On Humans, Plants and Disease: Algorithmic Strategies for Haplotype Assembly Problems

Sorin Istrail

Department of Computer Science
Brown University
HapCompass: A new Haplotype Assembly Algorithm Framework

Derek Aguiar
Haplotype Assembly


“Improving data quality is crucial, because if a human genome cannot be independently assembled then the sequence data cannot be sorted into the two sets of parental chromosomes, or haplotypes. This process haplotype phasing will become one of the most useful tools in genomic medicine. Establishing the complete set of genetic information that we received from each parent is crucial to understanding the links between heritability, gene function, regulatory sequences and our predisposition to disease.”

Craig Venter, “Multiple personal genomes await,” Nature, April 2010
Genome to phenotype
Ploidy

Haploid
5.4 thousand bases

Diploid
(2.5 billion bases) * 2

Polyploid
(240 million bases) * 10

Cancer
Zink et al. 2004
• Every living organism has a collection of DNA or RNA molecules we refer to as its genome

• The term genome was credited to Hans Winkler in 1920 as a combination of the terms gene and chromosome
• The genome is organized into molecule structures called chromosomes
mathematical modeling

- DNA is double stranded and chromosomes are paired (e.g. in human) so our model could contain all of this information

\[
\begin{align*}
\text{Chr 1A} & : & \text{ACGTGTCATCGACTGAA} & \text{sense strand} \\
& & \text{TGCACAGTAGCTGACTT} & \text{antisense strand} \\
\text{Chr 1B} & : & \text{ACGTGTCATCGACTGAA} & \text{sense strand} \\
& & \text{TGCACAGTAGCTGACTT} & \text{antisense strand}
\end{align*}
\]
However, given a sense (antisense) strand, we can completely determine the antisense (sense) strand.
mathematical modeling

• And the two chromosomes are largely the same, so we can remove one of the chromosome

Chr 1A  ACGTGTCATCGACTGAA

Chr 1B  ACGTGTCATCGACTGAA
genome sequencing

genome assembly

read alignment/variant calling

haplotype assembly

downstream haplotype analysis
genome assembly, coverage = 1

Minimum Contig Length: 3.
Maximum Contig Length: 641.
Gaps Between Contigs: {81, 27, 37, 149, 21, 42, 143, 139, 31, 23, 13, 139, 188, 218, 18, 163, 69, 108, 87}
Number of Contigs: 20
Lander-Waterman Estimated Average Number of Contigs: 18.0127
Average Contig Length: 161.95
Lander-Waterman Estimated Average Contig Length: 189.011
Percent of Genome Covered by Contigs: 0.601374
Lander-Waterman Estimated Average Percent of Genome Covered by Contigs: 0.632121
Minimum Contig Length: 324.
Maximum Contig Length: 4973.
Gaps Between Contigs: 82
Number of Contigs: 2
Lander-Waterman Estimated Average Number of Contigs: 1.64957
Average Contig Length: 2649.5
Lander-Waterman Estimated Average Contig Length: 3243.09
• During cell replication, mistakes can be made in copying the DNA

• Types of variation include

• Single Nucleotide Polymorphisms

• Insertions and deletions (indels)

[ACGTGTCATCGACTGAA]

[ACGTGT---ACTGAA]

variation in populations

• Common variant, $\text{MAF} \geq 0.03$

• Rare variant, $0.001 < \text{MAF} < 0.03$

• Personal or de-novo variant $< 0.001$

• Variants can either be homozygous, heterozygous, hemizygous
haplotypes

• We call an alternative form of a variant an **allele**.
• A **haplotype** is an ordered set of alleles.
• Represents the differences between individuals in a population.

ACGTGTCATCGACTGAAAGGGGTCCGATATTTTCGAC

ACGTGTCATCGACTGGAAGGGGTCCGATATTTTTGAC
haplotype assembly

• In genome assembly we reconstructed the chromosome sequences, in haplotype assembly we will reconstruct the haplotypes.

