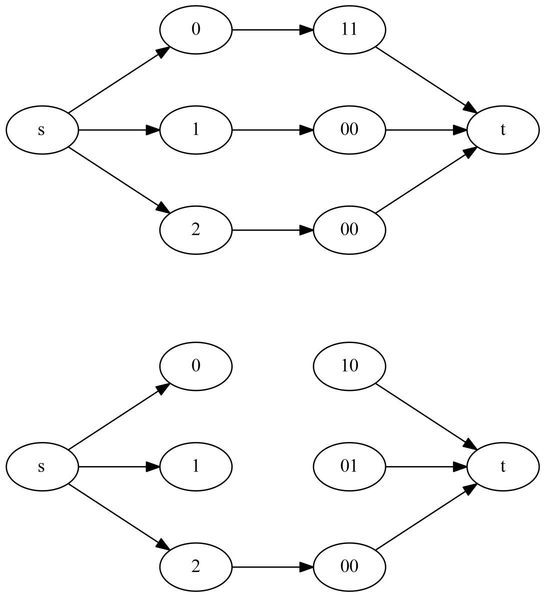


Figure 5.4: The auxiliary flow graphs (top) $G^1_{h,g}$ and (bottom) $G^1_{h,c}$. For a k-ploid organism (in this case $k = 3$, a flow of k with 1 capacity on each edge corresponds to a valid assignment of haplotype paths to haplotypes of the phasing a level 1.

and an insertion.

5.2.8 General chain graph

The general chain graph G_g is our final graph structure for representing the overall phasing of cancer genomes. Because there may be many matchings at each level of G_h, haplotype assembly of cycles in G_C will yield a set of potential phasings. Each of these partial phasings constrain the haplotype assembly to include one of the k disjoint path solutions

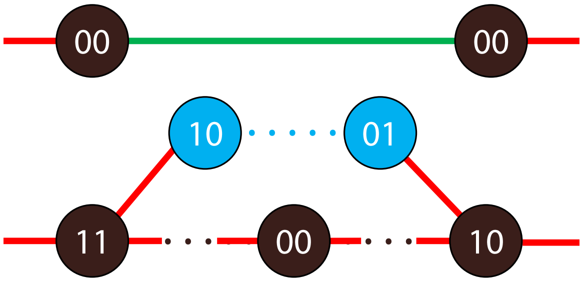


Figure 5.5: Deletions, insertions, and translocations of genomic material can be modeled using disjoint paths. The green edge models a deletion which effectively removed the deleted variants in the chain graph. The blue node insertion adds an extra path in G_h. Translocations can be modeled as an insertion and a deletion event.

G_g is a graph built on the spanning tree cycle basis of G_C. The vertices of G_g are constructed in a similar manner as G_h; each edge (v_i, v_j) of G_C generates a vertex for each haplotype in the phasing of (v_i, v_j). Every G_h constructed from a non-conflicting cycle of G_C defines a set of edge adjacencies; these adjacencies are represented in G_g. Therefore, if two edges are adjacent in a G_h, then they are also adjacent in G_g. Because of Lemma 10, we can determine the number of disjoint path solutions passing through adjacent levels i and j by simply computing the valid extensions of matchings from level i to j. Let s_{ij} be the probability of an extension between levels i and j. Then we can compute s_{ij} as the product of the probabilities of the particular phasing for the adjacent edges divided by the number of extensions. If we assume each of the l valid extensions of the sets of matchings at adjacent levels are equally likely, then the weight of a particular extension p_{ij}/l is added to the edges of G_h (and G_g).

However unlike G_h, G_g is not necessarily a trellis graph if the cycles in the basis do not agree on the ordering of edge adjacencies (Figure 5.6). If G_g were a tree, finding a phasing could be modeled as packing disjoint steiner trees or disjoint spanning trees. Instead, we model the computation of the tumor haplotype assembly as the k-maximum weight node-disjoint subgraph problem. That is, we compute a set of k node-disjoint subgraphs in G_g whose total weight is maximum over all k node-disjoint subgraphs and includes every vertex in G_g.

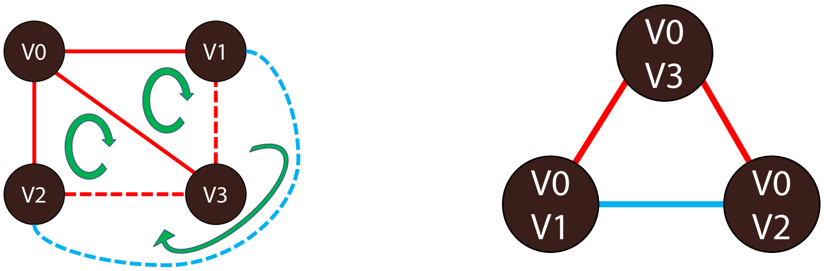


Figure 5.6: (Left) An example cancer genome compass graph G_C with three non-conflicting cycles. The dashed lines represent edges not in the spanning tree of G_C. The inclusion of each non-tree edge creates a cycle in the cycle basis of G_C. The two inner cycles $((v_0, v_1), (v_1, v_3), (v_3, v_0))$ and $((v_0, v_2), (v_2, v_3), (v_3, v_0))$ create the red-edge adjacencies in G_g (right). Computing the haplotype assembly of a tree (G_g with just the red edges) is simple. However, if the blue non-tree edge is added, the edge adjacency $((v_0, v_1), (v_0, v_2))$ must be included in G_g creating a cycle.

HapCompass polyploid and tumor algorithms

HapCompass-Poly and HapCompass-Tumor optimize the minimum weighted edge removal problem (MWER) formulation. MWER aims to compute a set of edges L of minimum weight, whose removal resolves all conflicting cycles of G_C. After all conflicting cycles have been removed, each non-conflicting cycle's chain graph is added to G_g. G_g represents the constrained solution space by containing the information each non-conflicting G_h. Because G_g may not be a tree, we select the maximum weight matching extension for each level.

Both the HapCompass-Poly and HapCompass-Tumor algorithms penalize cycles with many disjoint path solutions and encourage disjoint path solutions with strong edge phasings. HapCompass-Tumor additionally models haplotypes at different frequencies in the sample and handles variation typical of cancer genomes. The HapCompass-Poly and HapCompass-Tumor algorithms can be broken into four major steps.

HapCompass-Poly/Tumor algorithm.

1. Compute G_C, a spanning tree cycle basis, and the set of conflicting simple cycles.

2. For each conflicting simple cycle, remove the edge with the smallest likelihood.

3. After G_C is void of conflicting cycles, compute G_g and then for each non-conflicting cycle:

   (a) Compute the chain graph G_h.

   (b) Compute matchings at each level and k disjoint paths.

   (c) Add weight to adjacencies shared in G_h and G_g proportional to the likelihood of edge phasings and number of disjoint paths taken through the edge (equations 5.1 and 5.3).

4. For each edge in G_g choose the matchings for adjacent vertices that given the maximum weight extension.

We illustrate the modeling and algorithm with a series of examples. Let the compass graph G_C of a tumor sample with three unique haplotypes be shown in figure 5.1. Then, if (v_0, v_3), (v_2, v_1), and (v_3, v_2) are the non-tree edges of G_C, the chain graphs in figure 5.7 (left) are constructed.

Figure 5.7 (right) shows the G_g updated after the disjoint paths and weights of edges in G_h are computed and distributed to G_g.

5.3 Results

We implemented HapCompass-Tumor/Poly and evaluated its performance on simulated polyploid and tumor haplotypes. In these experiments we use insert size as a proxy for the computed haplotype length. It has been shown that the dominant factor in producing long haplotype assemblies is the

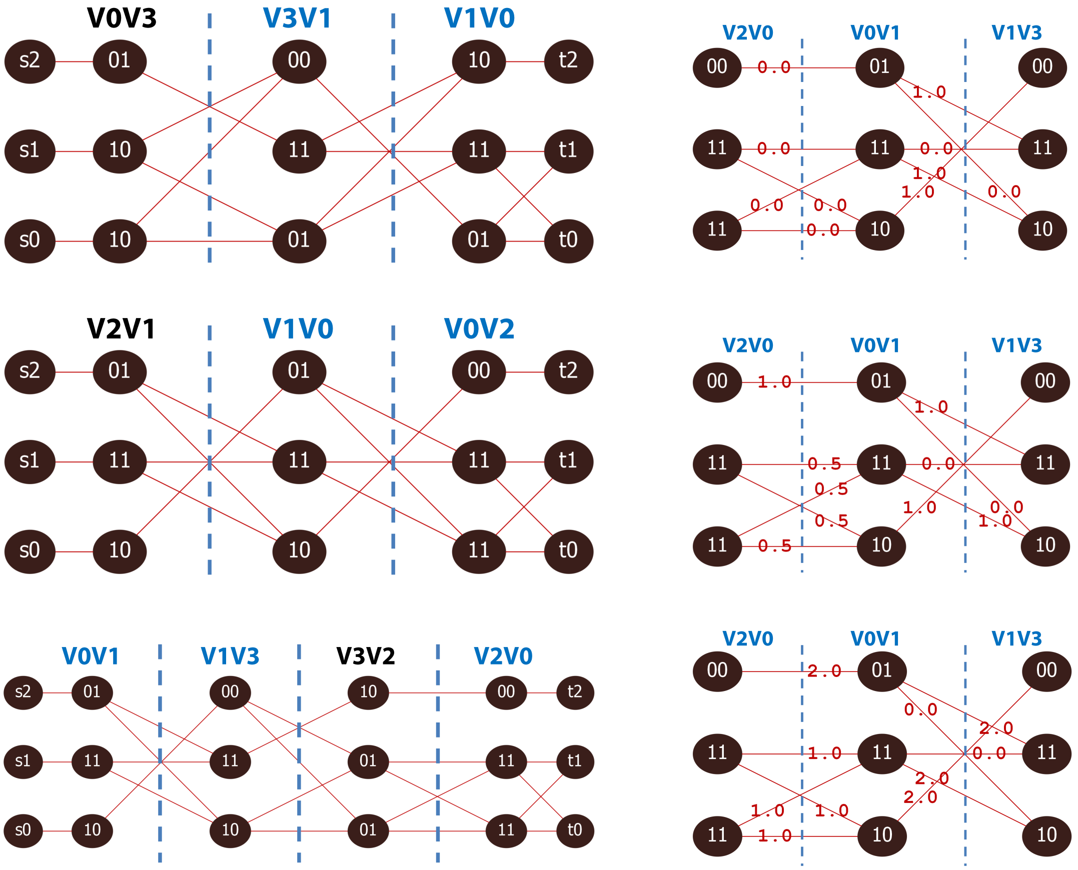


Figure 5.7: (Left) chain graphs (G_h) from the compass graph in figure 5.1. The level corresponding to edges in G_C are denoted by black (non-tree edges) and blue (spanning tree edges) lettering above the vertices. In this example, the edge phasing probabilities in G_C are all 1. So, an edge connecting level i to level j which is in b disjoint path solutions will receive a weight of b/d if there are d unique disjoint path solutions from level i to level j. The weights of edges calculated from disjoint $s_i t_i$ paths in each G_h are added to the G_g (right).

length between the read pairs (Aguiar and Istrail 2013; Halldorsson, Aguiar, and Istrail 2011). Briefly, if the length between two variants is x and the insert size is y, then a sequence read can never span the two variants if $x > y$.

5.3.1 Edge phasings

To evaluate HapCompass-Tumor/Poly, we simulated three haplotypes at random and simulated reads from these haplotypes. The simulated reads were guaranteed to contain two SNPs (assuring they are useful for haplotype assembly) and given normally distributed insert sizes. The polyploid algorithm was run using both the greedy and probabilistic binning algorithms for deciding edge phasings. Figure 5.8 demonstrates two interesting results: (1) for a small number of reads, the quality

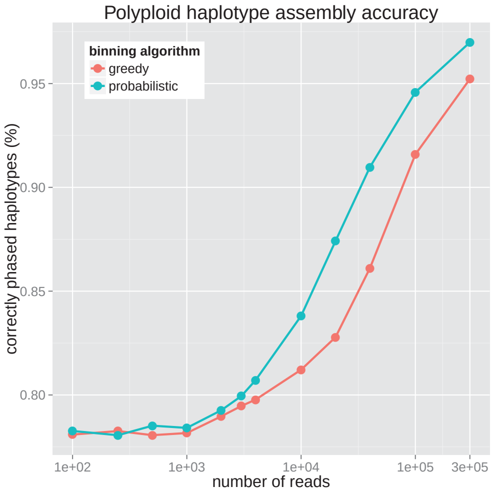


Figure 5.8: Comparison of the percentage of correctly phased polyploid SNP pairs for the greedy and probabilistic binning algorithms for varying number of input reads.

of haplotype phasing is independent of the choice of binning method and (2) that the probabilistic algorithm produces a more accurate phased solution than the greedy binning method for a large range of simulated read counts.

5.3.2 Dependence on insert size and error rates

Using the sequence for the *BRCA1* breast cancer susceptibility gene, we simulated three hyper variable tumor haplotypes. Distance between variants were distributed normally $\sim N(500, 50)$. The following procedure was repeated 250 times for each data point in Figure 5.9. Given the set of variants which remained fixed for each experiment, a random phasing is computed that is consistent with the allele distributions. We then sampled 10000 phase-informative simulated reads from the true haplotypes and computed the average edit distance between assembled and true haplotypes. We compared the distance of haplotype assemblies for the randomly generated triploid BRCA1 genes while varying sequence read insert size, standard deviation of insert size, and single base substitution error rate.

Figure 5.9 (left) demonstrates several interesting trends. First, as the insert size is increased the haplotype assemblies become more accurate. Second, the more variable the insert length, the more accurate the haplotype assembly. A hyper variable insert length appears to have a similar effect as increasing the insert size. These findings confirm patterns observed in conventional diploid haplotype assembly. Finally, while the error rate does affect haplotype assembly accuracy, as long as the error rate is less than 0.2%, the haplotype assemblies are similar in quality. This phenomenon is likely caused by the constant coverage coupled with uncertainty in phasing the edges of G_C. When the coverage is fixed and the insert sizes are short, haplotype assemblies are smaller but more accurate. Conversely, when error rates reach a threshold where edge phasings are no longer accurately called, the haplotype assembly quality suffers.

5.3.3 Cancer genome heterogeneity

We also compared the accuracy of haplotype assembly in terms of tumor genome heterogeneity (Figure 5.9 right). Sequencing parameters were fixed to produce insert sizes between 500 and 2500, short insert size standard deviations, 10000 sequence reads, and no errors. Each data point contains the average of 250 haplotype assembly edit distances. The more unique tumor haplotypes in the sample the less accurate the solution. The increasing edit distance with 5 unique haplotypes between insert sizes 2000 and 2500 is likely an effect of the rising uncertainty of edge phasings when coverage is kept fixed and more edges are being generated in G_C.

5.3.4 NA12878

We simulated paired tumor sequence reads and their mappings with Enhanced Artificial Genome Engine (EAGLE) developed by Illumina Cambridge Ltd (personal communications). The sequencing parameters were set to model paired-end Illumina data with $101bp$ read lengths and a mixture of long (length=$N(60000, 141^2)$) and short (empirical distribution from 2×101 runs, with median size $\sim 300bp$) fragment sizes. The variants simulated include SNV and indels called in NA12878 by the Genome in a Bottle Consortium (Genome in a Bottle Consortium 2013) and the HCC1187 tumor sample (Illumina Inc. 2013). Variants were combined then randomly divided into two sets for each homologous chromosome, with 30X coverage for the first chromosome and 15X coverage for the second to simulate tumor genome amplification. Sequence reads were mapped to their simulated location after single base mismatches were introduced according to empirical error rates.

