TUMOR HAPLOTYPE ASSEMBLY ALGORITHMS FOR CANCER GENOMICS

DEREK AGUIAR[†], WENDY S.W. WONG[‡,*], SORIN ISTRAIL[†,*]

[†] *Department of Computer Science and Center for Computational Molecular Biology, Brown University, Providence, RI 02912, USA*

[‡] *Inova Translational Medicine Institute, Inova Health Systems, Falls Church, VA 22042, USA*

[*] *Please address correspondence to* Sorin_Istrail@brown.edu *and Wendy S.W. Wong* wendy.wong@inova.org

The growing availability of inexpensive high-throughput sequence data is enabling researchers to sequence tumor populations within a single individual at high coverage. But, cancer genome sequence evolution and mutational phenomena like driver mutations and gene fusions are difficult to investigate without first reconstructing tumor haplotype sequences. Haplotype assembly of single individual tumor populations is an exceedingly difficult task complicated by tumor haplotype heterogeneity, tumor or normal cell sequence contamination, polyploidy, and complex patterns of variation. While computational and experimental haplotype phasing of diploid genomes has seen much progress in recent years, haplotype assembly in cancer genomes remains uncharted territory.

In this work, we describe HapCompass-Tumor a computational modeling and algorithmic framework for haplotype assembly of copy number variable cancer genomes containing haplotypes at different frequencies and complex variation. We extend our polyploid haplotype assembly model and present novel algorithms for (1) complex variations, including copy number changes, as varying numbers of disjoint paths in an associated graph, (2) variable haplotype frequencies and contamination, and (3) computation of tumor haplotypes using simple cycles of the *compass graph* which constrain the space of haplotype assembly solutions. The model and algorithm are implemented in the software package *HapCompass-Tumor* which is available for download from `http://www.brown.edu/Research/Istrail_Lab/`.

Keywords: haplotype assembly; haplotype phasing; tumor haplotypes.

1. Introduction

Cancer is the worldwide 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 is limited and the outlooks for certain types of cancer are usually dire. 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,[1] there have been hundreds of studies on numerous cancer types.[2–5] 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, the chromosomal sequence of alleles, or *haplotype*, is 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.[6] 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. Having the ability to reconstruct tumor haplotypes would enable the discovery of such compound heterozygous relationships between variants and enhance our ability to identify driver mutations.

The computational problem of *haplotype assembly* aims to compute the sequence of co-inherited variant alleles for each chromosome given a set of aligned sequence reads and variants.[7,8] Haplotype assembly of diploid genomes has been addressed by many researchers[9,10] and several haplotype assembly algorithms for diploid genomes are available for use.[11,12] However, the methodologies for diploid haplotype assembly are unable to model polyploid genomes or complex copy number aberrations (CNA). Recently, we developed HapCompass-Polyploidy, the first modeling and algorithm for haplotype assembly in genomes with more than two sets of homologous chromosomes (polyploidy).[13] The HapCompass-Polyploidy algorithm assembles pairs of variants in polyploid genomes and then produces a haplotype assembly consistent with the pairwise variant phasings.

Cancer 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.

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 in a single gene or a small set of genes, advances in technology have enabled us to study large structural variants such as 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.[3,4,14,15] 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.[16] 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.[17] Control-FREEC not only uses the coverage information but also the read count frequencies to estimate CNAs in tumor samples.[18] Control-FREEC 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 paired-end sequence 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 exhibit evidence of a break point for a structural variant, and then locates the other break point by scanning the location neighboring the paired read.[19] However, the accuracy of these methods can be seriously affected when there is contamination in the samples. Cibulskis *et al* developed a Bayesian model to estimate the level of cross-individual contamination in each sample.[20] Contamination may also exist within an individual; tumor tissue can be contaminated with normal DNA and vice versa. Both incorrect variant calling as well as sequence contamination represent sources of complexity and errors for haplotype assembly.