• Informally, we obtain millions of haploid sequence reads from a single diploid organism and want to reconstruct the two distinct haplotypes for the individual.
The haplotype assembly workflow

1. DNA sequencing
2. Read alignment; retain SNPs
3. Computational modeling of reads and sequencing errors
4. Problem optimization
5. Haplotypes
The haplotype assembly workflow

1. DNA sequencing
2. Read alignment; retain SNPs
3. Computational modeling of reads and sequencing errors
4. Problem optimization
5. Haplotype
DNA sequencing

• Why is it important to understand the experimental techniques?

• Statistical modeling of sequencing errors to incorporate into our algorithms
DNA sequencing

• Types of sequencing platforms
• Sanger sequencing
• Next-generation or high-throughput
  • 454, Illumina, or SOLiD
• Single molecule or third-generation
  • Pacific Biosciences Real Time Sequencing
• DNA is fragmented and adapters are attached
454 and PacBio

• 454 technology washes nucleotides (all of a certain type) over cells of single stranded DNA

• Light is emitted and captured

• PacBio tracks the actual DNA polymerase molecule

• When nucleotides are cleaved, the fluorescence attached to the nucleotide molecule is detected
other technologies

• Sanger
• Low throughput, low error rates, long reads (up to 1k bp)

• SOLiD
• High throughput, small reads (50-100bp), low error rates

bottom line on sequencing

- Illumina error rates are low even across short repeats and errors are mostly substitutions due to the DNA polymerase making mistakes.
- 454 error rates higher than Illumina but moderately low and are mostly indel errors.
- PacBio error rates are very high (~10%) and mostly substitution errors thought to be independent of position.
where can I find data?

- Currently the best source is the 1000 Genomes Project

- If we include only the sequence reads that successfully aligned to NA12878 and compressed the alignment files we have 488 gigabytes of data
1000 Genomes Project

• Goal: to catalog human variation of all types (95% variation at frequency >1%)

• Define the **haplotype structure** of the human genome

• Develop novel analyses and tools


• Carried out in pilot project and three phases
The haplotype assembly workflow

1. DNA sequencing
2. Read alignment; retain SNPs
3. Computational modeling of reads and sequencing errors
4. Problem optimization
5. Haplotypes
sequence read alignment

• The real problem is $1_i \ll L$ for $1 \leq i \leq n$

• What can we do to overcome this problem?
The haplotype assembly workflow

1. DNA sequencing
2. read alignment; retain SNPs
3. computational modeling of reads and sequencing errors
4. problem optimization
5. haplotypes
errors

• Two fundamental sources of errors

• Sequencing
  • A base was miscalled
  • An insertion or deletion was miscalled

• Read alignment
  • A read was aligned to the wrong location or the alignment itself is wrong
• We can explicitly model sequencing errors based on the technology used to generate the reads

• Also, we can use the phred quality scores
understanding the input

• Because genome assemblies produce consensus sequences
  
  A/C C/G A/G T/A

• There is no way to determine the haplotypes from the genotype
  
  A C A T
  C G G A

• But sequence reads (regardless of the technology) are sampled from haploid fragments
For now, we assume the organism we are sequencing has ploidy=2, or 2 sets of chromosomes.

Thus, each SNP can only have two alleles.

Our computational model of an aligned read is termed a fragment.

Given $n$ SNPs, the $i^{th}$ fragment $f_i$ is an $n$-dimensional vector with alphabet $\{0, 1, -\}$ where 0 and 1 represent the major and minor alleles and – represents a gap in coverage.
formal definition of input

- The input is a matrix $M$ with $m$ rows corresponding to $m$ fragments and $n$ columns corresponding to the $n$ SNPs.