We evaluated HapCompass-Tumor/Poly on all autosomes of the EAGLE simulated data and longer reads simulated using HapCompass. The reads simulated from HapCompass include medium ($200bp$) and long ($2000bp$) read lengths with error rates of 2% and 5% respectively to model the higher error rates associated with long-read high-throughput sequence technologies. We used the number of allele bit flips required to map the sequence reads to the assembled haplotypes as the evaluation metric. Table 5.1 shows the results for HapCompass-Tumor/Poly using the Kruskal-like

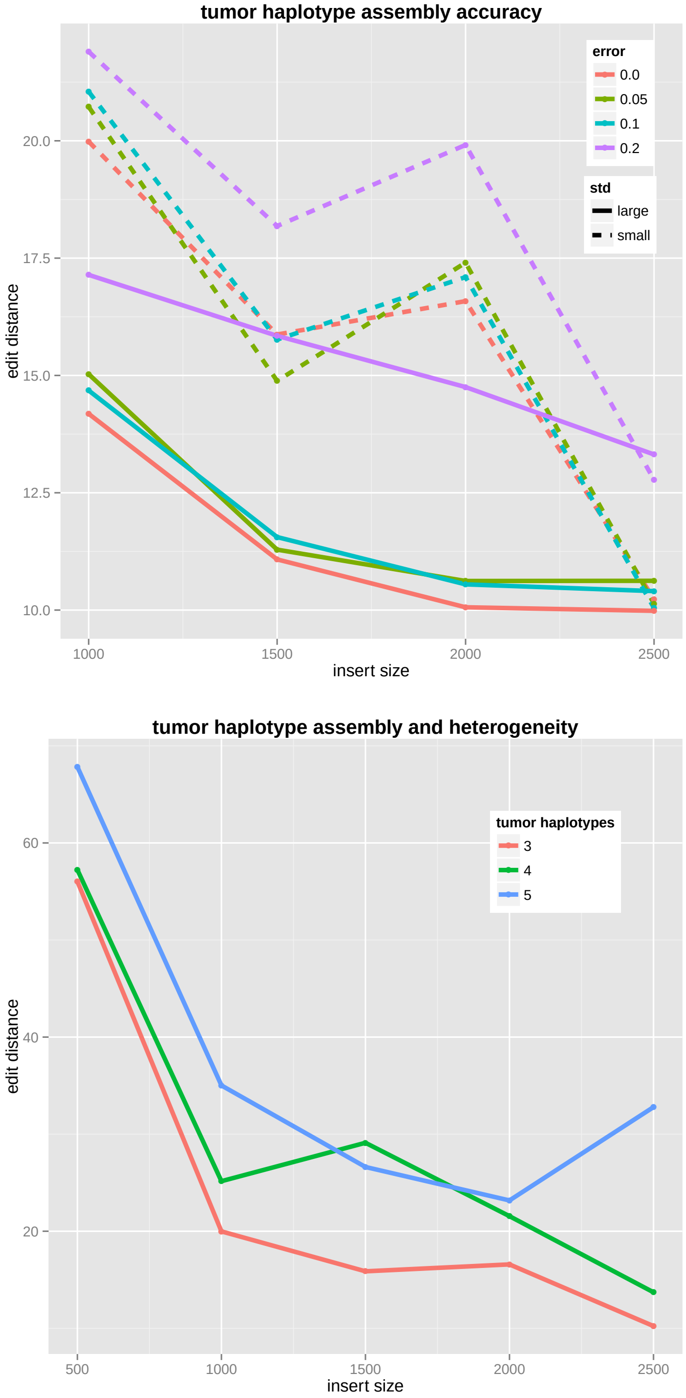


Figure 5.9: (Top) The average edit distance between haplotypes and the simulated true haplotypes is calculated with a fixed coverage and varying insert sizes, error rates (error), and standard deviations (std). (Bottom) Haplotype assembly accuracy is plotted as a function of the number of tumor haplotypes in the sample.

Table 5.1: The proportion of incorrectly mapped alleles (error) by G_g resolution algorithm. Sequence data was simulated for 1000 Genomes Project individual NA12878 using EAGLE to simulate Illumina reads and HapCompass to simulate reads with medium (200bp, 2% error rate) and long (2000bp, 5% error rate) read lengths.

G_g resolution	error (autosomes, EAGLE)	error (chr20, 200bp)	error (chr20, 2000bp)
Kruskal	0.002658	0.02079	0.04626
Kruskal Diverse	0.002659	0.02071	0.04679
Prim	0.002659	0.02789	0.05639
Prim Diverse	0.002659	0.02631	0.05867

and Prim-like algorithms for resolving G_g. Additionally, we implemented a scoring scheme that scores pairs of vertices with more diversity in haplotype sequence higher (termed *Diverse* in Table 5.1). This scheme is designed to limit uninformative pairs of vertices in the spanning tree of the compass graph G_C.

Table 5.1 demonstrates that the accuracy of the haplotype assembly depends minimally on the selection of algorithm when using Illumina-like sequencing parameters. However, as the read length increases, the Kruskal-like algorithm becomes favorable.

Part III

Identity-by-descent

Chapter 6

Identity-by-descent Algorithms

6.1 Introduction

When haplotypes are inherited from a common ancestor, they are identical-by-descent (IBD, Figure 6.1). Tracts of IBD are disrupted by recombination so the expected lengths of IDB tracts are related to the pedigree structure of the individuals involved and the number of generations till the least common ancestor at that haplotype region. Co-inherited haplotypes can be used to phase genotypes or map regions of the genome associated with a particular phenotype of interest.

6.1.1 Li-Stephens PAC-Likelihood Model and the $O(m^2n)$ time bound

Understanding and interpreting patters of linkage disequilibrium (LD) among multiple variants in a genome-wide population sample is a major technical challenge in population genomics. A large body of research literature is devoted to the topic including the computational framework presented in the seminal work of Li and Stephens (Li and Stephens 2003). Building on the work by Stephens, Smith, and Donnelly (2001), Hudson (1991), and Fearnhead and Donnelly (2001), the Li-Stephens framework led the way towards major advances in the understanding and modeling of linkage disequilibrium patterns and recombination.

The difficulties associated with modeling LD patterns at multiple loci include a number of long standing analytical obstacles. Among existing bottlenecks is the notorious (1) *curse of the pairwise*, as all the popular LD measures in the literature are pairwise measures, and the (2) *haplotype block-free* approach to avoid *ad hoc* haplotype block definitions and "fake blocks" due to recombination rate heterogeneity. Current methods for computing haplotype blocks result in the definition of *ad*

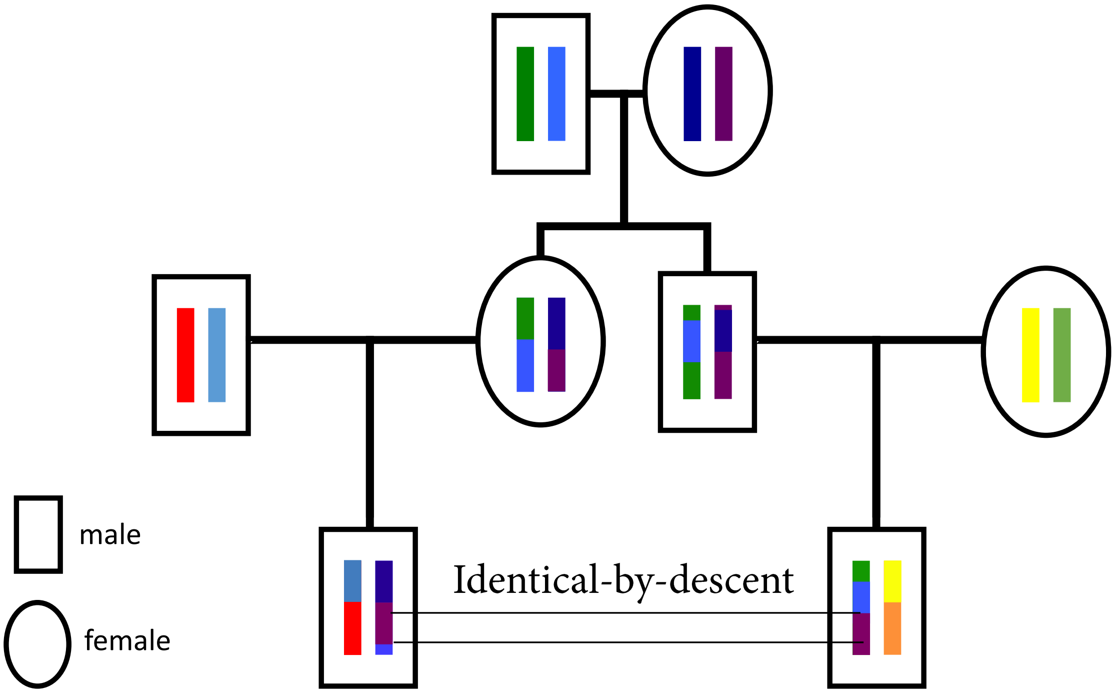


Figure 6.1: A pedigree is shown with unique haplotype tracts represented by colored lines. Due to recombination during meiosis, children inherit a mosaic of their parent's haplotypes. A single haplotype segment is shown passed identical-by-descent from the grandmother to the grandchildren.

hoc boundaries that sometimes present less LD within blocks than between blocks due to different patterns of recombination. This phenomenon leads to spurious block-like clusters. The Li-Stephens statistical model for LD, named the *Product of Approximate Conditionals* (PAC), is based on a generalization of coalescent theory to include recombination (Hudson 1991; Kingman 1982).

The optimization problem introduces the PAC likelihood $L_{PAC}(\rho)$

$$L_{PAC}(\rho) = \widehat{\pi}(h_1 \mid \rho)\widehat{\pi}(h_2 \mid h_1, \rho)...\widehat{\pi}(h_m \mid h_1, ..., h_{m-1}, \rho)$$

where $h_1, ..., h_m$ are the m sampled haplotypes, ρ denotes the recombination parameter, and $\widehat{\pi}$ represents an approximation of the corresponding conditional probabilities. Li and Stephens propose a number of such approximations for approximate likelihood functions (Li and Stephens 2003). $L_{PAC}(\rho)$ represents the unknown distribution

$$Prob(h_1, ..., h_m \mid \rho) = Prob(h_1 \mid \rho)Prob(h_2 \mid h_1, \rho)...Prob(h_m \mid h_1, ..., h_{m-1}, \rho)$$

The choice of $\widehat{\pi}$ gives the form of the likelihood objective function.

The PAC likelihood is based on expanding the modeling to capture realistic genomic structure while generalizing Ewens' sampling formula and coalescent theory. The framework iteratively samples the m haplotypes; if the first k haplotypes have been sampled $h_1, ..., h_k$, then the conditional distribution for the next sampled haplotype is $Prob(h_{k+1} \mid h_1, ..., h_k)$. $\widehat{\pi}$ approximates this distribution and is constructed to satisfy the following axioms:

1. h_{k+1} is more likely to match a haplotype from $h_1, ..., h_k$ that has been observed many times rather than a haplotype that has been observed less frequently.

2. The probability of observing a novel haplotype decreases as k increases.

3. The probability of observing a novel haplotype increases as $\theta = 4N\mu$ increases, where N is the population size and μ is the mutation rate.

4. If the next haplotype is not identical to a previously observed haplotype, it will tend to differ by a small number of mutations from an existing haplotype (as in the Ewens' sampling formula model).

5. Due to recombination, h_{k+1} will resemble haplotypes $h_1, ..., h_k$ over contiguous genomic regions; the average physical length of these regions should be larger in genomic regions where the local rate of recombination is low.

Intuitively, the next haplotype h_{k+1} should be an imperfect *mosaic* of the first k haplotypes, with the size of the mosaic fragments being smaller for higher values of the recombination rate. Although the proposed model ($\widehat{\pi_A}$ in the notation of Li and Stephens (2003)) satisfies the above axioms and has the desirable property of being efficiently computable, it has a serious disadvantage. As is stated in their article, this "unwelcome" feature of the PAC likelihoods corresponding to the choices for $\widehat{\pi}$ is *order dependence*, that is, the choices are dependent on the order of the haplotypes sampled. Other methods used in the literature, notable, Stephens, Smith, and Donnelly (2001) and Fearnhead and Donnelly (2001), present the same problem of order dependence. Different haplotype sampling permutations correspond to different distributions; these probability distributions *do not satisfy the property of exchangeability* that we would expect to be satisfied by the true but unknown distribution.

6.1.2 Identical-by-descent haplotype tracts

Haplotype tracts, or contiguous segments of haplotypes, are identical-by-descent (IBD) if they are inherited from a common ancestor (Browning and Browning 2012). Tracts of haplotypes shared

IBD are disrupted by recombination so the expected lengths of the IDB tracts depends on the pedigree structure of the sample and the number of generations till the least common ancestor at that haplotype region. The computation of IBD is fundamental to genetic mapping and can be inferred using the PAC likelihood model.

To model the effects of recombination, a hidden Markov model (HMM) is defined to achieve a mosaic construction. At every variant, it is possible to transition to any of the haplotypes generated so far with a given probability. Thus, a path through the chain starts with a segment from one haplotype and continues with a segment from another haplotype and so on. To enforce the mosaic segments to resemble haplotype tracts, the probability of continuing in the same haplotype without jumping is defined exponentially in terms of the physical distance (assumed known) between the markers; that is, if sites j and $j+1$ are at a small genetic distance apart, then they are highly likely to exist on the same haplotype. The computation of the L_{PAC} is linear in the number of variants (n) and quadratic in the number of haplotypes (m) in the sample, hence the $O(m^2n)$ time bound.

In this work we present results that remove the pairwise quadratic dependence by computing multi-shared haplotype tracts. Multi-shared haplotype tracts are maximally shared contiguous segments of haplotypes starting and ending at the same genomic position that cannot be extended by adding more haplotypes in the sample. Because we represent the pairwise sharing in sets of haplotypes, no more than $O(mn)$ multi-shared haplotype tracts may exist.

6.1.3 Prior work

Building on the PAC model, the IMPUTE2 (Howie, Donnelly, and Marchini 2009) and MaCH (Li et al. 2010) algorithms employ HMMs to model a sample set of haplotypes as an imperfect mosaic of reference haplotypes. The usage of the forward-backward HMM algorithm brings these methods in the same $O(m^2n)$ time bound class. The phasing program SHAPEIT (segmented haplotype estimation and imputation tool) also builds on the PAC model by decomposing the haplotype matrix uniformly into a number of segments and creating linear time mosaics within each such *ad hoc* segmented structure (Delaneau, Marchini, and Zagury 2011). The dependence on the number of segments is not considered in the time complexity.

PLINK (Purcell et al. 2007), FastIBD (Browning and Browning 2011a), DASH (Gusev et al. 2011), and IBD-Groupon (He 2013) are algorithms based on HMMs or graph theory clustering methods that consider pairs of haplotypes to compute IBD tracts. Iterating over all such pairs takes time $O(m^2n)$ and is intractable for large samples; this intractability is best described in the recent work of Gusev *et al.* 2011.

> “Although the HMM schemes offer high resolution of detection [of IBD], the implementations require examining all pairs of samples and are intractable for GWAS-sized cohorts. ... In aggregate, these identical-by-descent segments can represent the totality of detectable recent haplotype sharing and could thus serve as refined proxies for recent variants that are generally rare and difficult to detect otherwise.” Gusev et al. (2011)

Gusev et al. (2009) describes the computationally efficient algorithm GERMLINE which employs a dictionary hashing approach. The input haplotype matrix is divided into discrete slices or windows and haplotype words that hash to the same value are identified as shared. Due to this dependence on windows, the algorithm is inherently inexact. While the identification of small haplotype tracts within error-free windows can be performed in linear time, GERMLINE’s method for handling base call errors is worst case quadratic. However, GERMLINE’s runtime has been shown to be near linear time in practice (Browning and Browning 2012).