In this work, we leverage the existing literature and tools for cancer genome variant inference and build on the polyploid HapCompass model to construct the first methodology for cancer genome haplotype assembly. In Section 2 we provide the necessary details of the HapCompass polyploid model and extensions for cancer genome haplotype assembly. The modeling section is followed by Section 3 which describes the HapCompass-Tumor algorithm and Section 4 which evaluates the implementation of the algorithm on cancer genome data. Finally, Sections 5 and 6 present a discussion of alternative models of cancer genome haplotype assembly, limitations and extensions to our model, future work, and conclusions.

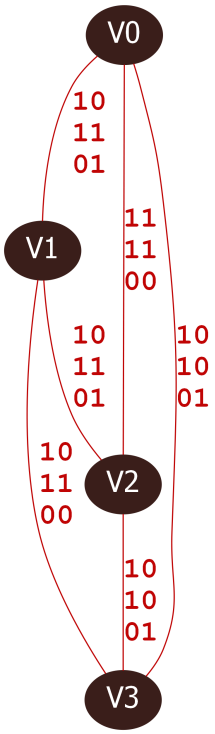

Fig. 1. An example tumor sample G_C with three unique haplotypes. Vertices are variants and edges show pairwise haplotype assemblies.

2. Modeling

Let k be an integer representing the number of unique tumor haplotypes in a sample of tumor tissue. Because the tumor is actively evolving, this k may vary for independent samples of the same tumor. We assume that each sequence read is sampled from a single haploid fragment generated from one of the k haplotypes; this property enables the building of haplotype phase relationships between alleles in sequence reads that contain two or more *heterozygous* variants (homozygous variants do not provide phase information for assembly). The *phase-informative* sequence reads and variants are modeled with two graph structures termed the compass graph, G_C, and chain graph, G_h. These data structures are described in Aguiar *et al.* 2013 but their definitions are repeated here in order to present the novel aspects of the model for tumor genomes.[13]

The compass graph $G_C(V_C, E_C)$ has $v \in V_C$ for each variant and $(v_i, v_j) \in E_C$ if variants v_i and v_j are contained within a sequence read. Edges (v_i, v_j) are annotated with the most likely haplotype phasing between variants v_i and v_j given the set of reads that contain both v_i and v_j (Figure 1).

2.1. *Phasing edges of G_C*

Given the probability of sequencing error, s_e and a set of reads overlapping the two variants, $r_1, ..., r_n$, the likelihood of a particular phasing, p_p from the set of all phasings between the two variants P can be computed as in Equation 1. Edges of G_C are phased by choosing the p_p that maximizes this likelihood.

$$L(p_p|s_e, r_1, r_2, ..., r_n) = \frac{P(r_1|s_e, p_p) \cdots P(r_n|s_e, p_p)}{\sum_{i=1}^{|P|} P(r_1, r_2, ..., r_n|s_e, p_i)} \quad (1)$$

Equation 1 models haplotypes that are in equal proportion which may not be true for heterogeneous tumor samples. Thus the likelihood must be modified to accommodate the different frequencies of haplotypes. Consider the normal haplotype contamination that is often present in tumor sequence samples. Contamination may be modeled 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 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

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

The number of unique phasings of an edge depends on the number of unique tumor haplotypes k and the allele content of the variant pair. Let the number of 1 alleles for variants v_i and v_j be $l(v_i)$ and $l(v_j)$ respectively. Then, the number of phasings of an edge is upper bounded by $min\left(\binom{k}{l(v_i)}, \binom{k}{l(v_j)}\right)$. This is a bound and not equality because some phasings may be repeated in this enumeration.

2.2. *Chain graph*

Haplotype phasings of the edges of G_C can be extended to paths. Because two adjacent edges share a variant, haplotypes with the same allele can be merged on the shared vertex. If two paths in G_C of length i and j vertices are merged, the new phasing will have $i + j - 1$ variants.

