Supplemental Methods for "HapCUT2: robust and accurate haplotype assembly for diverse sequencing technologies"

Peter Edge, Vineet Bafna and Vikas Bansal

Maximum Likelihood cut heuristic

Maximum-Likelihood-Cut(H,R)
Initialization: $c = 0,\ S^* = \varnothing$
Iteration: for $i = 1 \ldots M$:

1. Chose a pair of vertices (u, v)
2. Initialize $S_1 = \{u\}$, $S_2 = \{v\}$ and $S = S_1 \cup S_2$
3. **While** $|S| < |V|$
   (a) Let $w' = \arg\max_{w \in V - S} |L(w)|$.
   (b) **If** $L(w') < 0$, $S_1 = S_1 \cup \{w'\}$
   (c) **else if** $L(w') > 0$, $S_2 = S_2 \cup \{w'\}$
   (d) **else** add w' uniformly at random to S_1 or S_2
4. **If** $p(R|q, H(S_1)) > p(R|q, H(S^*))$
   (a) $S^* = S_1$, $c = 0$
   else $c = c + 1$
5. If $c > C$: break

Return: S^*

Implementation of HapCUT2

HapCUT2 operates on each connected component or haplotype block of the read data to search for good haplotypes iteratively using the maximum-likelihood-cut heuristic. Each fragment is stored as a list of blocks that cover consecutive variants. This compact representation is efficient at storing long reads as well as paired-end reads that span a large number of variants. For short read data, HapCUT2 stores all pairs of edges corresponding to each fragment in the read-haplotype graph. However, for long read datasets, HapCUT2 reduces the number of edges in the graph by only storing edges for adjacent variants in each fragment. This is sufficient for determining the connected components in the read-haplotype graph and also for selecting an edge to initialize the cut in the Maximum-Likelihood-Cut heuristic. Note that HapCUT2 still has to consider all pairs of edges per fragment in order to search for good cuts. Therefore, the computational complexity of HapCUT2 scales as V^2 where V is the maximum number of variants covered by a fragment.

The first step in the Maximum-Likelihood-Cut heuristic is to select a pair of vertices to initialize the cut. In the original HapCUT method (Bansal and Bafna 2008), edges were selected at random from the read-haplotype graph to initialize the cut. This requires a large number of iterations (proportional to number of edges in the graph) to ensure that 'good' edges are considered. An alternate greedy approach

(also used in the RefHap algorithm (Duitama et al. 2010)) is to identify edges such that the current phase between the pair of vertices (as defined by the haplotype H) is highly inconsistent with the fragment data. Such edges can be found by sorting the list of edges in the read-haplotype graph by weight and selecting the lowest weight edges. HapCUT2 uses a hybrid approach (combination of K lowest weight edges (default $K = 5$) and $\min(N/10, 100)$ randomly sampled edges where N is the number of variants in the block) to initialize the cut in the maximum-likelihood-cut heuristic. Therefore, the maximum number of iterations for the maximum-likelihood-cut routine is $M = K + \min(N/10, 100)$.

Likelihood-based variant pruning

Following haplotype assembly, HapCUT2's variant pruning scheme makes a single pass over the haplotype H in which it considers each variant i of H separately. The goal is two-fold: firstly, if the likelihood of the haplotype can be improved by changing the haplotype or genotype assignment at i, then the allele or genotype is reassigned. Secondly, if the haplotype is low confidence at i even after being reassigned, position i is pruned from the solution. While considering position i, it is assumed that that the rest of the haplotype $H[i' \neq i]$ is correct. At variant i, each of four possibilities are considered for the two alleles in the ordered pair of haplotypes: $\{10, 01, 11, 00\}$. Let $H_{i\to x}$ denote H that has been modified to have phasing $x \in \{10, 01, 11, 00\}$ at position i. Optionally, HapCUT2 supports obtaining the prior probabilities of the unordered genotype configurations $(00), (01), (11)$ from the VCF genotype likelihoods. The results obtained in this paper used the default behavior, which sets prior probabilities of 0 for the homozygous configurations (00) and (11), such that genotype calls are assumed to be correct. The prior probabilities for the two haplotype configurations (01) and (10) are set to be equal. Then, the posterior probability of each possibility can be calculated as:

$$P(H_{i\to x}|q, R, H) = \frac{p(x)p(R|q, H_{i\to x})}{\sum_{y\in\{10,01,11,00\}} p(y)p(R|q, H_{i\to y})} \tag{1}$$

The posterior probability of the most likely configuration is:

$$P_H[i] = \max_{x\in\{10,01,11,00\}} P(H_{i\to x}|q, R, H) \tag{2}$$

For a given position, the haplotype is assigned to the configuration that maximizes $P_H[i]$. If $P_H[i] < \alpha$ for some user-defined threshold $\alpha \in [0.5, 1]$, the variant is pruned from the final result. The default value of α is 0.8. HapCUT2 also offers the RefHap heuristic as an alternative to the likelihood based method, which may be preferable when quality scores are not accurate.