**SNP-fragment matrix**

<table>
<thead>
<tr>
<th></th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$s_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$f_2$</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>1</td>
</tr>
</tbody>
</table>
The haplotype assembly workflow

1. DNA sequencing
2. Read alignment; retain SNPs
3. Computational modeling of reads and sequencing errors
4. Problem optimization
5. Haplotypes
imagine a world without errors

How would you assemble the fragments?
fragment conflict

• Let $f_{ik}$ be the allele or ‘-’ at position $k$ of fragment $i$, then two fragments conflict if

$$f_{ik} \neq f_{jk} \land f_{ik} \neq '-' \land f_{jk} \neq '-'$$

• Informally, two fragments $f_i$ and $f_j$ are in fragment conflict if they cover a common SNP and have different alleles at that site

So, how would you assemble the fragments?
• The fragment conflict graph is a representation of the fragments which share a SNP but differ in alleles.

• The fragment conflict graph $G_F(V_F, E_F)$ has a vertex for every fragment and an edge between fragments that conflict (where $d(f_i, f_j)$ is the number of fragment conflicts between $f_i$ and $f_j$):

$$E_F = \{(v_i, v_j) | v_i \in V_F, v_j \in V_F, d(\vec{f}_i, \vec{f}_j) > 0\}$$
fragment conflict graph

- Find two sets such that there are no fragment conflicts internal to each set (bipartition of $G_F$)
- The shores of a bipartition of $G_F$ give the haplotype

Haplotype A: 1 0 0
Haplotype B: 0 1 1
minimum fragment removal

Lancia et al., 2001

But, exist in real data, sequencing errors do. Hmmmmm.

• **Optimization**: Compute a subset of vertices $V_S$ in $V_F$ such that $G(V_F - V_S)$ induces a bipartite graph.

• Sometimes referred to as the bipartization problem and is NP-hard
minimum edge removal

• **Optimization**: Compute a subset of edges $E_S$ in $E_F$ such that $G(E_F - E_S)$ induces a bipartite graph.

• Also referred to as bipartization and NP-hard

• But what exactly does an edge removal in $G_F$ mean?

• More on this later…
SNP conflict graph

- $G_S(V_S,E_S)$ has a vertex for every SNP and an edge between SNPs that exhibit 3 or more haplotypes:
The most used optimization in the literature is the Minimum Error Correction (MEC)

- FastHare\textsuperscript{4}, HAPCUT\textsuperscript{1}, HASH\textsuperscript{2}, He et al.\textsuperscript{3}

- Compute a minimum set of error corrections in the fragments (rows) of $M$ such that $G_F$ is bipartite.


Spanning trees

Spanning trees

- Connected, undirected graph, composed of all vertices of G and a subset of edges
- There must exist only 1 path between and two nodes
- How do we compute one?
maximum spanning tree

- $s_1$ to $s_2$: 8
- $s_1$ to $s_3$: 4
- $s_2$ to $s_4$: 2
- $s_3$ to $s_4$: -3
- $s_5$ to $s_5$: 2
- $s_6$ to $s_7$: 1
- $s_2$ to $s_6$: 6
- $s_3$ to $s_5$: 3
- $s_3$ to $s_4$: 10
maximum spanning tree

- Kruskal or Prim’s algorithm
- $O(E \log(E))$ time
- For integer edge weights, can be solved in linear time
Overview of the HapCompass model

• Models input as a graph where SNPs are nodes, fragments provide support for phasings

• Haplotype phasings = spanning trees

• Optimizes Minimum Weighted Edge Removal formulation

• Takes arbitrary sequence data – no constraints on size of gaps, number of gaps, length of reads
Construction of the compass graph

• Each SNP’s pairwise phasing relationship defined by the fragments is represented on the edges of the compass graph on the right.