In what follows, we describe the Tractatus algorithm for computing IBD multi-shared haplotype tracts from a sample of haplotypes and the Tractatus-HH algorithm for computing **h**omozygous **h**aplotypes in a sample of genotypes. First, we present the computational model and algorithms. We then compare the runtime of Tractatus to a generic pairwise algorithm, compares false positive rates and power with GERMLINE, and provides an example computation of homozygous haplotype regions in genome-wide association study data of autism.

Our work presented here addresses the lack of exchangeability in the sampling methods of the Li-Stephens model and provides a rigorous result that gives a basis for sampling with the assured exchangeability property. We also present a data structure that speeds up the HMM and the graph clustering models for the detection of identical-by-descent haplotype tracts. Informally, a *haplotype tract* or simply *tract* is a contiguous segment of a haplotype – defined by start and end variant indices – that is shared (identical) by two or more haplotypes in a given sample of haplotypes. One can then view each of the haplotypes in the set as a mosaic concatenation of tracts. Such a haplotype tract decomposition is unique and a global property of the sample. Our Tractatus algorithm computes the *Tract tree* of all the tracts of the haplotype sample in linear time in the size of the sample. The Tract tree, related to a suffix tree, represents each haplotype tract in a single root-to-internal-node path. Repeated substrings in distinct haplotypes are compressed and represented only once in the Tract tree.

6.2 Tractatus

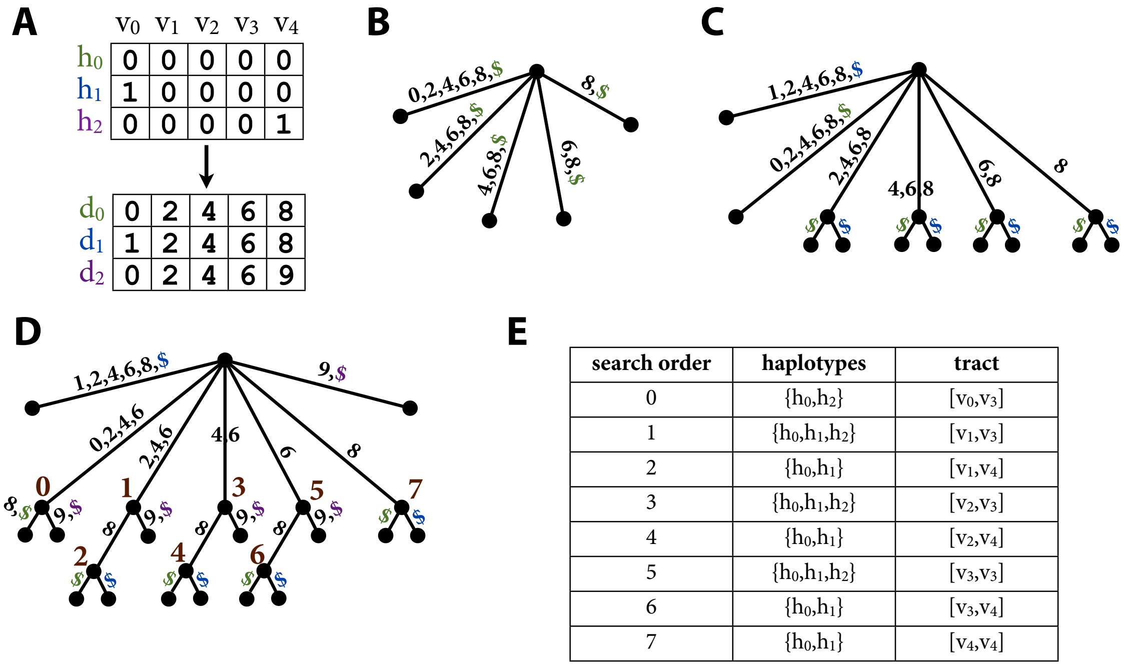
Suffix trees are graph theoretic data structures for compressing the suffixes of a character string. Several algorithms exist for suffix tree construction including the notable McCreight and Ukkonen algorithms that achieve linear time and space constructions for $O(1)$ alphabets (McCreight 1976; Ukkonen 1995). Farach (1997) introduced the first suffix-tree construction algorithm that is linear time and space for integer alphabets. Extensions to suffix-trees, commonly known as generalized suffix trees, allow for suffix-tree construction of multiple strings.

The input to the problem of IBD tract inference is m haplotypes which are encoded as n-length strings of 0's and 1's corresponding to the major and minor alleles of genomic variants $v_1, ..., v_n$. Because we are interested in IBD relationships which are by definition interhaplotype, naive application of suffix-tree construction algorithms to the set of haplotypes would poorly model IBD by including intrahaplotype relationships. Let haplotype i be denoted h_i and the allele of h_i at position j be denoted $h_{i,j}$. Then, we model each haplotype $h_i = h_{i,1}, h_{i,2}, ..., h_{i,n}$ with a new string $d_i = (h_{i,1}, 1), (h_{i,2}, 2), ..., (h_{i,n}, n)$ for $1 \leq i \leq m$. Computationally, the position-allele pairs can be modeled as integers $\in [0, 1, 2, ..., 2n-1]$ where $(h_{i,j}, j)$ is $2 * j + h_{i,j}$ where $h_{i,j} \in 0, 1$. The transformed haplotype strings are termed *tractized*.

6.2.1 The Tractatus algorithm without errors

The Tractatus algorithm incorporates elements from integer alphabet suffix trees with auxiliary data structures and algorithms for computing IBD haplotype tracts. Firstly, a suffix tree is built from the set of m tractized haplotypes each of length n. To represent the tractized haplotypes, the alphabet size is $O(n)$, so Farach's algorithm may be used to construct a suffix tree in linear time (Farach 1997). The suffix tree built from the tractized haplotypes is termed the *Tract tree*. After the Tract tree is built, an $O(mn)$ depth first post-order search (DFS) is computed to label each vertex with the number of haplotype descendants. These pointers enable the computation of groups of individuals sharing a tract in linear time.

Substrings of haplotypes are compressed if they are identical and contain the same start and end positions in two or more haplotypes. We consider a path from the root to a node with k descendants as maximal if it is not contained within any other path in the Tract tree. The maximal paths can be computed using a depth first search of the Tract tree, starting with suffixes beginning at 0 and ending at suffixes beginning with $2n-1$. Of course, if a tract is shared by $k \geq 2$ haplotypes, it is represented only once in the Tract tree. Figure 6.2 shows the construction of the Tract tree and



search order	haplotypes	tract
0	{h₀,h₂}	[v₀,v₃]
1	{h₀,h₁,h₂}	[v₁,v₃]
2	{h₀,h₁}	[v₁,v₄]
3	{h₀,h₁,h₂}	[v₂,v₃]
4	{h₀,h₁}	[v₂,v₄]
5	{h₀,h₁,h₂}	[v₃,v₃]
6	{h₀,h₁}	[v₃,v₄]
7	{h₀,h₁}	[v₄,v₄]

Figure 6.2: Construction of the Tract tree and running Tractatus on example input without errors or allele mismatches. Terminator characters \$ are colored to match tractized haplotypes and the empty string (simply the terminator character) is omitted in this example. (A) The haplotype matrix is encoded by an integer alphabet representing position-allele pairs. (B) Tractized haplotype d_0 is inserted in the Tract tree. The first tractized haplotype inserts $O(n)$ nodes into the Tract tree. (C) Tractized haplotype d_1 is inserted in the Tract tree. The suffix of d_1 starting at v_0 requires generation of a new node in the Tract tree but subsequent suffixes can be compressed along paths from the root. (D) Tractized haplotype d_2 is inserted in the Tract tree and the algorithmic search order is given in brown integers adjacent to internal nodes. Leaf nodes have exactly one terminating character (haplotype) and therefore do not have to be visited during the search. (E) The largest IBD tracts are found at search numbers 0, 1, and 2. Saving references to these tracts enables the determination that subsequent tracts are contained within already processed tracts.

computation of IBD tracts.

The internal nodes of the Tract tree also have an interpretation in regards to Fisher junctions. A Fisher junction is a position in DNA between two variants such that the DNA segments that meet in this virtual point were ancestrally on different chromosomes. Fisher junctions are represented in the Tract tree where maximal tracts branch.

After maximal tracts are computed, they are quantified as IBD or IBS. Tractatus implements two methods for calling maximally shared tracts IBD or IBS. A simple tract calling method thresholds the length L (number of variants) or area (variants × haplotypes) of the tract in terms of the haplotype matrix input. A more complex method considers the probability of a region being shared IBD or identical-by-state (IBS). If two individuals are k^{th} degree cousins, the probability they share a haplotype tract IBD is 2^{-2k} due to the number of meioses between them (Kong et al. 2008). Let the frequency of a variant at position i be f_i. Then, the probability of IBD and IBS can be combined to define the probability that a shared haplotype tract of length L and starting at position s for k^{th} degree cousins is IBD, Equation 6.1

$$P(IBD|L) = \frac{2^{-2k}}{2^{-2k} + \prod_{i=s}^{s+L}(f_i^2 + (1-f_i)^2)} \tag{6.1}$$

The value of k can be approximated if the population structure is known. Tractatus without errors is presented in Algorithm 1. Because the suffix tree is computable in $O(mn)$ time with $O(mn)$ nodes, the tree traversals can be computed in $O(mn)$ time thus giving Theorem 5.

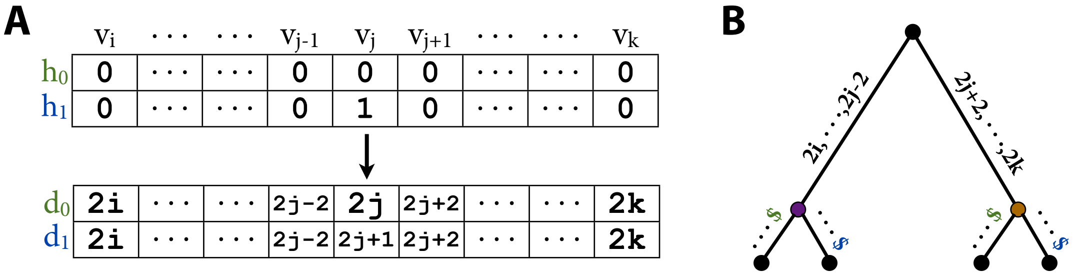
Theorem 5. *Given a set of m haplotypes each of length n, Algorithm 1 computes the Tract tree and the set of IBD tracts in $O(mn)$ time and space.*

```
input  : $m$ haplotypes each of length $n$, minimum length $L$ or IBD probability $p$
output: set of IBD tracts
$H \leftarrow$ tractized haplotypes
$T(H) \leftarrow$ Tract tree of $H$
Post-order DFS of T(H) to compute descendant haplotypes from each node
DFS of T(H):
if path in DFS is longer than $L$ or $P(IBD) > p$ and node has at least 2 descendant
haplotypes then
    if tract is not contained in previously computed tract then
        report as an IBD tract
    end
else
    push children nodes on stack
end
```

Algorithm 1: Tractatus (error free)

6.2.2 The Tractatus algorithm with errors and allele mismatches

Incorporating base call errors and additional variability gained after differentiation from the least common ancestor requires additional computations on the Tract tree and a statistical modeling of haplotype allele mismatches. The Tractatus algorithm with errors is parameterized by an estimated probability of error or mismatched alleles p_t, a p-value threshold corresponding to a test for the number of errors in a tract p_h, a minimum length partial IBD tract l, and a minimum length of calling a full IBD tract L (or alternatively P(IBD) as defined in Equation 6.1). We will, in turn, explain the significance of each parameter.



	vv_i	···	···	v_{j-1}	v_j	v_{j+1}	···	···	v_k
h_0	0	···	···	0	0	0	···	···	0
h_1	0	···	···	0	1	0	···	···	0

↓

d_0	2i	···	···	2j-2	2j	2j+2	···	···	2k
d_1	2i	···	···	2j-2	2j+1	2j+2	···	···	2k

Figure 6.3: Construction of the Tract tree and running Tractatus on example input with allele mismatches. (A) h_0 and h_1 share a tract IBD in the interval $[v_i, v_k]$ with a single allele mismatch at v_j. (B) By the construction of the Tract tree, there must be some path (here shown as a single edge but it may be split into a path by other haplotypes) from root to internal node that includes both $[v_i, v_k)$ and (v_k, v_i).

The algorithm proceeds similarly to the error-free case. We build the tractized haplotypes, Tract tree and populate necessary data structures with a DFS. Because errors and additional variation now exist which can break up tracts (and therefore paths in the Tract tree), we compute partial tracts as evidence of IBD. We compute a DFS from the root, and a maximal partial tract is saved when the algorithm arrives at a node with path length at least l and at least 2 haplotype descendants. If we find a partial tract in a subsequent traversal, we can check in $O(1)$ time is it is contained in a maximal partial tract already computed. Figure 6.3 shows an example of the Tract tree construction with a single allele mismatch.

Because we computed the partial tracts using a DFS, the tracts are ordered by starting position. For each tract, tracts starting at a position prior and including a subset of the same haplotypes are combined if the extension is statistically probable. To determine the scan distance, we can compute a probability of observing a partially shared tract of length l given a window distance w (or this can be user defined). Assuming the generation of errors is independent and the probability of generating an error is p_e, we model the probability of generating at least k errors in an interval of l_i in t haplotypes as a Poisson process with $\lambda = p_e l_i t$. For each extension we calculate the probability of observing at least k mismatches and accept the extension if the probability is greater than p_h. The parameter p_t is used as an approximation of p_e. The haplotype consensus sequence of the tract is taken by majority rule at each variant position.

Pseudocode is given in Algorithm 2. While the algorithm is parameterized with five parameters, they are optional and default values are suitable in most cases.

Construction of the Tract-tree takes $O(mn)$ time and space. $O(mn)$ time is needed to prepare

```
input : $m$ haplotypes each of length $n$, partial tract length $l$, minimum length $L$ or IBD
        probability $p$, p-value threshold $p_h$, estimated probability of error $p_t$, length of
        scan $w$
output: set of IBD tracts
$H \leftarrow$ tractized haplotypes
$T(H) \leftarrow$ Tract tree of $H$
Post-order DFS of T(H) to compute descendant haplotypes from each node
DFS of T(H):
if path in DFS is longer than $l$, node has at least 2 descendant haplotypes, and is maximal
then
    add partial IBD tract to set of tracts $S$
else
    push children nodes on stack
end
for tract $s \in S$ do
    Check for extension in previously processed tracts within scan region $w$
    Compute probability according to number of errors in extension, $p_t$, the length of the
    extension, and the number of individuals
    If probability $> p_h$, merge tracts
end
for tract $s \in S$ do
    If length of $s$ is greater than $L$ or $P(IBD) > p$, report as IBD tract
end
```

Algorithm 2: Tractatus (with errors)

data structures and compute maximally shared partial tracts (DFS). A tract can be checked if it is contained in a previously processed tract in $O(1)$ time. It takes $O(mnw)$ to merge partial IBD tracts under reasonable assumptions of the merging process and in the worst case when we have to extend tracts covering the entire matrix, thus yielding Theorem 6.