Block Splitting

HapCUT2 includes an optional scheme for splitting blocks at low-confidence sites. Let $H_{s(i)}$ denote H that has been edited to have a switch starting at position i. That is, every position from i onwards is flipped with respect to those before i. Similarly to before, we assume that $H[1..(i-1)]$ and $H[i..N]$ are correct.

$$P(H_{s(i)}|q, R, H) = \frac{p(R|q, H_{s(i)})}{p(R|q, R, H) + p(R|q, R, H_{s(i)})} \tag{3}$$

Under the assumption that $H[1..(i-1)]$ and $H[i..N]$ are correct, this is equivalent to computing a Bayesian posterior probability of a switch at i with equal priors. After computing the posterior probability of each phasing, a block is split at i if $1 - P(H_{s(i)}|q, R, H) < \alpha_2$ for some user-defined threshold $\alpha_2 \in [0.5, 1]$

Estimating $\tau(I)$ for Hi-C reads

In order to properly model h-trans error we must know the probability that a read pair with insert size I is h-trans, i.e. the two ends of the paired-end read originate from different homologous chromosomes. Selvaraj et al. estimated these probabilities using mouse Hi-C data where the haplotypes was known. We estimated the function τ for the NA12878 MboI data using the known trio phase and observed that τ varies from chromosome to chromosome, with certain chromosomes such as chromosome 17 and 19 having rates of h-trans error several times larger than others. It is possible that the rate of h-trans error

may also vary across different cell types. Therefore, it would be ideal to estimate the rate of h-trans error directly from the data as a part of the haplotype assembly process.

Assume that we have assembled the haplotypes (H) from the Hi-C reads using HapCUT2-Assemble. Our goal is to estimate the probability $\tau(I)$ that a paired-end read with distance between the two inner ends equal to I represents an h-trans read. We assume that this probability is the same for all reads that have insert size I. For one such read R, let us assume without loss of generality that the haplotype pair for the two variants covered by the read is $(00, 11)$. If the read sequence is also 00 or 11, the read matches the haplotype pair H. This can happen if the read is a cis-read and has 0 or 2 sequencing errors or if it is a trans-read and has a sequencing error at only one end. The probability of this is:

$$(1-\tau(I))\left[(1-q_1)(1-q_2)+q_1q_2\right]+\tau(I)\left[(1-q_1)q_2+(1-q_2)q_1\right]=(1-\tau(I))a+(\tau(I))(1-a)$$

where q_1 and q_2 are sequencing error probabilities and a is the probability that the read pair has 0 or 2 sequencing errors (at the variant sites). Conversely, if the read sequence at the two variants is 01 or 10, the read can be either (i) a cis-read with one sequencing error, or (ii) a trans-read with 0 or 2 sequencing errors. The likelihood of the read in this case is:

$$(\tau(I))a+(1-\tau(I))(1-a)$$

The likelihood of each read can be calculated using the above two expressions. The joint likelihood of all reads with an insert length equal to I is simply the product of individual read likelihoods and is a function of the variable $\tau(I)$. To get a maximum likelihood estimate of $\tau(I)$, we simply find the value of $\tau(I)$ that maximizes this likelihood function. It is not difficult to show that the maximum likelihood estimate of $\tau(I)$ is:

$$\frac{\sum_{R_i,R_i=H} b_i+\sum_{R_i,R_i\neq H} a_i}{N_I} \tag{4}$$

where N_I is the total number of reads with insert length I.

HapCUT2-HiC-Mode(R)
Initialization: $H=H^0, H_{old}=H^0$
$\tau(I)=0$ for all I
Iteration: while $p(R|q,H,\tau)\geq p(R|q,H_{old},\tau)$:

1. $H_{old}=H$
2. $H=$ HapCUT2-HiC-Assemble(R,τ)
3. estimate $\tau(I)=\frac{\sum_{R_i,R_i=H} b_i+\sum_{R_i,R_i\neq H} a_i}{N_I}$ for each insert size I using all reads with insert size I

Return: H

HapCUT2-HiC-Assemble, used as a subroutine by HapCUT2-HiC-Mode, is the exact same as HapCUT2-Assemble described in the main text, except that it incorporates τ into all read likelihood calculations. Therefore, the algorithm works by iteratively assembling a complete haplotype H from the reads and τ using an "h-trans aware" version of the assembly algorithm, estimating a new τ from H, and repeating. Note that the haplotypes assembled by HapCUT2 are expected to have some errors, particularly at low coverage. If we can calculate the posterior probability of the phasing between each pair of variants, we could estimate τ using an exact EM approach. However, the posterior probability of the phase between a pair of variants depends on the errors in the paths in the read-haplotype graph between the two variants and is computationally infeasible to calculate. Nevertheless, we found that this EM-like approach was able to accurately estimate the h-trans error rates at sufficient coverage (Figure S5), and the model consistently improved switch error and mismatch accuracy both at modest coverage levels such as 30× and 40× and high coverage levels such as 90×.