• The majority rule phasing is shown in red on the edges of the compass graph.
A compass graph of NA12878 in a single component of the 1000 Genomes Data
Key properties of compass graphs

Compass Graph

- non-spanning tree edge
- spanning tree edge
Key properties of compass graphs

(I) A non-zero weighted edge defines the phasing between its two SNPs which is given by the sign of its weight (i.e. majority rule phasing). If \( w(s_i, s_j) > 0 \) then the phasing is \( \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix} \). Else if \( w(s_i, s_j) < 0 \) then the phasing is \( \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \).
Key properties of compass graphs

(II) Unique pairwise phasings of edges can be extended to paths, that is, the phasing is transitive among the SNPs along a path. E.g. the path \((s_6, s_5, s_3)\) defines \((100, 011)\) the phasing.
Key properties of compass graphs

Compass Graph

(iii) A **non-conflicting cycle** contains an even number of negative edges and no 0-weight edges.

non-spanning tree edge  spanning tree edge
Key properties of compass graphs

(IV) A **conflicting cycle** is a simple cycle that contains either an odd number of negative weight edges or at least one 0-weight edge or both.
Key properties of compass graphs

(V) A compass graph has a unique phasing if and only if it has no conflicting cycles. We call such a compass graph happy.
Minimum weighted edge removal (MWER): Given a compass graph $G$, remove a set of edges of minimum weight such that $G$ is happy.
Basic Algorithm

- construct a maximum spanning tree cycle basis
- iterate the following steps until the compass graph is happy:
  1. select a random conflicting cycle
  2. remove the edge with the least amount of evidence
  3. rebuild the cycle basis
Key properties of compass graphs

Compass Graph

- construct a maximum spanning tree cycle basis

non-spanning tree edge  spanning tree edge
Key properties of compass graphs

Compass Graph

-2  

S1: 2  S2: -2  S3: 4  S4: -2  S5: 3  S6: -2

-3

• iterate the following steps until the compass graph is happy:

(1) select a random conflicting cycle

non-spanning tree edge  spanning tree edge
Key properties of compass graphs

Compass Graph

(1) remove the edge with the least amount of evidence

non-spanning tree edge  spanning tree edge
Key properties of compass graphs

Compass Graph

-2
-2
-2
4
2
3
-3
-2

(1) rebuild the cycle basis

non-spanning tree edge  spanning tree edge

s1, s2, s3, s4, s5, s6
Key properties of compass graphs

(VI) The edge removal \( \{s_2,s_3\} \) makes \( G_C \) happy. All spanning trees of the happy \( G_C \) correspond to the same phasing; one such spanning tree is shown.

Happy
Compass Graph

\[
\begin{array}{c}
S_1 & S_2 & S_3 & S_4 & S_5 & S_6 \\
2 & -2 & 4 & 3 & -2 & -2
\end{array}
\]

non-spanning tree edge  spanning tree edge
Phasings of compass graphs

Happy Compass Graph

Example spanning tree

<table>
<thead>
<tr>
<th>edges</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
<th>S6</th>
</tr>
</thead>
<tbody>
<tr>
<td>{S1,S3}</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>{S1,S2}</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>{S1,S6}</td>
<td>1</td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>{S3,S4}</td>
<td>1</td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>{S3,S5}</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

haplotype 1 1 1 0 1 0
set cover

• Given $m$ elements \{1, 2, 3, \ldots, m\} and a set $S$ of $n$ sets such that the union of all sets $s$ in $S$ contain all elements, determine the minimum number of sets that cover all elements

• We can also define the weighted version by including weights on sets
set cover example

(greedy algorithm is actually a $\ln(n)+1$ approximation)
Set cover local optimization

- construct a maximum spanning tree cycle basis
- iterate the following steps until the compass graph is happy:
  1. select an edge that is a member of the most (many) conflicting cycles
  2. formulate a minimum set cover: conflicting cycles are the elements, edges are the covering sets if they are within a conflicting cycle, their weight is the weight of the edge in the compass graph
  3. a minimum set cover in this graph corresponds to a set of edges of smallest weight whose removal resolves every conflicting cycle
  4. rebuild the cycle basis
Key properties of compass graphs

Compass Graph

conflicting cycles are the elements

edges are the covering sets if they are within a conflicting cycle their weight is the weight of the edge in the compass graph

non-spanning tree edge spanning tree edge
Set cover

Diagram showing sets S2, S3, and S4 with a C-shaped intersection among them, and another set S1 with a loop connection.
The haplotype assembly workflow