Theorem 6. *Given a set of m haplotypes each of length n, a scan distance w and a set of partial haplotype tracts, Algorithm 2 computes the Tract tree and set of IBD tracts in time and space $O(mnw)$.*

6.2.3 Extensions for homozygous haplotypes

A particular class of identical-by-descent relationships are long regions of extended homozygosity in genotypes. The two dominant concepts of extended regions of allelic homozygosity are the homozygous haplotype (HH) concept introduced by Miyazawa et al. (2007) and the well-known region or run of homozygosity (ROH). A HH is defined as a genotype after the removal of heterozygous variants such that only homozygous variants remain. Miyazawa *et al.* 2007 compared every pair of HH in a small cohort and reported regions of consecutive matches over a threshold. ROHs are defined as extended genomic regions of homozygous variants allowing for a small number of heterozygous variants contained within. We can rigorously capture both concepts using Tractatus.

A naive model for computing HH would consider each heterozygous site as a wildcard allowing for either the 0 or 1 allele. A haplotype with k heterozygous sites would require insertion of 2^k haplotypes into the Tract tree. This immediately suggests a fixed-parameter tractable algorithm using the same machinery as Tractatus. However, we can remove the dependence on k using a key insight regarding the structure of the Tract tree and tractized haplotypes.

Errors split tracts in the Tract tree such that the shared tract fragments are on different paths from the root. Instead of encoding all 2^k possible haplotypes, we simply remove the heterozygous alleles from the tractized string. Because the position is inherently encoded in the tractized string, the removal of the heterozygous alleles would have the same effect as an error. Therefore, if we encode genotypes by simply removing heterozygous variants in the tractized string, we can run Algorithm 2 to produce all the homozygous haplotypes for a set of genotypes in linear time and space.

6.3 Results

The principle advantages of Tractatus over existing methods are the theoretically guaranteed subquadratic runtime and exact results in the error-free case which translate to improved results in the case with errors and allele mismatches. We evaluate the runtime of Tractatus against a generic algorithm that processes individuals in pairs using phased HapMap haplotypes from several populations. We then compare the power and false positive rates of both Tractatus and GERMLINE which is a leading method for IBD inference (Gusev et al. 2009). Finally, we show an application of Tractatus-HH by inferring homozygous haplotypes in a previously known homozygous region in the Simons Simplex Collection genome-wide association study data (Fischbach and Lord 2010).

6.3.1 Tractatus vs. pairwise algorithm runtimes

To evaluate the runtime of Tractatus versus pairwise methods, we implemented the pairwise equivalent algorithm which iterates through pairs of individuals and reports tracts of variants occurring in both individuals over some threshold length of variants. The data consist of phased haplotypes from HapMap Phase III Release 2 in the ASW, CEU, CHB, CHD, GIH, JPT, LWK, MEX, MKK, TSI, and YRI populations (International HapMap Consortium 2003). Figure 6.4 left shows the independence between chromosome and computation time for the Tractatus suffix tree and the pairwise

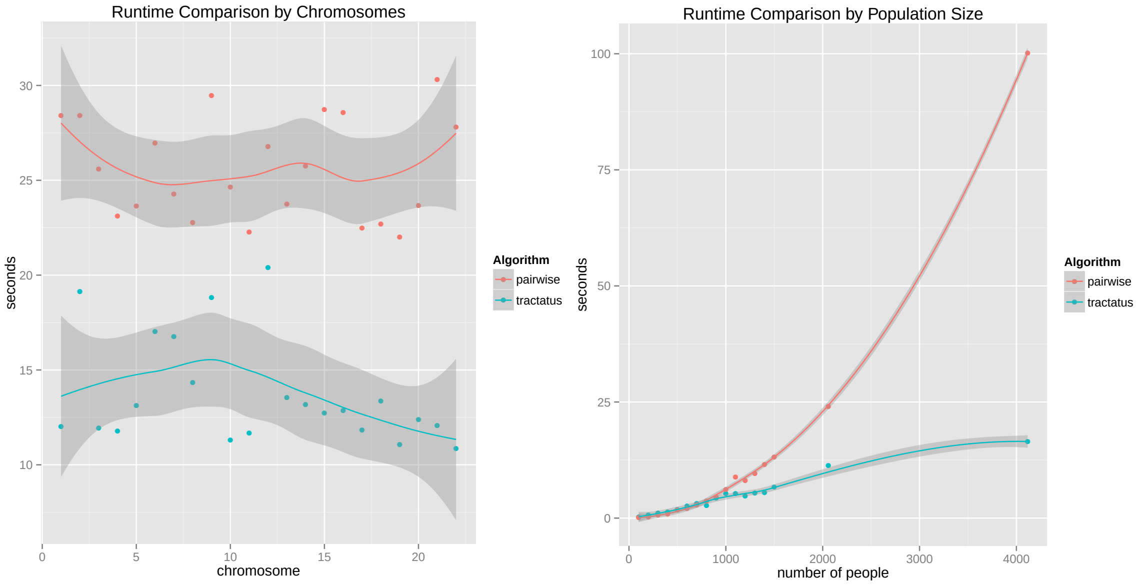


Figure 6.4: Left: Tractatus and the pairwise algorithm were run on haplotypes from each chromosome of all HapMap populations for a minimum tract length of 100, and a randomly selected interval of 1000 variants. The experiment was repeated 100 times for each chromosome and elapsed time was averaged. Right: Tractatus and the pairwise algorithm were run on a randomly selected interval of 1000 variants from chromosome 22. The population size varied from 100 to twice the actual population size by resampling haplotypes with a 0.05 allele switch rate (per base).

algorithm. Because the runtime of each algorithm does not depend on the chromosome, we varied the population sizes while keeping the number of variants constant for chromosome 22. Figure 6.4 right shows the quadratic computation time growth for the pairwise algorithm while Tractatus tree construction remains linear in the number of individuals.

6.3.2 False positive rates

Because it is difficult to construct a gold-standard baseline of true IBD regions in real data, our false positive rate and power calculations are performed on simulated data. To estimate the false positive rates for GERMLINE and Tractatus we simulated haplotypes at random and generated a single IBD region defined as having identical haplotype alleles in the region of IBD. We generated 100 haplotype matrices where $m = n = 500$ for all possible combinations of the number of individuals sharing a segment IBD $\in [3, 5, 10]$, the number of variants in the IBD region $\in [50, 60, 70, 80, 90, 100, 150, 200]$ and the single base substitution error rates $\in [0.0, 0.01, 0.05]$. In total, we generated 7200 haplotype matrices but aggregated the data across the number of individuals and variants in the IBD region because the false positive rates did not vary over these dimensions.

Table 6.1 shows that both algorithms have very low false positive rates in terms of the number of bases incorrectly called in an IBD region. However, Tractatus incorrectly calls less individuals in IBD regions than GERMLINE. In this experiment, IBD regions were generated in block sizes and GERMLINE benefits from calling IBD regions in terms of blocks or windows. GERMLINE and Tractatus call a similar amount of bases IBD because Tractatus can over-estimate the ends of blocks. However, when individuals are compared, Table 6.1 shows that Tractatus computes a significantly smaller number of false positive IBD regions.

Table 6.1: False positive rates for the GERMLINE (G) and Tractatus (T) algorithms as a function of error rate. Each row corresponds to 2400 randomly generated haplotype matrices. The error rate was varied in a simulated haplotype matrix containing a single IBD region. False positive rates were calculated in terms of the number of non-IBD bases being called IBD (bases) and the number of individuals called IBD who were not in an IBD region (people) for the GERMLINE and Tractatus algorithms.

error rate	G FPR bases	T FPR bases	G FPR people	T FPR people
0.0	$1.3 \cdot 10^{-4}$	$1.16 \cdot 10^{-4}$	0.016	$2.13 \cdot 10^{-3}$
0.01	$1.2 \cdot 10^{-4}$	$1.11 \cdot 10^{-4}$	0.012	$8.72 \cdot 10^{-3}$
0.05	$6.1 \cdot 10^{-5}$	$4.18 \cdot 10^{-5}$	0.015	$7.43 \cdot 10^{-3}$

6.3.3 Power

We apply Tractatus and GERMLINE to the simulated data from Section 6.3.2 and estimate power by computing the number of times GERMLINE and Tractatus correctly call the IBD region in terms of variants and individuals. We considered an individual being called correctly if an IBD region was called and overlapped anywhere in the interval of the true IBD tract. We set the -bits and min_m options of GERMLINE to 20 and 40 respectively which sets the slice size for exact matches to 20 consecutive variants and the minimum length of a match to be 40 MB (which corresponds to 40 variants in our simulated data). For a valid comparison, we set Tractatus to accept partial tract sizes of 20 variants and a minimum length of an IBD region to 40 variants.

Figure 6.5 shows the power of GERMLINE and Tractatus to infer IBD as a function of IBD region length, number of haplotypes sharing the region, and the probability of base call error. Figure 6.5 right displays a *jagged* curve for GERMLINE which is likely due to the algorithmic dependence on window size. Both algorithms perform relatively well for shorter IBD tracts but Tractatus is clearly more powerful when the number of haplotypes sharing the tract increases or the base call error rates are low. Additionally, the minimum partial tract length for Tractatus could be lowered to increase the power to find smaller IBD tracts (at a cost of higher false positive rates). Another interesting observation is that both GERMLINE and Tractatus are able to perfectly infer all individuals sharing

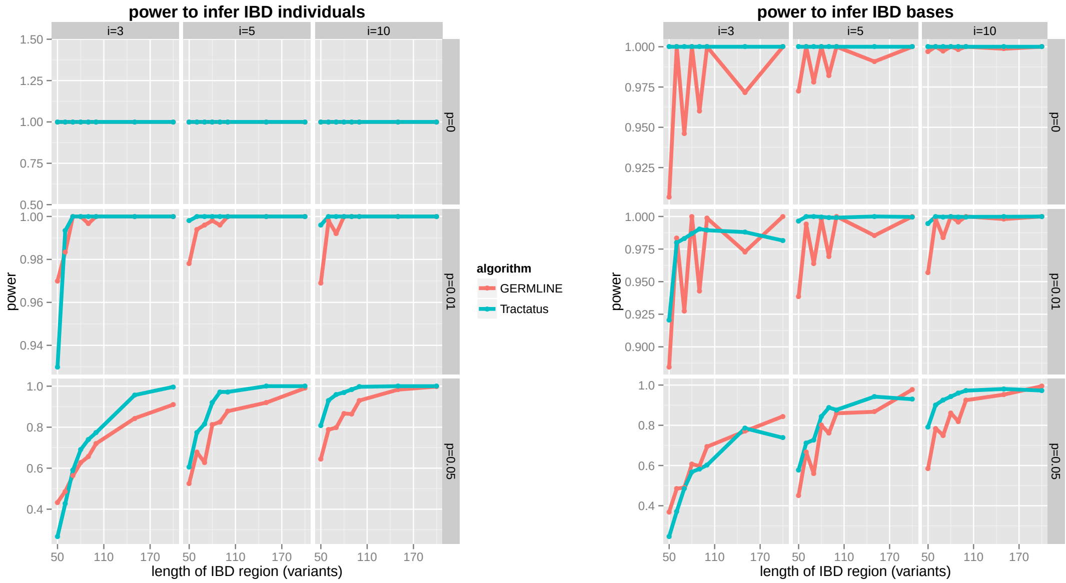


Figure 6.5: The power to infer IBD by individual haplotypes (left) and variant bases (right) as a function of the length of the IBD region in variants (x-axis), the probability of base call error (p), and the number of individual haplotypes sharing the IBD segment (i).

the IBD region in the perfect data case, but, GERMLINE is unable to compute the entire IBD interval in some data.

6.3.4 Homozygous haplotypes in autism GWAS data

As a proof of concept for Tractatus-HH, we extracted a 250kb genomic region identified as having a strong homozygosity signal in the Simons Simplex Collection (Gamsiz et al. 2013). The families analyzed include 1,159 simplex families each with at least one child affected with autism and genotyped on the Illumina 1Mv3 Duo microarray. Gamsiz et al. (2013) approached the problem by treating a homozygous region as a marker and testing for association or burden for the region as a whole. Our analysis shows that regions of homozygosity are more complex than previously assumed and there can be multiple regions overlapping and sharing some segments of homozygous haplotypes but largely different in other segments (Table 6.2). We found more individuals possessing a homozygous haplotype than Gamsiz *et al.* 2013 because the probability of generating an error or heterozygous site was set to a large value (0.1) but in general this parameter can be adjusted to be more conservative.

Table 6.2: Analysis of a $250kb$ region of homozygosity in the Simons Simplex Collection. The homozygous interval is defined as a region start and end in terms of variants in the genomic interval, a number of individuals (size), and the number of individuals unique to the particular homozygous haplotype group (unique). One region is dominant and contains most of the individuals, but there are smaller regions with some overlap that contain unique individuals not sharing a homozygous haplotype with the larger region.

region start	region end	size	unique
0	111	20	10
0	109	20	12
0	109	252	238

Part IV

Conclusion

Chapter 7

Discussion

7.1 Haplotype phasing

7.1.1 Deletion inference

Instead of small recurrent deletions, DELISHUS can easily be modified to target different deletion architectures. Under Formulation 1, DELISHUS computes all inherited or *de novo* deletions with maximal clique size above a user-defined threshold. However, researchers may want to find deletions that are large and rare instead of small and recurrent. By restricting edges in the deletion graph to within trios or pairs, DELISHUS essentially reduces to the algorithm of Conrad et al. (2006). The new deletion graph would be much less connected and thus the threshold can be lowered (to about two in our internal tests) while retaining a tolerable false positive rate.

While we have found Formulation 1 to be the most useful, it only considers the case for which an error might convert a normal inheritance pattern to an evidence of deletion. However, all potential conversions between deletion categories are possible (Fig. 7.1). Formulation 3 represents an alternative to Formulation 1 which models deletions and genotyping errors without the usage of a threshold.

Formulation 3. *We are now allowed to correct any* $1 \to X$ *and any* $X \to 1$ *in* M'*. Find the minimum number of switches from* $1 \to X$ *or* $X \to 1$ *such that all cliques are disjoint.*

Regardless of the formulation, there may still be other types of errors in SNP data such as technical artifacts producing completely erroneous SNPs. These are usually filtered in a preprocessing quality control (QC) step, but it is often advantageous to allow DELISHUS to process the pre-QC data. For example, a small deletion encompassing a single SNP and associated to the phenotype of

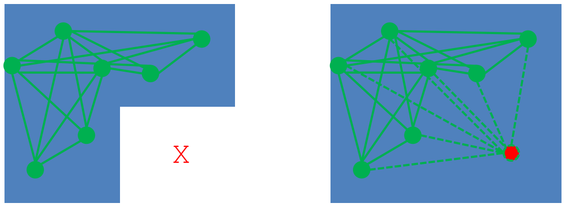


Figure 7.1: M' is shown on the left with a superimposition of evidence of deletion vertices and edge connections. On the right, we demonstrate that making one $X \rightarrow 1$ correction unifies evidence of deletion sites into one larger deletion.

interest could mimic the behavior of a technical artifact and should not be removed prior to running DELISHUS.

Before experimental wet-lab validation, recurrent deletions should be prioritized in terms of association to disease. Because the inferred deletions are hemizygous, a natural choice for association is the transmission disequilibrium test (TDT) which measures over-transmission of the deletion to affected children (Spielman and Ewens 1996; Spielman, McGinnis, and Ewens 1993). However, due to the transmitted deletion signal being undetectable in a single individual, DELISHUS cannot compute aberrations in parents that are not transmitted. Deletions that remain undetected in non transmitted cases will introduce bias in the TDT. The sibling transmission disequilibrium test (sib TDT) is an alternative to the TDT when variation cannot be detected in parents (Spielman and Ewens 1998). In the sib TDT, data from unaffected siblings are used in place of parents making it a more appropriate test for association if unaffected siblings are available.

Another important step before enacting expensive experimental validation is providing additional computational support for the deletion calls. Both signal intensities from probes and sequencing data can be used for orthogonal analyses to DELISHUS.