For paths or trees in G_C, there is exactly (at least) one consistent haplotype phasing, with respect to the edge phasings along the path or tree, for genomes with $k = 2$ ($k > 2$). In contrast, simple cycles in G_C may be either *conflicting* or *non-conflicting* depending on how many phasings are consistent with the cycle. A *conflicting* cycle does *not* have a consistent phasing while a *non-conflicting* cycle has at least one. The *chain graph* G_h is constructed for each simple cycle to determine its conflicting state.[13]

The chain graph $G_h(V_h, E_h)$ is constructed for a path or simple cycle $c = ((v_1, v_2), ..., (v_{s-1}, v_s), (v_s, v_1))$ in the compass graph G_C. We introduce k haplotype vertices corresponding to the phasing for each edge (v_i, v_j) in the path or cycle. Vertices in G_h created

from adjacent edges of G_C share a variant; edges connect vertices in G_h if they share a variant and allele. Then, source nodes $s_1, ..., s_k$ are arbitrarily assigned to vertices at level 1 and sink nodes $t_1, ..., t_k$ are assigned to vertices at level s if the level s vertex shares an allele with the level 1 vertex. Vertices are annotated with t_i if there exists at least one s_i to t_i path which is computed by a depth first from each source. G_h can be described as a *trellis graph* in which the vertices can be divided into levels; each level in this case corresponds to an edge of G_C. Trellis graphs have a wide range of applications including communication network topology and survivability, encryption, encoding and decoding, and are a central data structure in Markov models.

2.3. *Disjoint s_it_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 in the corresponding G_h. 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 in the corresponding G_h. 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 2.1).

Lemma 2.1. *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 node 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 one or more phased haplotype nodes contain no sink annotation, then at least one phased haplotype node must be shared by 2 or more haplotype paths which breaks vertex disjointness. □

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

2.4. *Copy number aberrations and translocations in G_h*

The chain graph and disjoint paths framework accommodates modeling the types of variation typical of tumor genomes (Figure 2). CNAs insert or remove large genomic regions. Genomic deletions are modeled as an 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 combining deletions and insertions.

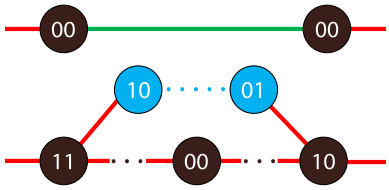

Fig. 2. Deletions and insertions are modeled with disjoint paths. The green edge models a deletion which effectively removes the deleted variants in the chain graph. The blue node insertion adds an extra path in G_h.

2.5. *Disjoint subgraphs in the general chain graph*

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

G_g is built from the conflict-free 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). Each 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 2.1, 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. We assume each of the l valid extensions of the sets of matchings at adjacent levels e_i and e_j are equally likely. Then, the weight of a particular extension $\frac{w(e_i)w(e_j)}{l}$ where $w(e_i)$ is the score or likelihood of edge e_i, 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 3). 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 spanning tree problem. That is, we compute a set of k node-disjoint (within levels) spanning trees in G_g whose total weight is maximum over all k node-disjoint spanning trees and includes every vertex in G_g.

3. HapCompass-Tumor Algorithm

HapCompass-Tumor optimizes the minimum weighted edge removal (MWER) problem. 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 G_h is added to G_g. G_g represents the constrained solution space by incorporating the

Fig. 3. (Left) An example tumor genome G_C with three non-conflicting cycles. 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 $((v_0, v_1), (v_0, v_2))$ is included in G_g creating a cycle.

valid haplotype assemblies on subsets of variants computed from each non-conflicting G_h (Algorithm 1).

input : Sequence reads, variant calls, and number of distinct haplotypes k
output: k haplotypes

$G_C \leftarrow$ spanning tree cycle basis
$C_C \leftarrow$ set of conflicting simple cycles with respect to G_C
for $c_C \in C_C$ **do**
  Remove edge with smallest likelihood in c_C
  Reconstruct G_C
end
Compute G_g
$C_N \leftarrow$ set of non-conflicting simple cycles with respect to G_C
for $c_N \in C_N$ **do**
  Compute G_h with respect to c_N
  Compute matchings at each level of G_h
  Compute disjoint paths of G_h
  Increase the weight of each edge e between levels shared by G_h and G_g in G_g proportional to the number of disjoint paths using edge e and the likelihood of each edge (Equations 1 and 2)
end
Compute a maximum weight spanning tree of the adjacencies in G_g
Output the haplotype assembly computed from the spanning tree of G_g

