# Read-based phasing for dense and accurate haplotyping of individual genomes

# Outline

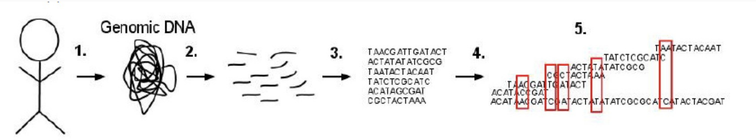
1. Haplotype Phasing
2. Diploid phasing
3. Not Only Diploid
4. References

# Haplotype Phasing

# Haplotype Phasing

**A haplotype is the sequence of nucleotides along a single chromosome.**

- **Why?**
  - Understanding genetic variation in disease and reconstructing population history.
- **How?**
  - Pedigree (e.g. trio-based phasing).
  - Phasing by linkage disequilibrium.
  - Identity-by-descent in unrelated individuals.
  - Assemble multiple reads generated by different sequencing technologies into long haplotypes (the only viable approach for haplotype phasing on a single individual as other approaches either require family members or a population).



# Diploid phasing

# HapCUT2[1]

Only consider heterozygous variants for phasing.

- Notation:
  - $H = (H_1, H_2)$: pair of haplotypes with length n, denoted by binary string.
  - R: reads, denoted by a string of length n over the alphabet {0,1,-} where - corresponds to heterozygous loci not covered by the read.
  - $q_i[j]$: the probability that the allele call at variant j in read $R_i$ is incorrect.
- Likelihood:

$$p\left(R_i|q,h\right) = \prod_{j,R_i[j]\neq -} \delta\left(R_i[j],h[j]\right)\left(1-q_i[j]\right) + (1 - \delta\left(R_i[j],h[j]\right))\,q_i[j]$$

$$p\left(R_i|q,H\right) = \frac{p\left(R_i|q,H_1\right) + p\left(R_i|q,H_2\right)}{2}$$

$$p(R|q,H) = \prod_i p\left(R_i|q,H\right)$$

# HapCUT2-Methods

HapCUT2: a greedy algorithm for finding the maximum likelihood cut is to find a subset of variants or vertices S such that the haplotype H(S) has better likelihood than the current haplotype H.

- MAX-CUT: a subset S of the vertex set such that the number of edges between S and the complementary subset is as large as possible.

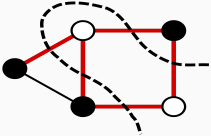


**Figure 1:** From Wiki

- Read-haplotype graph $G_R(H)$: variants are nodes, and edges correspond to pairs of variants that are connected by a read.
- Partial likelihood function: $p_S(R|q,H) = \prod_i p_S(R_i|q,H)$.

# HapCUT2-Methods

Algorithm:

- Initialize the two vertices $S_1$ and $S_2$.
- Add vertex v to $S_1$ such that it maximizes the absolute difference:

$$L(v) = \log\left[p_S\left(R|q, H\left(S_1 \cup \{v\}\right)\right)\right] - \log\left[p_S\left(R|q, H\left(S_1\right)\right)\right]$$

  and $L(v) < 0$, where $S = \{S_1 \cup S_2 \cup v\}$.
- Results in a new haplotype $H\left(S_1 \cup v\right)$ if $p\left(R|q, H\left(S_1 \cup v\right)\right) > p(R|q, H)$.
- Stop until $p(R|q, H)$ stops changing.

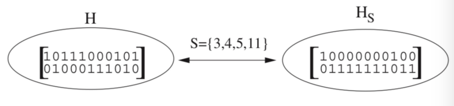


**Figure 2:** Final step – flip the order

# Not Only Diploid

# Ployploid Haplotype Phasing

Polyploid haplotypes mainly come from plant genomes.

**Why?**

Crop breeding is very important. Most widely cultivated species of some economically important crops such as wheat, cotton, apple and peanuts are polyploids.

**More difficult!**

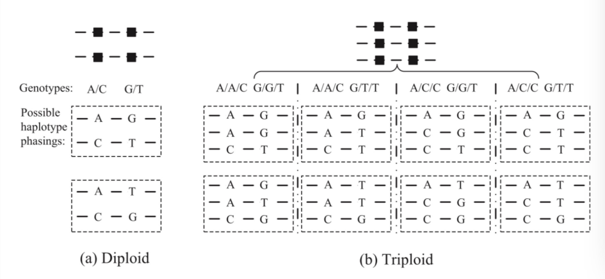


**Figure 3:** From [3]

# Poly-Harsh[2]

What if we have **disconnected haplotype**?