Signal intensities also provide a valuable resource for deletion inference and programs like PennCNV have been designed to exploit the change in signal intensity in deletion intervals (Wang et al. 2007). But HMM-based algorithms for deletion inference in signal intensity data have difficulties inferring small deletions with few probes spanning the interval. An alternate approach considers the distribution of log R ratios for each deletion. The log R ratio distribution within the deletion interval in individuals harboring the deletion can be compared to the theoretical null distribution. The same computation is possible for the set of individuals without deletions.

As sequencing becomes cheaper and the sequencing of thousands of individuals becomes feasible, GWAS will shift from SNP arrays to deep genome or exome sequencing. Due to the higher density of variant calls in sequence data, DELISHUS will be able to infer deletions at a higher resolution than array data. The 1000 Genomes Project is one example of a large scale sequencing project where a number of the genomes sequenced include parent-child trios and pairs of HapMap individuals. DELISHUS can be used on the SNP data to validate previous calls in the HapMap data.

7.2 Haplotype assembly

The size of the haplotype blocks produced and, ultimately, the quality of the assembled haplotypes is a function of several factors. The primary difficulty for obtaining large haplotype blocks is the small nature and lack of diversity of insert lengths. We demonstrated a novel modeling and computational method that begins to address this difficulty by exploiting shared IBD haplotype structure. In general, assembling the haplotypes of related individuals has considerable benefits which help overcome undesirable properties of the sequencing data. The first benefit comes from the extra coverage on the shared haplotype, which helps in differentiating actual phasings from sequencing errors. However, the most notable advantage is being able to include more SNPs into the haplotype assembly which helps extend the assembly (past regions of low read coverage for example). But, the major advances in block sizes will likely be the result of novel experimental procedures and technologies; for instance, not only do the single molecule sequencers promise larger read lengths, they also enable the inclusion of multiple and large insert lengths. Projects like the Assemblathon are proving that chromosome-wide haplotype assembly is possible using only second generation sequencing technologies (Bradnam et al. 2013).

7.2.1 Diploid genomes

Due to the exponential solution space, complicated error signatures, and other factors, genome-wide diploid haplotype inference is still a difficult task. HapCompass is a proven framework for haplotype assembly but there are a number of extensions that may improve results. For instance, we did not mention the usage of base call or read mapping quality scores in our computations. HapCompass can filter sequence reads or base calls but a more elegant solution would be to convert the base call quality score into a probability the allele was called correctly. This probability can then be incorporated into the pairwise phasing likelihood for variants in the compass graph G_C. Also we demonstrated in the Pacific Biosciences experiments that the choice of assembly method should be informed by the sequencing technology and desired result. The Levy et al. (2007) method mapped more fragments error free than HapCompass but contained many more single base changes in fragments required to reproduce the inferred haplotypes. Considering the Pacific Biosciences data has very high error rates and generating an error-free read is unlikely, a solution with the minimum number of corrected errors is likely preferred over a solution that successfully maps more fragments without errors.

7.2.2 Polyploid and tumor genomes

Organisms having more than two sets of homologous chromosomes are becoming the target of many research groups interested in studying the genomics of disease, phylogenetics, and evolution (Chen and Ni 2006; Leitch and Leitch 2008). Polyploidy typically occurs in human disease due to the duplication of a particular chromosome, for example, in Edwards, Patau, and Down syndrome. While far fewer mammalian organisms are polyploid, specific mammalian cells may undergo polyploidization, for example in human liver hepatocytes (Gentric, Celton-Morizur, and Desdouets 2012). In addition, polyploid organisms are ubiquitous in the Plant and Fungi clades, present in crops that we ingest, convert into bioenergy, and feed to livestock. Understanding the genomics of both the desirable – e.g. increased crop yield – and undesirable – e.g. susceptibility to disease – properties of plants may lead to critical advances in many research areas but requires untangling the polyploid genome and its variation. As more data becomes available, haplotype assembly will become an essential component for understanding the relationship between genome and phenome in polyploid organisms.

Opportunities exist to extend HapCompass-Tumor/Poly to address some of the limitations in the current model. First, HapCompass-Tumor/Poly only computes a single solution when the compass graph model allows computation of suboptimal solutions. Phase extension in G_g is deterministic but many highly probable suboptimal solutions may exist. As long as the number of alternative disjoint paths is bounded by a low degree polynomial, we can carry these partial solutions to the assembly step and report multiple haplotype assemblies.

Second, incorporating *a priori* knowledge of haplotype distributions from population samples or long read lengths can improve the assembly. For example, we assumed each valid haplotype phasing for a cycle in G_C is equally likely. However, this assumption can be easily modified to accommodate known haplotype likelihoods in the area (e.g. linkage disequilibrium). Consider a collection of valid disjoint paths for a cycle in G_C; if the probabilities of all phasings are equally likely and the edge extension has i distinct matchings, then each matching is given a weight $\frac{1}{i}$. If, however, one of the haplotypes in an extension is never observed in the population, HapCompass-Tumor/Poly could penalize the extension.

A related application of HapCompass-Tumor/Poly is in cancer panomics. Much attention in cancer research has been focused on allelic specific expression (ASE). Studies have shown that germline ASE is associated with cancer risk (Gao et al. 2012; Valle et al. 2009); and somatic ASE is associated with tumor development (Tuch et al. 2010). ASE in cancer was found not only correlated with CNAs (Tuch et al. 2010), but also with allelic specific methylation (ASM) (Lin, Giannopoulou, et al. 2013). Existing algorithms for detecting ASE with RNA-seq and detecting ASM with Bisulfite-Seq do not usually make use of phased genotype information (Fang, Hodges, et al. 2012; Tuch et al. 2010). It is possible that phased haplotypes from whole genome sequencing of tumor samples can be used as a reference for RNA-seq and Bisulfite-seq alignment when such data is available.

Finally, we consider the connection between the viral quasispecies reconstruction (VQR) problem and polyploid haplotype assembly. VQR aims to compute the spectrum of viral quasispecies haplotypes from the sequence reads of a heterogeneous viral sample. The problems of haplotype assembly and VQR are similar, but the research literature is largely independent due to the inability of haplotype assembly algorithms to model more than two sets of homologous haplotypes. However, it is possible to model VQR with HapCompass-Tumor/Poly by leaving the number of haplotypes in the sample (k) as an unknown parameter. Two possible approaches include inferring the number of quasispecies *a priori* and then performing haplotype assembly with k unique haplotypes or computing assemblies for a number of different k and comparing the quasispecies solutions. But, using a general haplotype assembly tool for VQR does not take advantage of two critical properties of most viral genomes: (1) knowledge of the phylogenetic relationships between mutations is known for well-studied viral genomes especially those under selective pressures from treatment and (2) the genomes are many orders of magnitude smaller than eukaryotes.

7.3 Identity-by-descent

The importance of provable bounds and exact solutions is exemplified in Section 6.3 and, in particular, Figure 6.5. Even in the error free case, GERMLINE approximates computing IBD tracts by processing windows or vertical slices of the haplotype matrix. Tractatus is able to compute maximally shared partial tracts exactly (which are exactly the IBD tracts in the error-free case). Moreover, the inexactness of GERMLINE, due to the dependence of hashing windows, is exacerbated in the case of errors. If errors fall in a pattern that cause individuals sharing a segment IBD to hash to different values, then GERMLINE produces false negatives. Tractatus computes all maximally shared partial tracts without dependence on windows. Lastly, in the worst case, the number of matches per word is quadratic giving GERMLINE a complexity quadratic in the number of individuals. Even though this is unrealistic in practice, Tractatus compresses individuals sharing a partial tract into a single path of the suffix tree.

The Tract tree alone is an interesting data structure with many possible applications. Once the Tract tree is computed for a set of haplotypes, the statistics of constructing the mosaic of tract combinations can be done rigorously such that sampling can be implemented in an order independent manner satisfying the exchangeability property. For the HMM constructions, the availability of the complete set of tracts would provide a rigorous basis for defining the transition probabilities and overall linear time construction. For the graph clustering methods, the Tract tree represents tracts occurring multiple times together and thus this construction will maximize the power in association studies.

Unfortunately, the issue of acquiring haplotypes remains. Almost exclusively, algorithms for computing IBD require haplotypes due, in part, to the higher power to infer a more subtle IBD sharing than in genotype data. However, this is not a major roadblock considering haplotype phasing algorithms can be highly parallelized or made more efficient using reference panels. Additionally, haplotype assembly algorithms are very efficient and can extend genome-wide (Aguiar and Istrail 2012).

Our analysis of the autism genome-wide association study data shows that homozygous regions cannot simply be treated as a biallelic markers. Distinct homozygous haplotypes, while having a similar signature of homozygosity, can be composed of entirely different alleles. These findings suggest that homozygous regions should be considered as complex, multi-allelic markers.

We note that a similar linear time construction could be used for constructing a Tract tree for a set of haplotypes where there is known genetic information about the distance between variants as in the Li-Stephens PAC model (Li and Stephens 2003). The genetic distance can be modeled approximately as an integer and used in a similar encoding to compress "identical" tracts.

7.4 Future Work

7.4.1 Haplotype specificity

An interesting problem to investigate is haplotype specificity in other areas of life sciences, for example, transcriptomics. Haplotype specific interactions are a component of the genetic heterogeneity puzzle which has yet to be fully explored. One goal is to investigate haplotype specific expression at the gene or splice-form level (Turro et al. 2011). Allelic expression imbalance is not only important in the context of medical genomics but also independently interesting in the context of haplotype assembly of transcriptome data and splice-form identification. The genome and transcriptome sequences can be compared using heterozygous variants as a basis for haplotype assembly of the transcriptome and estimating allelic expression imbalance.

7.4.2 Identity-by-descent in genotypes

The haplotype phase uncertainty that exists in genotypes creates a number of problems for identity-by-descent tract inference in genotype data. IBD inference methods that rely on allele sharing require much longer tracts to differentiate IBD and IBS haplotype tracts. Furthermore, subquadratic algorithms (in terms of the number of individuals), such as Tractatus and GERMLINE, use suffix trees and hashing respectively; both methods have difficulties modeling the heterozygous sites which are fundamentally wildcard (or don't care) characters. An extremely important open problem is to develop IBD inference algorithms in genotypes that are subquadratic in terms of the input and retain the resolution of IBD algorithms on haplotypes. We believe an approach exploiting the Tract tree would be able to infer IBD in genotypes in subquadratic time perhaps with a direct application of the Tractatus-HH algorithm. However, the number of heterozygous variants is usually very high, so additional computation would be required to handle the large quantity of ambiguous sites.

7.4.3 Tractatus applications

Tractatus is applicable in areas external to IBD inference. Primer design is essential for genotyping, allele specific PCR, and sequencing. In the context of disease, primers must be designed to maximize the typing of variants associated with disease susceptibility or resistance. HIV subtype-aware variant identification is an area where there is need for such tools. The Tractatus framework can be modified to compute the minimum number of probes to cover HIV viral haplotype sequences in specific subtypes. The computation of the minimum set of probes can be defined for various objectives, for example, (1 – unlimited resources) covering $x\%$ of the viral haplotypes, (2 – uncertainty in sequence) covering at least 1 of the IUPAC-code expanded sequences for each haplotype, and (3 – stochastic probe binding) allowing y mismatches between probe and sequence. These problems can be solved exactly with a combination of Tractatus and graph theoretic optimizations, e.g., set cover for (2) and maximum independent set for (3).

7.4.4 Haplotype assembly of tumor genomes

Generally, it is believed that cancer can be viewed as a evolutionary process, where normal cells acquire somatic mutations followed by clonal expansion (Greaves and Maley 2012). The different sub-clones, related by a phylogenetic tree, usually co-exist in equilibrium controlled by the micro-environment. This equilibrium can be disturbed when either some cell acquires an advantageous mutation that favors their fitness; or when external forces, such as chemotherapy, favor certain clones. Most of the algorithms for detecting tumor somatic mutations assume that there is a major clone and only detect those mutations in the major clone (Boeva et al. 2011; Gusnanto et al. 2012). In the clinical setting, failure to detect the variants in minor clones can lead to missed treatments for a cancer patient. Furthermore, the ability of detecting different subclones and their frequencies can be useful in monitoring the cancer treatment progress.

Future work could include extensions to HapCompass-Tumor to accommodate clonal haplotypes. For simplicity of the discussion, we will focus on single point mutations, although it is trivial to include small indels as well. While the tumor somatic SNV density varies widely between different cancer types, they occur at orders of magnitude less than germline SNP (Greenman et al. 2009). It is then safe to assume that the intermediate "neighboring" mutations of most somatic SNVs are germline SNPs. First, we can phase the tumor sample assuming there are two haplotypes with different proportions, using the SNPs from the germline mutation only. Then we identify those somatic mutations that are linked to germline mutations to infer tumor haplotypes. The clonal frequencies and normal contamination can be inferred using finite mixture models in a Baysian framework.

7.4.5 Joint haplotype assembly, phasing, and identity-by-descent inference

An interesting and emerging area of research focuses on combining the statistically dominated world of haplotype phasing with the combinatorics of haplotype assembly into a unified haplotype reconstruction framework. But combining haplotype phasing, assembly, IBD inference, and polyploidy into a single framework complicates the model and algorithms. The more variables to consider, the more complex the algorithm, and haplotype reconstruction algorithms that do not extend genome-wide are significantly less useful than genome-wide algorithms. One possibility is to extend the HapCompass algorithm, which represents the relationship between alleles encoded in sequence reads as a likelihood, to incorporate evidence from other sources. For example, haplotypes shared identical-by-descent restrict the solution space of phased haplotypes. Reference haplotypes can be used to increase the likelihood of haplotype assemblies which share haplotype tracts at the population level.

7.4.6 Haplotypes to phenotypes

The problem of mapping genomic variation to genes and pathways is an open problem with potential medical applications. One difficulty with interpreting results from large genome-wide associations is that variants with low p-values often cluster in regions of no known functional significance or regions overlapping many genes. Both of these situations are commonplace due to the complex patterns of linkage disequilibrium throughout the human genome and incomplete knowledge of functional components. Without a discernible connection between variant and protein, it is difficult to design targeted drug therapy. But, with a more complete knowledge of factors influencing phenotype, the links between genome and phenome can start to be more completely inferred. For example, along with single variant associations, information extracted from IBD shared haplotypes and expression data (e.g. expression quantitative trait loci) can inform exactly which variants influence the expression of pathways and genes. Moreover, reconstructions of evolutionary history, for example with ancestral recombination graphs, can shed light on the particular mutations explaining the segregation of cases and controls.

Chapter 8

Summary of contributions

8.1 Haplotype phasing

With increasingly dense SNP arrays and whole-exome sequencing becoming commonplace for studies of association, we are now ready for the genome-wide search for smaller deletion variants. Although the power of these newer technologies is enormous, genetic heterogeneity remains a daunting challenge and the identification of all polymorphisms is paramount to the understanding of complex disease. While many large genomic deletions have already been found and replicated, the problem of identifying small deletions remains a considerable challenge.

In this dissertation we presented three computational problems related to deletion inference in SNP data with a focus on small recurrent deletions in autism. We introduced the DELISHUS algorithmic framework for computing inherited deletions, *de novo* deletions, and critical regions. Using a formulation inspired by the complexity of the deletion signature in autism, we showed that the problem of computing all inherited and *de novo* deletion configurations in SNP data can be solved in polynomial time (and empirically within minutes). We presented systematic methods to compute false positive rates and power for the DELISHUS and single individual algorithms and demonstrated how to use the calculations to evaluate algorithmic performance and tune the threshold parameter. Comparisons of power while controlling for false positive rates, show that the DELISHUS algorithm excels at inferring small recurrent deletions. We also showed that finding critical regions of recurrent deletions may also be solved in polynomial time. Furthermore, we have shown that long-range phasing using Clark consistency graphs is practical for very large datasets and the accuracy of the algorithm improves rapidly with the number of individuals in the dataset.