Algorithm 1: HapCompass-Tumor

The final step involves computing k spanning trees in G_g which are node disjoint in respect to haplotype level vertices. Adjacencies between levels in G_g correspond to matchings between the haplotype nodes (Figure 4 right). So, HapCompass-Tumor computes k disjoint spanning trees corresponding to the k tumor haplotypes. We have implemented two algo-

rithms inspired by Kruskal's and Prim's algorithms for computing maximum spanning trees. The principle difference between the two algorithms in the context of HapCompass is the Kruskal-like algorithm focuses on constructing disjoint trees by including strong phasings on the same haplotype (edges of G_g) while the Prim-like algorithm phases all haplotypes between two levels at a time (vertices of G_g).

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 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 4 (left) are constructed. Figure 4 (right) shows the G_g updated after the disjoint paths and weights of edges in G_h are computed and distributed to G_g.

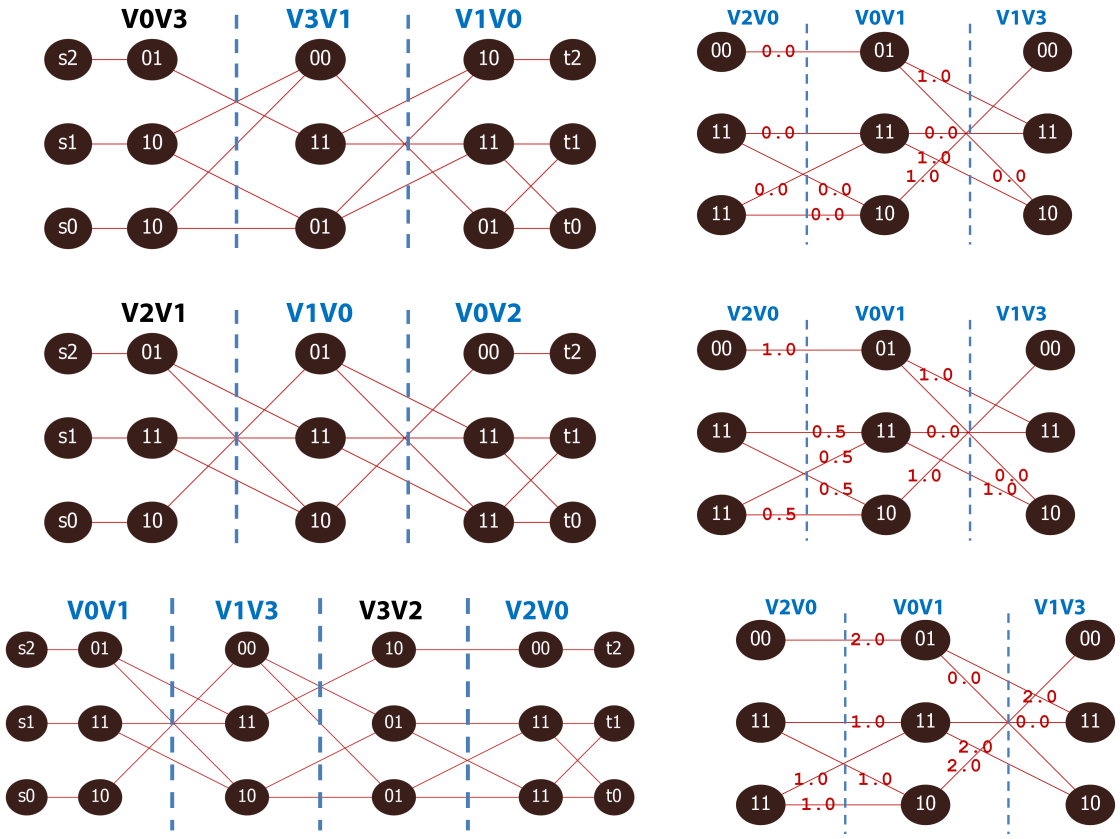

Fig. 4. (Left) chain graphs (G_h) from the compass graph in Figure 1. The level corresponding to edges in G_C are denoted by black (non-tree edges) and blue (spanning tree edges) lettering above the vertices. In this example, the edge phasing probabilities in G_C are all 1. So, an edge connecting level 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).

4. Results

We implemented HapCompass-Tumor and evaluated its performance on simulated 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

length between the read pairs.[13,21] 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 $x > y$.

4.1. *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. 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 (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.

4.2. *Cancer genome heterogeneity*

We also compared the accuracy of haplotype assembly in terms of tumor genome heterogeneity (Figure 5 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.