**Why?**

1. Adjacent variants might be far from each other, namely their distance is longer than the length of the reads.

2. The sequencing coverage is low and thus not all variants are covered.

**Solution — split the haplotype into blocks:**

1. Create a graph where the nodes are variants and an edge between two variants indicates that the two variants are connected by some reads.

2. Identify the connected components of the graph, which are the variants contained in each haplotype block.

3. Phase each block independently, using only the reads covering the variants for that specific block.

# Pahsing Algorithm

Assume k haplotypes in total.

- Notation
  - Read assignment vector: $r_j = [a_1, a_2, \ldots, a_k]$, 1 if the read is assigned to the i-th haplotype, 0 otherwise.
  - Binary encoding: $h_i = [g_{1,i}, g_{2,i}, \ldots, g_{k,i}]$, compare a $h_i$ to the reference, if the allele is the same the value is 0, 1 otherwise.
  - Define the probability of the correct read assignment given the matches between the read and the haplotypes:

$$\theta\left(h_i, r_j, x_j\right) = \ln(1-\epsilon)^t + \ln(\epsilon)^{k-t}$$
$$t = \text{match}\left(h_i, r_j \times x_{j,i}\right)$$

, where $\epsilon$ is the sequencing error rate, $x_{j,i}$ is the i-th value of read $x_j$, match(A, B) is the vector-wise matches between two vectors A and B. e.g. $h_i = [1,1,0,0], r_j = [1,0,0,0], x_{j,i} = 1$, then t = 3.

# Pahsing Algorithm

**2 Main Steps:**

Sample $H = [h_1, h_2, \ldots, h_n]$ based on conditional probability (for ploidy k, there are $2^k$ haplotype values for a variant):

$$P(h|R) = \frac{\exp\left(\sum_{j=1}^{n} \theta\left(h, r_j, x_j\right)\right)}{\exp\left(\sum_{i=1,j=1}^{i=2^k,j=n} \theta\left(h_i, r_j, x_j\right)\right)}$$

Update the read assignment $R = [r_1, r_2, \ldots, r_n]$:

$$P(r|H) = \frac{\exp\left(\sum_{j=1}^{2^k} \theta\left(h_j, r, x\right)\right)}{\exp\left(\sum_{i=1,j=1}^{i=k,j=2^k} \theta\left(h_j, r_i, x\right)\right)}$$

**Goal** – Find optimal $H$ that minimizes MEC:

$$\mathrm{MEC}(X, H) = \sum_{j=1}^{n} \sum_{i=1}^{k} r_{ji} \times D\left(x_j, H_i\right)$$

, where $D\left(x_j, H_i\right)$ is the number of mismatches between $x_j$ and $H_i$.

# Pahsing Algorithm

```
Require: ploidy k, set of aligned reads X, error rate $\epsilon$.
Ensure: k phased haplotypes
1: Randomly Initialize k haplotypes H
2: For fixed haplotype H, sample read origin R
3: For fixed read origin R, sample haplotype H
4: mec = MEC(H, R)
5: Repeat steps 2 and 3 for sufficient rounds until equilibrium
6: Collect haplotypes and the corresponding MEC by repeating steps 2 and 3, and output the one with the minimum MEC.
```

# Contiguous Reconstruction Algorithm

1. At the beginning for each sample it builds all the candidate haplotypes by concatenating the subsequences in each possible ways from the ordered list of blocks.
2. Find the set of candidate haplotypes which occurs at least twice across the entire set of samples.
3. By utilizing the pruned set of candidate haplotypes, detect all the 4 haplotypes of each sample.

# References

# References

[1]Edge,P. et al. (2017) HapCUT2: robust and accurate haplotype assembly for diverse sequencing technologies. Genome Res., 27, 801812.

[2]He D., Saha S., Finkers R., Parida L., 2018. Efficient algorithms for polyploid haplotype phasing. BMC Genomics 19: 110.

[3]Xie M, Wu Q, Wang J, Jiang T. H-pop and h-popg: heuristic partitioning algorithms for single individual haplotyping of polyploids. Bioinformatics. 2016;32(24):373544.

[4]Delaneau,O. et al. (2013) Haplotype estimation using sequencing reads. Am. J. Hum. Genet., 93, 687696.

[5]Bansal V, Bafna V. 2008. HapCUT: an efficient and accurate algorithm for the haplotype assembly problem. Bioinformatics 24: i153i159.