- DNA sequencing
- read alignment; retain SNPs
- computational modeling of reads and sequencing errors
- problem optimization
- haplotypes
evaluation

• 5 main metrics for evaluating accuracy of haplotypes
  • Minimum error correction
  • Haplotype reconstruction rate
  • Haplotype switch error
  • Fragments mapping phase relationship
  • Boolean fragments mapped
minimum error correction

• Counts the number of bit flips in the reads to construct the haplotypes

• The distance \( d(r,h) \) between a read \( r \) and a haplotype \( h \) is just the Hamming distance (number of positions they differ)

• Define the distance \( d \) of a read \( r \) and the computed haplotypes \( h \) and \( h' \): \( d(r,h,h') = \min(d(r,h),d(r,h')) \)

• So, the total distance is the sum over all read fragments

\[
\text{MEC}(h,h') = \sum_{r_i \in M} d(r_i,h,h')
\]
reconstruction rate

• Using the same distance function now counting the distance between two haplotypes, the reconstruction rate is the percentage of correctly computed bases

• Requires knowledge of the true haplotypes

• Given the computed haplotypes h and h', known haplotypes g and g', and n SNPs

\[ RR(h, h') = 1 - \frac{\min(d(h, g) + d(h', g'), d(h', g) + d(h, g'))}{2n} \]
haplotype switch error

- Counts the number of haplotype switches when comparing computed haplotypes to true haplotypes
- Requires knowledge of true haplotypes
- Was originally developed for haplotype phasing and not well suited for haplotype assembly

\[
\begin{array}{c|c|c}
\text{assembled haplotypes} & \text{true haplotypes} & SE = 4 \\
110|--|0111111111 & 1111111111 & \\
0010--0100000000 & 00000000000000
\end{array}
\]
FMPR and BFM

• We proposed two measures inspired by genome assembly

• Fragment mapping phase relationship (FMPR) counts the number of pairwise phase relationships in the reads that are not present in the solution

• Boolean fragment mapping (BFM) counts the percentage of fragments that do not map without error

\[
\begin{align*}
\text{FMPR} &= 3 \\
\text{BFM (\%)} &= 66.7\%
\end{align*}
\]
Connecting the Compass Graph

We simulated a number of paired Illumina reads on top of the 1000 Genomes Project data for NA12878 to estimate the number of reads required to fully assemble chromosome 22.
Results on real data

Block sizes are quite small due to Illumina data having small insert sizes and read lengths.
Results on real + sim data

After simulating reads on top of the real data, block sizes are larger and more difficult to solve.
Polyploidy

$k=3$

$k=4$

$k=4,6,8,10$
Edge decidability

- New to the haplotype assembly model
- Phasings of edges are no longer disjoint

<table>
<thead>
<tr>
<th></th>
<th>diploid</th>
<th></th>
<th>triploid</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>00</td>
<td>11</td>
<td>01</td>
</tr>
<tr>
<td>01</td>
<td>10</td>
<td>00</td>
<td>10</td>
</tr>
</tbody>
</table>

shared haplotypes= { }  
shared haplotypes= { 00 }
Polyploid edge decidability

- Edge phasing is decided by maximum likelihood

\[
L(p_r|s_e, r_1, r_2, \ldots, r_n) = \frac{P(r_1, r_2, \ldots, r_n|s_e, p_r)}{\sum_{i=1}^{|P|} P(r_1, r_2, \ldots, r_n|s_e, p_i)}
\]

\[
= \frac{P(r_1|s_e, p_p) \cdot P(r_2|s_e, p_p) \cdots P(r_n|s_e, p_p)}{\sum_{i=1}^{|P|} P(r_1, r_2, \ldots, r_n|s_e, p_i)}
\]
Polyploid phase extension