4.3. *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[22] and the HCC1187 tumor sample

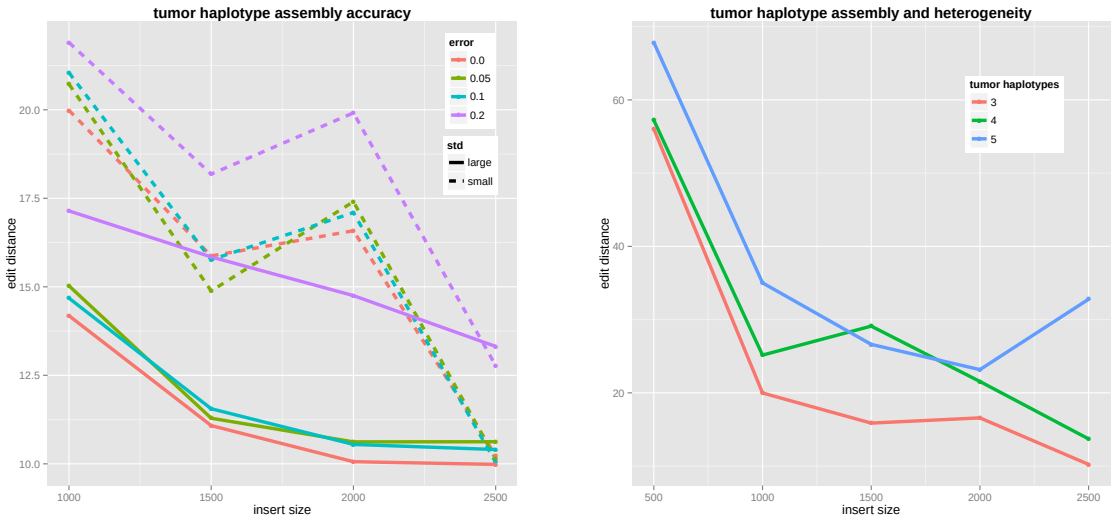

Fig. 5. (Left) The average edit distance between haplotypes and the simulated true haplotypes is calculated with a fixed coverage and varying insert sizes, error rates (error), and standard deviations (std). (Right) Haplotype assembly accuracy is plotted as a function of the number of tumor haplotypes in the sample.

(downloaded from Illumina's Basespace[23]). 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 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 1 shows the results for HapCompass-Tumor using the Kruskal-like 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 1). This scheme is designed to limit uninformative pairs of vertices in the spanning tree of the compass graph G_C.

Table 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.

5. Discussion

Opportunities exist to extend HapCompass-Tumor to address some of the limitations in the current model. First, HapCompass-Tumor 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

Table 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

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 would 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 probability of both phasings is 1 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 could penalize the extension.

A related application of HapCompass-Tumor 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;[24,25] and somatic ASE is associated with tumor development.[26] ASE in cancer was found not only correlated with CNAs,[26] but also with allelic specific methylation (ASM).[27] Existing algorithms for detecting ASE with RNA-seq and detecting ASM with Bisulfite-Seq do not usually make use of phased genotype information.[26,28] We therefore propose using the phased haplotypes from whole genome sequencing of tumor samples as a reference for RNA-seq and Bisulfite-seq alignment when such data is available.