8.2 Haplotype assembly

8.2.1 Diploid genomes

Haplotype assembly is becoming increasingly important as the cost of sequencing plummets and more genome-wide and whole-exome studies are conducted (Levy et al. 2007; Tewhey et al. 2011). We have designed and implemented a haplotype assembly algorithm that is widely applicable to these studies because it does not make any prior assumptions on the input data. Through the use of simulations, we show that supplementing 1000 Genomes Project data with sequencing data of a particular type connects G_C, enabling the haplotype assembly of entire chromosomes. We described the fragment mapping phase relationship and Boolean fragment mapping metrics that capture the quality of the haplotype assembly through support from mapped fragments. These metrics can be used independent of the algorithm and without knowing the true haplotypes to evaluate the quality of the haplotype assembly.

We compared HapCompass to leading haplotype assembly software packages that can also process arbitrary input sequence data: HapCut, the Levy et al. (2007) algorithm, and the Genome Analysis ToolKit's read-backed phasing algorithm. HapCompass is shown to be more accurate on real 1000 genomes data for the BFM and FMPR metrics. We also show that HapCompass is more accurate when we supplement the existing 1000 genomes real data with simulated Illumina reads for BFM, FMPR and haplotype switch metrics on haplotype blocks of unprecedented size. As high-throughput sequencing becomes more available to a greater number of researchers, we believe HapCompass will provide a valuable tool to quickly and accurately identify the haplotypes of diploid organisms.

8.2.2 Polyploid and tumor genomes

In this work, we developed algorithms and models for tumor genome assembly building on our existing haplotype assembly framework HapCompass. We demonstrated how to model tumor haplotype heterogeneity and haplotypes containing CNAs and translocations. The HapCompass-Tumor/Poly algorithm was presented using the combined evidence of cycles in G_C and disjoint paths in G_h to inform which haplotype assemblies in G_g are probable. Finally, we evaluated the HapCompass-Tumor/Poly algorithm on simulated cancer data showing that, while the accuracy is a function of many parameters, including the level of cancer genome heterogeneity, we are still able to produce accurate haplotype assemblies.

8.3 Identity-by-descent

Lastly, we described the Tractatus algorithm for computing IBD tracts with and without errors and homozygous haplotypes. Tractatus represents the first provably exact algorithm for finding multi-shared IBD tracts given a set of haplotypes as input; it computes all subsets of individuals that share tracts and the corresponding shared tracts in time linear in the size of the input. By starting from an exact and rigorous algorithmic baseline, we are able to modify downstream decisions based on the global IBD tract decomposition. We demonstrate that the runtime of Tractatus grows linearly with the size of the input while a generic pairwise algorithm that process individuals in pairs grows quadratically using phased HapMap haplotypes from several populations. Also, we exhibit superior statistical power to infer IBD tracts with less false positives than GERMLINE. Finally, with a conceptual change to the interpretation of genotypes, we show that homozygous haplotype inference in genotypes can be modeled in the same Tractatus framework and demonstrated Tractatus-HH in a previously known homozygous region of the Simons Simplex Collection autism data.

8.4 Concluding Remarks

Although many pieces of the genetic heterogeneity puzzle have yet to be fully understood, haplotypes and haplotype reconstruction algorithms are emerging as an integral component. Deletion haplotypes have been strongly associated with numerous conditions. Haplotype phasing and assembly algorithms provide the fundamental sequences for phylogenetic reconstruction, genotype imputation, linkage disequilibrium, and characterizing the connection between genotype and phenotype. Identity-by-descent allows for the mapping of regions associated with disease and inference of population substructure. As the technology changes, new problems will undoubtedly emerge, but haplotypes and haplotype reconstruction algorithms will remain fundamentally important to population genomics and medical bioinformatics.

Appendix A

Supplemental figures

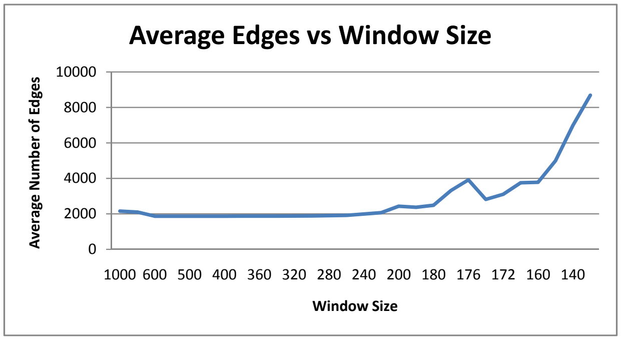


Figure A.0.1: The average number of edges per window size stays relatively constant until a window size of about 180. The graph becomes more connected at this point likely because the window size is small enough to not be largely affected by recombination (but still large enough for the shared tracts to not likely be IBS).

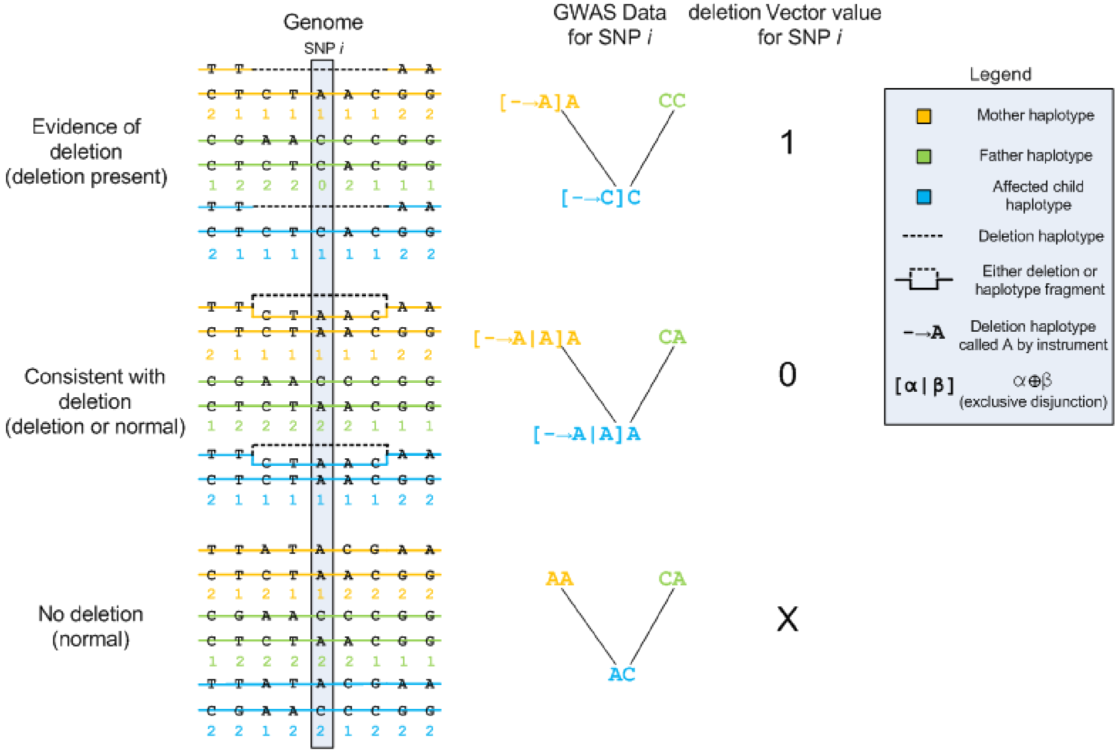

Genome
SNP i
GWAS Data for SNP i
deletion Vector value for SNP i
Evidence of deletion (deletion present)
Consistent with deletion (deletion or normal)
No deletion (normal)
[-→A]A
CC
[-→C]C
1
[-→A|A]A
CA
[-→A|A]A
0
AA
CA
AC
X
Legend
Mother haplotype
Father haplotype
Affected child haplotype
Deletion haplotype
Either deletion or haplotype fragment
-→A
Deletion haplotype called A by instrument
[α|β]
α⊕β (exclusive disjunction)

Figure A.0.2: The evidence of deletion, consistent with deletion, and no deletion Mendelian inheritance patterns are shown with the true genome sequence and deletion vector calls.

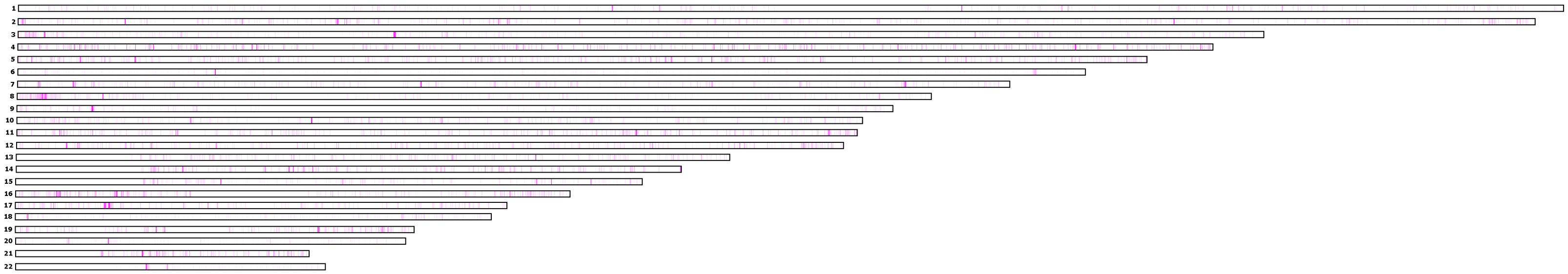


Figure A.0.3: DELISHUS produces a deletion map for chromosome-wide or genome-wide runs. This figure is the deletion map for the AGRE autism GWAS autosomal data. DELISHUS was run with a threshold of 5 for inherited deletions. The density of the deletion is represented by the intensity of the violet color and normalized by chromosome.

Appendix B

Supplemental tables

Probability of Error per Site	For all SNP-trio pairs, we add a Mendelian error according to this probability (assumed independent for each site).
Interval Length	The exact length of the generated deletion.
Trios in Deletion	The exact number of trios sharing the generated deletion.
Probability of evidence of deletion	The probability a SNP is an evidence of deletion site within the generated deletion interval.
Coefficient of Genotype Error Call	The objective function cost for calling an evidence of deletion site a genotyping error (parameter k_1 in our objective function)
Coefficient of Inherited Deletion Call	The objective function cost for calling a set of evidence of deletion sites an inherited deletion (parameter k_2 in our objective function)
True Positive	There is one interval that contains the inherited deletion, thus a true positive corresponds to correctly identifying an inherited deletion in this region.
False Positive	We have a false positive if we identify an inherited deletion in a region disjoint from the generated deletion's region.

Table B.0.1: Six tunable parameters and two scoring metrics for testing of the deletion inference algorithm.

Abbreviations

AGRE	Autism Genetic Resource Exchange
ASD	Autism spectrum disorders
ASE	allelic specific expression
ASM	allelic specific methylation
BFM	Boolean fragment mapping
chr	chromosome(s)
CNV/CNA	copy number variation/aberration
DELISHUS	algorithm for inferring **del**etions **i**n **s**hared **h**aplotypes **u**sing **S**NPs)
DFS	depth first search
DNA	deoxyribonucleic acid
EAGLE	Enhanced Artificial Genome Engine
FMPR	fragment mapping phase relationship
GATK	Genome Analysis ToolKit
GWAS	genome-wide association study
HC	HapCompass
HH	homozygous haplotype
HIV	human immunodeficiency virus
HMM	hidden Markov model
IBD	identity-by-descent or identical-by-descent
IBS	identity-by-state or identical-by-state
LD	linkage disequilibrium
LE	linkage equilibrium
LOH	loss of heterozygosity
MEC	minimum error correction

MER	minimum edge removal
MFR	minimum fragment removal
MSR	minimum SNP removal
MWER	minimum weighted edge removal
MWVR	minimum weighted vertex removal
P1/P2	HapMap Phase 1/2+3
PAC	Product of Approximate Conditionals
PCR	polymerase chain reaction
QC	quality control
RNA	ribonucleic acid
ROH	run of homozygosity
SE	switch error
SI	single individual algorithm (in context of inferring deletions)
sib TDT	sibling transmission disequilibrium test
SNP/SNV	single nucleotide polymorphism/variant
ST	spanning trees
SV	structural variant/variation
TDT	transmission disequilibrium test
VQR	viral quasispecies reconstruction

Table B.0.2: A list of abbreviations and acronyms used throughout the dissertation.

References

Aguiar, Derek and Sorin Istrail (2012). HAPCOMPASS: A fast cycle basis algorithm for accurate haplotype assembly of sequence data. *J. Comput. Biol.* 19 (6), 577–590.

Aguiar, Derek and Sorin Istrail (2013). Haplotype assembly in polyploid genomes and identical by descent shared tracts. *Bioinformatics* 29 (13). Appeared in the Proceedings of 21st Annual International Conference on Intelligent Systems for Molecular Biology, i352–i360.

Aguiar, Derek, Eric Morrow, and Sorin Istrail (2014). Tractatus: An Exact and Subquadratic Algorithm for Inferring Identical-by-Descent Multi-shared Haplotype Tracts. *Research in Computational Molecular Biology*. Ed. by Roded Sharan. Vol. 8394. Lecture Notes in Computer Science. Springer International Publishing, 1–17.

Aguiar, Derek, Wendy SW Wong, and Sorin Istrail (2014). Tumor haplotype assembly algorithms for cancer genomics. *Pacific Symposium on Biocomputing*. Vol. 19. World Scientific, 3–14.

Aguiar, Derek, Bjarni V. Halldorsson, Eric M. Morrow, and Sorin Istrail (2012). DELISHUS: an efficient and exact algorithm for genome-wide detection of deletion polymorphism in autism. *Bioinformatics* 28 (12). Appeared in the Proceedings of 20th Annual International Conference on Intelligent Systems for Molecular Biology, i154–i162.

Bafna, Vineet, Sorin Istrail, Giuseppe Lancia, and Romeo Rizzi (2005). Polynomial and APX-hard cases of the individual haplotyping problem. *Theoretical Computer Science* 335 (1), 109–125.

Bansal, Vikas and Vineet Bafna (2008). HapCUT: an efficient and accurate algorithm for the haplotype assembly problem. *Bioinformatics* 24 (16), i153–159.

Bansal, Vikas, Aaron L. Halpern, Nelson Axelrod, and Vineet Bafna (2008). An MCMC algorithm for haplotype assembly from whole-genome sequence data. *Genome Research* 18 (8), 1336–1346.

Boeva, Valentina, Andrei Zinovyev, Kevin Bleakley, Jean-Philippe Vert, Isabelle Janoueix-Lerosey, Olivier Delattre, and Emmanuel Barillot (2011). Control-free calling of copy number alterations in deep-sequencing data using GC-content normalization. *Bioinformatics (Oxford, England)* 27 (2), 268–9.

Bradnam, Keith, Joseph Fass, Anton Alexandrov, Paul Baranay, Michael Bechner, Inanc Birol, Sebastien Boisvert, Jarrod Chapman, Guillaume Chapuis, et al. (2013). Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species. *GigaScience* 2 (1), 10.

Broad Institute HapMap Pacific Biosciences Data (15 January 2013). `https://github.com/PacificBiosciences/DevNet/wiki/Datasets`.

Browning, B. L. and S. R. Browning (2009). A unified approach to genotype imputation and haplotype-phase inference for large data sets of trios and unrelated individuals. *American journal of human genetics* 84 (2), 210–223.