Extraction of haplotype informative reads

Most haplotype assembly algorithms use a haplotype "fragment file" as input. This file represents each haplotype informative read or fragment as a list of heterozygous variants (indices of variants in the VCF file) and the corresponding alleles (with quality values) at each variant site. Consecutive heterozygous variants covered by a single read are compressed to form a single block. This format was first utilized in the assembly of the whole-genome Sanger sequence data for HuRef (Levy et al. 2007). The ExtractHAIRs (Extract HAplotype Informative Reads) program was created to process aligned reads in a sorted BAM file and heterozygous variants from a VCF file (with variant calls) to create the haplotype "fragment file". It has been available as part of the HapCUT software package since 2011 and can efficiently process paired-end sequence data as well as long read datasets such as PacBio reads. For analyzing Hi-C data, the fragment file format was modified to store extra information for each read including the data type (Hi-C or long read), variant start index of the second read and paired-end insert size.

For 10X Genomics linked-reads, reads with the same barcode may originate from the same DNA molecule or from several other DNA molecules scattered over the genome. For this reason, reads with the same barcode that were separated by 20 Kb or more were called as originating from separate molecules. The molecule boundaries were derived from the aligned BAM file using a python script and the haplotype fragment for each molecule was extracted from the BAM file using the extractHAIRS tool (code available at https://github.com/vibansal/hapcut2).

Post processing of alignments for Hi-C reads

We observed that a significant fraction of the reads (∼10-20%, depending on the experiment) contained a chimeric mate resulting from the ligation junction being located towards the ends of DNA fragment. This resulted in one read reading past the ligation point of the Hi-C fragment, making it a chimera of its own sequence and a sequence originating from near the other read's location. These chimeras appeared in the BWA alignment as primary and secondary alignments. For the purpose of haplotype assembly, it is important to have as much sequence material as possible. For this reason, we repaired chimeric alignments by cutting the chimeric portion and pasting it to the other mate with an added gap. This post-processing script is freely available with HapCUT2. Following Hi-C chimera repair, the single end alignments for each paired end reads were combined and mate information was filled in with samtools fixmate (Li et al. 2009). Subsequently, the BAM files were sorted with samtools sort and PCR duplicates were marked for removal with Picard MarkDuplicates (http://broadinstitute.github.io/picard). Finally, bam files were split by chromosome with BamTools split (Barnett et al. 2011) and converted to HapCUT fragment matrix format using extractHAIRs.

Experiment and Pipeline Management

Processing and experiment pipelines were managed with Snakemake software (Koster and Rahmann 2012). A Snakemake snakefile is available with the HapCUT2 software that will reproduce the results of this paper (all main and supplemental figures) from raw online data sources.

References

Bansal V and Bafna V. 2008. HapCUT: an efficient and accurate algorithm for the haplotype assembly problem. *Bioinformatics* **24**: i153–159.

Barnett DW, Garrison EK, Quinlan AR, Strömberg MP, and Marth GT. 2011. Bamtools: a c++ api and toolkit for analyzing and managing bam files. *Bioinformatics* **27**: 1691–1692.

Duitama J, Huebsch T, McEwen G, Suk EK, and Hoehe MR. 2010. Refhap: A reliable and fast algorithm for single individual haplotyping. In *Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology*, BCB '10, pp. 160–169. ACM, New York, NY, USA.

Duitama J, McEwen GK, Huebsch T, Palczewski S, Schulz S, Verstrepen K, Suk EK, and Hoehe MR. 2012. Fosmid-based whole genome haplotyping of a HapMap trio child: evaluation of Single Individual Haplotyping techniques. *Nucleic Acids Res.* **40**: 2041–2053.

Koster J and Rahmann S. 2012. Snakemake: a scalable bioinformatics workflow engine. *Bioinformatics* **28**: 2520–2522.

Levy S, Sutton G, Ng P, Feuk L, Halpern A, Walenz B, Axelrod N, Huang J, Kirkness E, Denisov G, et al.. 2007. The diploid genome sequence of an individual human. *PLoS Biol.* **5**: e254.

Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, and Durbin R. 2009. The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **25**: 2078–2079.