Finally, the viral quasispecies reconstruction (VQR) problem 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 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.

6. Conclusions

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 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 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. HapCompass-Tumor is available for download from `http://www.brown.edu/Research/Istrail_Lab/`.

7. Acknowledgements

We thank Lilian Janin and Anthony Cox at Illumina Cambridge Ltd for sharing and helping us with the EAGLE simulator. This work was supported by the National Science Foundation [1048831 and 1321000 to S.I.].

References

1. T. J. Ley, E. R. Mardis *et al.*, *Nature* **456**, 66 (November 2008).
2. E. D. Pleasance, R. K. Cheetham *et al.*, *Nature* **463**, 191 (2009).
3. M. Meyerson, S. Gabriel and G. Getz, *Nature Reviews Genetics* **11**, 685 (October 2010).
4. The Cancer Genome Atlas, *Nature* **490**, 61 (September 2012).
5. E. R. Mardis, *Curr. Opin. Genet. Dev.* **22**, 245 (Jun 2012).
6. A. G. Knudson, *Proceedings of the National Academy of Sciences* **68**, 820 (1971).
7. R. Lippert, R. Schwartz, G. Lancia and S. Istrail, *Brief Bioinform* **3**, 23 (March 2002).
8. S. Istrail, *The Haplotype Phasing Problem*, tech. rep., Celera Genomics (2002).
9. R. Schwartz, *Commun. Inf. Syst.* **10**, 23 (2010).
10. F. Geraci, *Bioinformatics (Oxford, England)* **26**, 2217 (September 2010).
11. D. Aguiar and S. Istrail, *J. Comput. Biol.* **19** (2012).
12. V. Bansal and V. Bafna, *Bioinformatics* **24**, i153 (August 2008).
13. D. Aguiar and S. Istrail, *Bioinformatics* **29**, i352 (2013).
14. L. Ding, M. J. Ellis *et al.*, *Nature* **464**, 999 (May 2010).
15. W. Lee, Z. Jiang *et al.*, *Nature* **465**, 473 (May 2010).
16. C. T. Saunders, W. S. W. Wong *et al.*, *Bioinformatics (Oxford, England)* **28**, 1811 (July 2012).
17. D. C. Koboldt, Q. Zhang *et al.*, *Genome research* **22**, 568 (March 2012).
18. V. Boeva, A. Zinovyev *et al.*, *Bioinformatics (Oxford, England)* **27**, 268 (January 2011).
19. J. Wang, C. G. Mullighan *et al.*, *Nature Methods* **8**, 652 (2011).
20. K. Cibulskis, A. McKenna *et al.*, *Bioinformatics (Oxford, England)* **27**, 2601 (September 2011).
21. B. V. Halldorsson, D. Aguiar and S. Istrail, *Pacific Symposium of Biocomputing* , 88 (2011).
22. Genome in a Bottle Consortium, NIST NA12878 Highly Confident integrated genotype (April 2013), `ftp://ftp-trace.ncbi.nih.gov/giab/ftp/data/NA12878/variant_calls/NIST/`.
23. Illumina Inc., BaseSpace G.C.C. (April 2013), `https://basespace.illumina.com/home/index`.
24. L. Valle, T. Serena-acedo *et al.*, *Science* **321**, 1361 (2009).
25. C. Gao, K. Devarajan *et al.*, *BMC genomics* **13**, p. 570 (January 2012).
26. B. B. Tuch, R. R. Laborde *et al.*, *PloS one* **5**, p. e9317 (January 2010).
27. P.-C. Lin, E. G. Giannopoulou *et al.*, *Neoplasia* **15**, 373 (2013).
28. F. Fang, E. Hodges *et al.*, *Proceedings of the National Academy of Sciences* **109**, 7332 (2012).