Browning, Brian L. and Sharon R. Browning (2011a). A fast, powerful method for detecting identity by descent. *American journal of human genetics* 88 (2), 173–182.

Browning, Sharon R. and Brian L. Browning (2011b). Haplotype phasing: existing methods and new developments. *Nat Rev Genet* 12 (10), 703–714.

Browning, Sharon R. and Brian L. Browning (2012). Identity by Descent Between Distant Relatives: Detection and Applications. *Annual Review of Genetics* 46 (1), 617–633.

Bruining, Hilgo et al. (2010). Dissecting the Clinical Heterogeneity of Autism Spectrum Disorders through Defined Genotypes. *PLoS ONE* 5 (5), e10887.

Cazals, F. and C. Karande (2008). A note on the problem of reporting maximal cliques. *Theoretical Computer Science* 407 (1-3), 564–568.

Chen, Z. Jeffrey and Zhongfu Ni (2006). Mechanisms of genomic rearrangements and gene expression changes in plant polyploids. *BioEssays* 28 (3), 240–252.

Ching, Michael S.L. et al. (2010). Deletions of NRXN1 (neurexin-1) predispose to a wide spectrum of developmental disorders. *American Journal of Medical Genetics Part B: Neuropsychiatric Genetics* 153B (4), 937–947.

Cibulskis, Kristian, Aaron McKenna, Tim Fennell, Eric Banks, Mark DePristo, and Gad Getz (2011). ContEst: estimating cross-contamination of human samples in next-generation sequencing data. *Bioinformatics (Oxford, England)* 27 (18), 2601–2.

Clark, AG (1990). Inference of haplotypes from PCR-amplified samples of diploid populations. *Mol Biol Evol* 7 (2), 111–122.

Conrad, Donald F., T. Daniel Andrews, Nigel P. Carter, Matthew E. Hurles, and Jonathan K. Pritchard (2006). A high-resolution survey of deletion polymorphism in the human genome. *Nature genetics* 38 (1), 75–81.

Conrad, Donald F. et al. (2009). Origins and functional impact of copy number variation in the human genome. *Nature* 464 (7289), 704–712.

Consortium, The International Multiple Sclerosis Genetics (2007). Risk Alleles for Multiple Sclerosis Identified by a Genomewide Study. *N Engl J Med* 357 (9), 851–862.

Corona, Erik et al. (2007). Identification of Deletion Polymorphisms from Haplotypes. *Research in Computational Molecular Biology.* Lecture Notes in Computer Science 4453. Ed. by Terry Speed and Haiyan Huang, 354–365.

Delaneau, Olivier, Jonathan Marchini, and Jean-Francois Zagury (2011). A linear complexity phasing method for thousands of genomes. *Nat Meth* 9 (2), 179–181.

Deo, Narsingh, G. Prabhu, and M. S. Krishnamoorthy (1982). Algorithms for Generating Fundamental Cycles in a Graph. *ACM Trans. Math. Softw.* 8 (1), 26–42.

DePristo, Mark A., Eric Banks, Ryan Poplin, Kiran V. Garimella, Jared R. Maguire, Christopher Hartl, Anthony A. Philippakis, Guillermo del Angel, Manuel A. Rivas, et al. (2011). A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nat Genet* 43 (5), 491–498.

Ding, Li, Matthew J Ellis, et al. (2010). Genome remodelling in a basal-like breast cancer metastasis and xenograft. *Nature* 464 (7291), 999–1005.

Fang, Fang, Emily Hodges, et al. (2012). Genomic landscape of human allele-specific DNA methylation. *Proceedings of the National Academy of Sciences* 109 (19), 7332–7337.

Farach, M. (1997). Optimal suffix tree construction with large alphabets. *Proceedings of the 38th Annual Symposium on Foundations of Computer Science.* FOCS '97. Washington, DC, USA: IEEE Computer Society, 137–143.

Fearnhead, Paul and Peter Donnelly (2001). Estimating Recombination Rates From Population Genetic Data. *Genetics* 159 (3), 1299–1318.

Fiegler, H. et al. (2006). High resolution array-CGH analysis of single cells. *Nucleic Acid Research* 35, 1–10.

Fischbach, Gerald D. and Catherine Lord (2010). The Simons Simplex Collection: A Resource for Identification of Autism Genetic Risk Factors. *Neuron* 68 (2), 192–195.

Gamsiz, Ece D, Emma W Viscidi, Abbie M Frederick, Shailender Nagpal, Stephan J Sanders, Michael T Murtha, Michael Schmidt, Elizabeth W Triche, Daniel H Geschwind, et al. (2013). Intellectual Disability Is Associated with Increased Runs of Homozygosity in Simplex Autism. *The American Journal of Human Genetics* 93 (1), 103–109.

Gao, Chuan, Karthik Devarajan, Yan Zhou, Carolyn M Slater, Mary B Daly, and Xiaowei Chen (2012). Identifying breast cancer risk loci by global differential allele-specific expression (DASE) analysis in mammary epithelial transcriptome. *BMC genomics* 13 (1), 570.

Genome in a Bottle Consortium (2013). *NIST NA12878 Highly Confident integrated genotype.*

Gentric, G., S. Celton-Morizur, and C. Desdouets (2012). Polyploidy and liver proliferation. *Clinics and Research in Hepatology and Gastroenterology* 36 (1), 29–34.

Geraci, Filippo (2010). A comparison of several algorithms for the single individual SNP haplotyping reconstruction problem. *Bioinformatics (Oxford, England)* 26 (18), 2217–2225.

Glessner, Joseph T. et al. (2009). Autism genome-wide copy number variation reveals ubiquitin and neuronal genes. *Nature* 459 (7246), 569–573.

Greaves, Mel and Carlo C Maley (2012). Clonal evolution in cancer. *Nature* 481 (7381), 306–13.

Greenman, Christopher, Philip Stephens, Raffaella Smith, Gillian L Dalgliesh, Christopher Hunter, Graham Bignell, Helen Davies, Jon Teague, Adam Butler, et al. (2009). Patterns of somatic mutation in human cancer genomes. *Nature* 446 (7132), 153–158.

Gudbjartsson, Daniel F., G. Bragi Walters, Gudmar Thorleifsson, Hreinn Stefansson, Bjarni V. Halldorsson, Pasha Zusmanovich, Patrick Sulem, Steinunn Thorlacius, Arnaldur Gylfason, et al. (2008). Many sequence variants affecting diversity of adult human height. *Nat Genet* 40 (5), 609–615.

Guilmatre, Audrey et al. (2009). Recurrent Rearrangements in Synaptic and Neurodevelopmental Genes and Shared Biologic Pathways in Schizophrenia, Autism, and Mental Retardation. *Arch Gen Psychiatry* 66 (9), 947–956.

Gusev, Alexander, Jennifer K. Lowe, Markus Stoffel, Mark J. Daly, David Altshuler, Jan L. Breslow, Jeffrey M. Friedman, and Itsik Pe'er (2009). Whole population, genome-wide mapping of hidden relatedness. *Genome Res.* 19 (2), 318–326.

Gusev, Alexander, Eimear E. Kenny, Jennifer K. Lowe, Jaqueline Salit, Richa Saxena, Sekar Kathiresan, David M. Altshuler, Jeffrey M. Friedman, Jan L. Breslow, et al. (2011). DASH: A Method for Identical-by-Descent Haplotype Mapping Uncovers Association with Recent Variation. *Am J Hum Genet* 88 (6), 706–717.

Gusnanto, Arief, Henry M Wood, Yudi Pawitan, Pamela Rabbitts, and Stefano Berri (2012). Correcting for cancer genome size and tumour cell content enables better estimation of copy number alterations from next-generation sequence data. *Bioinformatics (Oxford, England)* 28 (1), 40–7.

Hague, Stephen et al. (2003). Early-onset Parkinson's disease caused by a compound heterozygous DJ-1 mutation. *Annals of Neurology* 54 (2), 271–274.

Halldorsson, Bjarni V., Derek Aguiar, and Sorin Istrail (2011). Haplotype Phasing by Multi-Assembly of Shared Haplotypes: Phase-Dependent Interactions Between Rare Variants. *Proceedings of the Pacific Symposium on Biocomputing*, 88–99.

Halldórsson, Bjarni V. and D.F. Gudbjartsson (2011). An algorithm for detecting high frequency copy number polymorphisms using SNP arrays. *Journal of Compuational Biology* 18, 955–966.

Halldórsson, Bjarni V., Vineet Bafna, Nathan Edwards, Ross Lippert, Shibu Yooseph, and Sorin Istrail (2004). A Survey of Computational Methods for Determining Haplotypes. *Computational Methods for SNPs and Haplotype Inference*. Ed. by Sorin Istrail, Michael Waterman, and Andrew Clark. Vol. 2983. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 26–47.

Halldórsson, Bjarni V., Derek Aguiar, Ryan Tarpine, and Sorin Istrail (2010). The Clark Phase-able Sample Size Problem: Long-Range Phasing and Loss of Heterozygosity in GWAS. *Research in Computational Molecular Biology*. Ed. by Bonnie Berger. Vol. 6044. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 158–173.

Halldórsson, Bjarni V., Derek Aguiar, Ryan Tarpine, and Sorin Istrail (2011). The Clark Phaseable Sample Size Problem: Long-Range Phasing and Loss of Heterozygosity in GWAS. *Journal of Computational Biology* 18 (3), 323–333.

Harley, ER. (2004). Comparison of Clique-Listing Algorithms. *Proceedings of the International Conference on Modeling, Simulation and Visualization Methods (MSV'04)*, 433–438.

He, Dan (2013). IBD-Groupon: an efficient method for detecting group-wise identity-by-descent regions simultaneously in multiple individuals based on pairwise IBD relationships. *Bioinformatics* 29 (13), 162–170.

He, Dan, Arthur Choi, Knot Pipatsrisawat, Adnan Darwiche, and Eleazar Eskin (2010). Optimal algorithms for haplotype assembly from whole-genome sequence data. *Bioinformatics* 26 (12), i183–i190.

Hill, W.G. and Alan Robertson (1968). Linkage disequilibrium in finite populations. English. *Theoretical and Applied Genetics* 38 (6), 226–231.

Howie, Bryan N., Peter Donnelly, and Jonathan Marchini (2009). A Flexible and Accurate Genotype Imputation Method for the Next Generation of Genome-Wide Association Studies. *PLoS Genet* 5 (6), e1000529.

Hudson, Richard R. (1991). Gene genealogies and the coalescent process. *Oxford Survey in Evolutionary Biology* 7. Ed. by D. Futuyama and J. Antonovics, 1–44.

Hudson, Richard R. (2002). Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics* 18 (2), 337–338.

Iafrate, A.J. et al. (2004). Detection of large-scale variation in the human genome. *Nature Genetics* 36, 949–951.

Illumina Inc. (2013). *BaseSpace G.C.C.* https://basespace.illumina.com/home/index.

International HapMap Consortium (2003). The International HapMap Project. *Nature* 426 (6968), 789–796.

KA., Wetterstrand (2009). *DNA Sequencing Costs: Data from the NHGRI Genome Sequencing Program (GSP).* Accessed 12/01/2013. URL: www.genome.gov/sequencingcosts.

Kawarabayashi, Kenichi, Yusuke Kobayashi, and Bruce Reed (2012). The disjoint paths problem in quadratic time. *Journal of Combinatorial Theory, Series B* 102 (2), 424–435.

Khaja, R. et al. (2006). Genome assembly comparison identifies structural variants in the human genome. *Nature Genetics* 38, 1413–1418.

Kingman, J. F. C. (1982). On the Genealogy of Large Populations. *Journal of Applied Probability* 19, 27–43.

Knudson, Alfred G (1971). Muation and Cancer: Statistical Study of Retinoblastoma. *Proceedings of the National Academy of Sciences of the United States of America* 68 (4), 820–823.

Koboldt, Daniel C, Qunyuan Zhang, David E Larson, Dong Shen, Michael D McLellan, Ling Lin, Christopher a Miller, Elaine R Mardis, Li Ding, et al. (2012). VarScan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing. *Genome research* 22 (3), 568–76.

Koboldt, Daniel C., Karyn M. Steinberg, David E. Larson, Richard K. Wilson, and Elaine R. Mardis (2013). The Next-Generation Sequencing Revolution and Its Impact on Genomics. *Cell* 155 (1), 27–38.

Kong, Augustine, Gisli Masson, Michael L. Frigge, Arnaldur Gylfason, Pasha Zusmanovich, Gudmar Thorleifsson, Pall I. Olason, Andres Ingason, Stacy Steinberg, et al. (2008). Detection of sharing by descent, long-range phasing and haplotype imputation. *Nat Genet* 40 (9), 1068–1075.

Krawitz, Peter M., Michal R. Schweiger, Christian Rodelsperger, Carlo Marcelis, Uwe Kolsch, Christian Meisel, Friederike Stephani, Taroh Kinoshita, Yoshiko Murakami, et al. (2010). Identity-by-descent filtering of exome sequence data identifies PIGV mutations in hyperphosphatasia mental retardation syndrome. *Nature Genetics* 42 (10), 827–829.

Lam, Fumei, Ryan Tarpine, and Sorin Istrail (2011). Conservative Extensions of Linkage Disequilibrium Measures from Pairwise to Multi-loci and Algorithms for Optimal Tagging SNP Selection. *Research in Computational Molecular Biology.* Ed. by Vineet Bafna and S. Sahinalp. Vol. 6577. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 468–482.

Lancia, Giuseppe, Vineet Bafna, Sorin Istrail, Ross Lippert, and Russell Schwartz (2001). SNPs Problems, Complexity, and Algorithms. *ESA '01: Proceedings of the 9^{th} Annual European Symposium on Algorithms.* London, UK: Springer-Verlag, 182–193.

Lee, William, Zhaoshi Jiang, et al. (2010). The mutation spectrum revealed by paired genome sequences from a lung cancer patient. *Nature* 465 (7297), 473–477.

Leitch, A. R. and I. J. Leitch (2008). Genomic Plasticity and the Diversity of Polyploid Plants. *Science* 320 (5875), 481–483.

Levy, Samuel, Granger Sutton, Pauline C. Ng, Lars Feuk, Aaron L. Halpern, Brian P. Walenz, Nelson Axelrod, Jiaqi Huang, Ewen F. Kirkness, et al. (2007). The diploid genome sequence of an individual human. *PLoS biology* 5 (10), e254.

Lewontin, RC (1988). On measures of gametic disequilibrium. *Genetics* 120 (3), 849–852.

Ley, Timothy J., Elaine R. Mardis, et al. (2008). DNA sequencing of a cytogenetically normal acute myeloid leukaemia genome. *Nature* 456 (7218), 66–72.

Li, Heng, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, Richard Durbin, et al. (2009). The Sequence Alignment/Map format and SAMtools. *Bioinformatics* 25 (16), 2078–2079.

Li, Na and Matthew Stephens (2003). Modeling Linkage Disequilibrium and Identifying Recombination Hotspots Using Single-Nucleotide Polymorphism Data. *Genetics* 165 (4), 2213–2233.

Li, Yun, Cristen J Willer, Jun Ding, Paul Scheet, and Gonçalo R Abecasis (2010). MaCH: using sequence and genotype data to estimate haplotypes and unobserved genotypes. *Genetic epidemiology* 34 (8), 816–834.

Li, Zhen-ping, Ling-yun Wu, Yu-ying Zhao, and Xiang-sun Zhang (2006). A Dynamic Programming Algorithm for the k-Haplotyping Problem. *Acta Mathematicae Applicatae Sinica (English Series)* 22 (3), 405–412.

Lin, Pei-Chun, Eugenia G Giannopoulou, et al. (2013). Epigenomic Alterations in Localized and Advanced Prostate Cancer. *Neoplasia* 15 (4), 373–383.