• New to the haplotype assembly model

• Haplotypes no longer compliment, phase extension is no longer unique
3 Polypliod chain

\[
\begin{align*}
\text{V}_1\text{V}_2 & : t_0 \quad \text{11} \quad \text{s}_0 \quad t_0 \\
\text{V}_2\text{V}_3 & : t_0 \quad \text{11} \quad \text{s}_1 \quad t_1 \\
\text{V}_3\text{V}_1 & : t_0 \quad \text{11} \quad \text{s}_2 \quad t_2
\end{align*}
\]
Related haplotype assembly work

• Set cover algorithm
• Feasibility study in parrot
• Third generation sequencing
• HapCompass-ILP
• Copy number variation
• Tumor genomes
Desirable statistical and combinatorial algorithmic properties

Genome-wide (*practical*)

Exact, or as close to as possible (*rigorous*)

General, works for all data (*widely applicable*)

Used in practice (*in use*)

Make a biological and medical impact (*impactful*)
So, what can we do with high quality haplotypes?

- Linkage disequilibrium
- Tag SNP selection
- Phylogenetics
- ARG reconstruction
- Haplotype based associations
- Identity-by-descent
Probability of sharing a locus IBD for $n^{th}$ degree cousins is $2^{-2n}$

Given a locus is shared IBD, the length of sharing is expected to be $\frac{200}{2n+2}$ Mbp on average.
Polyploidy and Aneuploidy

• While polyploidy refers to numerical change in the whole set of chromosomes, aneuploidy refers to organisms in which a part of the set of chromosomes (e.g. a particular chromosome or a segment of a chromosome) is under- or over-represented. Polyploidy and aneuploidy phenomena are recognized as disease mechanisms.
• Examples for polyploidy: triploidy birth conceptions end in miscarriages, although mixoploidy, when both diploid and triploid cells are present, could lead to survival; triploidy, as a result of either digyny (the extra haploid set is from the mother by failure of one meiotic division during oogenesis) or diandry (mostly caused by reduplication of paternal haploid set from a single sperm or dispermic fertilization of the egg) could have parent-of-origin (genomic imprinting) medical consequences: diandry predominate among preterm labor miscarriages while digyny predominates into survival into fetal period, although with a poor grown fetus and very small placenta).

• Examples for aneuploidy: trisomy in the the Down syndrome, cells with one chromosome missing while others with an extra copy of the chromosome, cells with unpredictably many chromosomes of a given type; mosaicism (when two or more populations of cells with different genotypes derived from a single individual) aneuploidy occurs in virtually all cancer cells.
DECĂT O-INCHINE\_RE
ZERO ȘI INFINT SUNT FRĂȘI,
PARCĂ-Ș EU-O FUNIE LECAȚI,
SUNT ȘI CERINȚI SĂU ÎMPĂCAȚI?
VĂ INVIT SĂ JUDECAȚI,
Când știi ÎNFINIT REȘEȘTE
UNU PÎNĂ NU SE OPREȘTE,
DE ZERO ȘI TOT MAI APRUNĂ,
Dar să-l ATINŞA NU SE POATE,
PREȘUȘI ȘI ȘI, MEREU MAI MALE,
Dar din FINIT N-ARE SCĂVARE,
AH, ȘTĂR AR VREU EL SĂ-L ÎNVINGĂ,
DE INFINT SĂ SE ATINŞĂ!
ZERO ȘI INFINT SUNT FRĂȘI,
DE ACEASTĂ LUME DEȘTEPTĂȚI,
CU NOI EI PAR ȘI MEREU,
Dar nu-i DESECE CE PAR ȘI FI,
NU-I ROȚI VEDEREA, MĂI PIPAI.
ȘI TOATE MATEMĂTICA
EȘTE UN ZBUCIU ÎNTEACȚĂ,
NU É DEȘTĂ O-NOCIIȘĂRIKU
DE CARE, VAI, SUSFLETUL MEU,
NICĂND ÅV SĂ-AK UNTER CÎPȘI,