Lin, S., D. J. Cutler, M. E. Zwick, and A. Chakravarti (2002). Haplotype inference in random population samples. *Am. J. Hum. Genet.* 71 (5), 1129–1137.

Lippert, R., R. Schwartz, G. Lancia, and S. Istrail (2002). Algorithmic strategies for the single nucleotide polymorphism haplotype assembly problem. *Brief Bioinform* 3 (1), 23–31.

Lodish, Harvey (2008). *Molecular cell biology.* Macmillan.

Marchini, J., D. Cutler, N. Patterson, M. Stephens, E. Eskin, E. Halperin, S. Lin, Z. S. Qin, H. M. Munro, et al. (2006). A comparison of phasing algorithms for trios and unrelated individuals. *American journal of human genetics* 78 (3), 437–450.

Marchini, Jonathan and Bryan Howie (2010). Genotype imputation for genome-wide association studies. *Nat Rev Genet* 11 (7), 499–511.

Mardis, Elaine R. (2012). Genome sequencing and cancer. *Curr. Opin. Genet. Dev.* 22 (3), 245–250.

Mardis, Elaine R (2013). Next-generation sequencing platforms. *Annual review of analytical chemistry* 6, 287–303.

McCarroll, Steven A., Tracy N. Hadnott, George H. Perry, Pardis C. Sabeti, Michael C. Zody, Jeffrey C. Barrett, Stephanie Dallaire, Stacey B. Gabriel, Charles Lee, et al. (2005). Common deletion polymorphisms in the human genome. *Nature Genetics* 38 (1), 86–92.

McClellan, Jon and Mary-Claire King (2010). Genetic Heterogeneity in Human Disease. *Cell* 141 (2), 210–217.

McCreight, Edward M. (1976). A Space-Economical Suffix Tree Construction Algorithm. *J. ACM* 23 (2), 262–272.

McKenna, Aaron, Matthew Hanna, Eric Banks, Andrey Sivachenko, Kristian Cibulskis, Andrew Kernytsky, Kiran Garimella, David Altshuler, Stacey Gabriel, et al. (2010). The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res.* 20 (9), 1297–1303.

Medvedev, Paul, Monica Stanciu, and Michael Brudno (2009). Computational methods for discovering structural variation with next-generation sequencing. *Nat Meth* 6 (11s), S13–S20.

Mefford, Heather C and Evan E Eichler (2009). Duplication hotspots, rare genomic disorders, and common disease. *Current Opinion in Genetics and Development* 19 (3), 196–204.

Meyerson, Matthew, Stacey Gabriel, and Gad Getz (2010). Advances in understanding cancer genomes through second-generation sequencing. *Nature Reviews Genetics* 11 (10), 685–696.

Mills, Ryan E, Christopher T Luttig, Christine E Larkins, Adam Beauchamp, Circe Tsui, W Stephen Pittard, and Scott E Devine (2006). An initial map of insertion and deletion (INDEL) variation in the human genome. *Genome Research* 16, 1182–1190.

Mills, Ryan E, Klaudia Walter, Chip Stewart, Robert E Handsaker, Ken Chen, Can Alkan, Alexej Abyzov, Seungtai Chris Yoon, Kai Ye, et al. (2011). Mapping copy number variation by population-scale genome sequencing. *Nature* 470 (7332), 59–65.

Minichiello, Mark J. and Richard Durbin (2006). Mapping Trait Loci by Use of Inferred Ancestral Recombination Graphs. *The American Journal of Human Genetics* 79 (5), 910–922.

Miyazawa, H., M. Kato, T. Awata, M. Kohda, H. Iwasa, N. Koyama, T. Tanaka, N. Huqu, S. Kyo, et al. (2007). Homozygosity Haplotype Allows a Genomewide Search for the Autosomal Segments Shared among Patients. *The American Journal of Human Genetics* 80 (6), 1090–1102.

Morrow, Eric M. (2010). Genomic Copy Number Variation in Disorders of Cognitive Development. *Journal of the American Academy of Child and Adolescent Psychiatry* 49 (11), 1091–1104.

Morrow, Eric M, Seung-Yun Yoo, Steven W Flavell, Tae-Kyung Kim, Yingxi Lin, Robert Sean Hill, Nahit M Mukaddes, Soher Balkhy, Generoso Gascon, et al. (2008). Identifying Autism Loci and Genes by Tracing Recent Shared Ancestry. *Science* 321 (5886), 218–223.

Mousavi, Sayyed R., Maryam Mirabolghasemi, Nadia Bargesteh, and Majid Talebi (2011). Effective haplotype assembly via maximum Boolean satisfiability. *Biochemical and biophysical research communications* 404 (2), 593–598.

O'Roak, Brian J, Pelagia Deriziotis, Choli Lee, Laura Vives, Jerrod J Schwartz, Santhosh Girirajan, Emre Karakoc, Alexandra P MacKenzie, Sarah B Ng, et al. (2011). Exome sequencing in sporadic autism spectrum disorders identifies severe de novo mutations. *Nature Genetics* 43 (6), 585–589.

Panconesi, Alessandro and Mauro Sozio (2004). Fast Hare: A Fast Heuristic for Single Individual SNP Haplotype Reconstruction. *Proceedings of the 4^{th} International Workshop on Algorithms in Bioinformatics WABI04*. Vol. 3240, 266–277.

Park, Hansoo, Jong-Il Kim, Young Seok Ju, Omer Gokcumen, Ryan E Mills, Sheehyun Kim, Seungbok Lee, Dongwhan Suh, Dongwan Hong, et al. (2010). Discovery of common Asian copy number variants using integrated high-resolution array CGH and massively parallel DNA sequencing. *Nature Genetics* 42 (400-405).

Pierson, Tyler Mark, Dimitre R. Simeonov, Murat Sincan, David A. Adams, Thomas Markello, Gretchen Golas, Karin Fuentes-Fajardo, Nancy F. Hansen, Praveen F. Cherukuri, et al. (2012). Exome sequencing and SNP analysis detect novel compound heterozygosity in fatty acid hydroxylase-associated neurodegeneration. *Eur J Hum Genet* 20 (4), 476–479.

Pleasance, Erin D, R Keira Cheetham, Philip J Stephens, David J McBride, Sean J Humphray, Chris D Greenman, Ignacio Varela, Meng-Lay Lin, Gonzalo R Ordóñez, et al. (2010). A comprehensive catalogue of somatic mutations from a human cancer genome. *Nature* 463 (7278), 191–196.

Purcell, Shaun, Benjamin Neale, Kathe Todd-Brown, Lori Thomas, Manuel A. R. Ferreira, David Bender, Julian Maller, Pamela Sklar, Paul I. W. de Bakker, et al. (2007). PLINK: A Tool Set for Whole-Genome Association and Population-Based Linkage Analyses. *The American Journal of Human Genetics* 81 (3), 559–575.

Rivadeneira, F., U. Styrkarsdottir, K. Estrada, B. Halldorsson, Y. Hsu, J. B. Richards, M. C. Zillikens, F. Kavvoura, N. Amin, et al. (2009). Bone. Vol. 44. Elsevier Science. Chap. Twenty loci associated with bone mineral density identified by large-scale meta-analysis of genome-wide association datasets, S230–S231.

Rizzi, Romeo, Vineet Bafna, Sorin Istrail, and Giuseppe Lancia (2002). Practical Algorithms and Fixed-Parameter Tractability for the Single Individual SNP Haplotyping Problem. *Proceedings* *of the Second International Workshop on Algorithms in Bioinformatics*. WABI '02. London, UK, UK: Springer-Verlag, 29–43.

Robertson, N. and P.D. Seymour (1995). Graph Minors .XIII. The Disjoint Paths Problem. *Journal of Combinatorial Theory, Series B* 63 (1), 65–110.

Sanders, Stephan J, A Gulhan Ercan-Sencicek, Vanessa Hus, Rui Luo, Michael T Murtha, Daniel Moreno-De-Luca, Su H Chu, Michael P Moreau, Abha R Gupta, et al. (2011). Multiple recurrent de novo CNVs, including duplications of the 7q11.23 Williams syndrome region, are strongly associated with autism. *Neuron* 70 (5), 863–885.

Saunders, Christopher T, Wendy S W Wong, Sajani Swamy, Jennifer Becq, Lisa J Murray, and R Keira Cheetham (2012). Strelka: accurate somatic small-variant calling from sequenced tumor-normal sample pairs. *Bioinformatics (Oxford, England)* 28 (14), 1811–7.

Scheet, Paul and Matthew Stephens (2006). A Fast and Flexible Statistical Model for Large-Scale Population Genotype Data: Applications to Inferring Missing Genotypes and Haplotypic Phase. *The American Journal of Human Genetics* 78 (4), 629–644.

Schwartz, Russell (2010). Theory and Algorithms for the Haplotype Assembly Problem. *Commun. Inf. Syst.* 10 (1), 23–38.

Sebat, Jonathan, B Lakshmi, Dheeraj Malhotra, Jennifer Troge, Christa Lese-Martin, Tom Walsh, Boris Yamrom, Seungtai Yoon, Alex Krasnitz, et al. (2007). Strong Association of De Novo Copy Number Mutations with Autism. *Science* 316 (5823), 445–449.

Sharan, Roded, Bjarni V. Halldórsson, and Sorin Istrail (2006). Islands of Tractability for Parsimony Haplotyping. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 3 (3), 303–311.

Siva, Nayanah (2008). 1000 Genomes project. *Nature biotechnology* 26 (3), 256.

Spielman, Richard S and Warren J Ewens (1996). The TDT and other family-based tests for linkage disequilibrium and association. *American journal of human genetics* 59 (5), 983.

Spielman, Richard S and Warren J Ewens (1998). A sibship test for linkage in the presence of association: the sib transmission/disequilibrium test. *The American Journal of Human Genetics* 62 (2), 450–458.

Spielman, Richard S, Ralph E McGinnis, and Warren J Ewens (1993). Transmission test for linkage disequilibrium: the insulin gene region and insulin-dependent diabetes mellitus (IDDM). *American journal of human genetics* 52 (3), 506.

Stefansson, Hreinn, Dan Rujescu, Sven Cichon, Olli PH Pietiläinen, Andres Ingason, Stacy Steinberg, Ragnheidur Fossdal, Engilbert Sigurdsson, Thordur Sigmundsson, et al. (2008). Large recurrent microdeletions associated with schizophrenia. *Nature* 455 (7210), 232–236.

Stephens, Matthew, Nicholas J. Smith, and Peter Donnelly (2001). A New Statistical Method for Haplotype Reconstruction from Population Data. *The American Journal of Human Genetics* 68 (4), 978–989.

Styrkarsdottir, Unnur, Bjarni V. Halldorsson, Solveig Gretarsdottir, Daniel F. Gudbjartsson, G. Bragi Walters, Thorvaldur Ingvarsson, Thorbjorg Jonsdottir, Jona Saemundsdottir, Jacqueline R. Center, et al. (2008). Multiple Genetic Loci for Bone Mineral Density and Fractures. *N Engl J Med* 358 (22), 2355–2365.

Tarpine, Ryan, Fumei Lam, and Sorin Istrail (2011). Conservative extensions of linkage disequilibrium measures from pairwise to multi-loci and algorithms for optimal tagging SNP selection. *Proceedings of the 15th Annual International Conference on Research in Computational Molecular Biology*. RECOMB'11. Vancouver, BC, Canada: Springer-Verlag, 468–482.

Tewhey, Ryan, Vikas Bansal, Ali Torkamani, Eric J. Topol, and Nicholas J. Schork (2011). The importance of phase information for human genomics. *Nature Reviews Genetics* 12 (3), 215–223.

The 1000 Genomes Project Consortium (2010). A map of human genome variation from population-scale sequencing. *Nature* 467 (7319), 1061–1073.

The Cancer Genome Atlas (2012). Comprehensive molecular portraits of human breast tumours. *Nature*, 1–10.

Thorisson, Gudmundur A, Albert V Smith, Lalitha Krishnan, and Lincoln D Stein (2005). The international HapMap project web site. *Genome research* 15 (11), 1592–1593.

Tsukiyama, Shuji, Mikio Ide, Hiromu Ariyoshi, and Isao Shirakawa (1977). A New Algorithm for Generating All the Maximal Independent Sets. *SIAM Journal on Computing* 6 (3), 505–517.

Tuch, Brian B, Rebecca R Laborde, Xing Xu, Jian Gu, Christina B Chung, Cinna K Monighetti, Sarah J Stanley, Kerry D Olsen, Jan L Kasperbauer, et al. (2010). Tumor transcriptome sequencing reveals allelic expression imbalances associated with copy number alterations. *PloS one* 5 (2), e9317.

Turro, Ernest, Shu-Yi Su, Angela Goncalves, Lachlan Coin, Sylvia Richardson, and Alex Lewin (2011). Haplotype and isoform specific expression estimation using multi-mapping RNA-seq reads. *Genome Biology* 12 (2), R13.

Ukkonen, E. (1995). On-line construction of suffix trees. English. *Algorithmica* 14 (3), 249–260.

Valle, Laura, Tarsicio Serena-acedo, Sandya Liyanarachchi, Heather Hampel, Zhongyuan Li, Qinghua Zeng, Hong-tao Zhang, Michael J Pennison, Maureen Sadim, et al. (2009). Germline Allele-Specific Expression of TGFBR1 Confers an Increased Risk of Colorectal Cancer. *Science* 321 (5894), 1361–1365.

Venter, J. Craig, Mark D. Adams, Eugene W. Myers, Peter W. Li, Richard J. Mural, Granger G. Sutton, Hamilton O. Smith, Mark Yandell, Cheryl A. Evans, et al. (2001). The Sequence of the Human Genome. *Science* 291 (5507), 1304–1351.

Walsh, Christopher A., Eric M. Morrow, and John L.R. Rubenstein (2008). Autism and Brain Development. *Cell* 135 (3), 396–400.

Wang, Jianmin, Charles G Mullighan, John Easton, Stefan Roberts, Sue L Heatley, Jing Ma, Michael C Rusch, Ken Chen, Christopher C Harris, et al. (2011). CREST maps somatic structural variation in cancer genomes with base-pair resolution. *Nature Methods* 8 (8), 652–656.

Wang, Kai, Mingyao Li, Dexter Hadley, Rui Liu, Joseph Glessner, Struan FA Grant, Hakon Hakonarson, and Maja Bucan (2007). PennCNV: an integrated hidden Markov model designed for high-resolution copy number variation detection in whole-genome SNP genotyping data. *Genome research* 17 (11), 1665–1674.

Wang, Rui-Sheng, Ling-Yun Wu, Zhen-Ping Li, and Xiang-Sun Zhang (2005). Haplotype reconstruction from SNP fragments by minimum error correction. *Bioinformatics* 21 (10), 2456–2462.

Weiss, Lauren A, Yiping Shen, Joshua M Korn, Dan E Arking, David T Miller, Ragnheidur Fossdal, Evald Saemundsen, Hreinn Stefansson, Manuel AR Ferreira, et al. (2008). Association between microdeletion and microduplication at 16p11.2 and autism. *New England Journal of Medicine* 358, 667–675.

Zerr, Troy, Gregory M Cooper, Evan E Eichler, and Deborah A Nickerson (2010). Targeted interrogation of copy number variation using SCIMMkit. *Bioinformatics* 26 (1), 120–122.

Zhao, Yu-Ying, Ling-Yun Wu, Ji-Hong Zhang, Rui-Sheng Wang, and Xiang-Sun Zhang (2005). Haplotype assembly from aligned weighted SNP fragments. *Computational Biology and Chemistry* 29 (4), 281–287.

Index