HapCUT: An efficient and accurate algorithm for Haplotype Assembly

Vikas Bansal and Vineet Bafna

UC San Diego

Genetic variation and disease association

- Different human individuals have small variation in their DNA (genetic variance).
- Small genetic variation often have important phenotypic consequences.

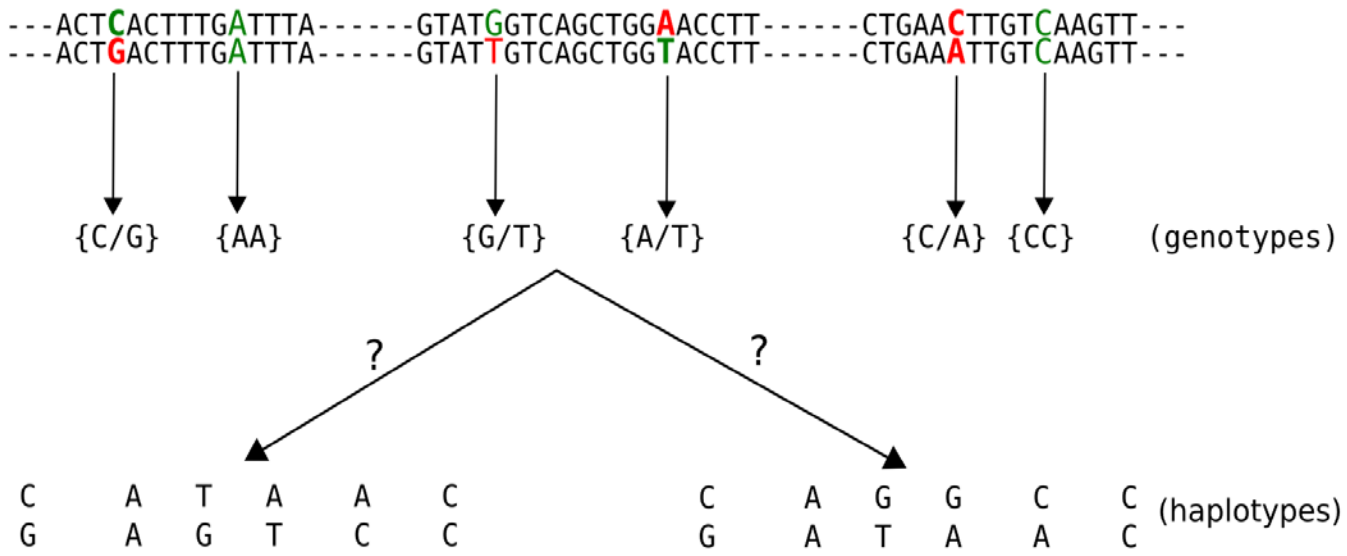
- Therefore variants that are in common in populations are being genotyped and correlated with phenotypes (diseases).

Haplotypes and Disease Association

G	–	T	-	G	*Dis*.
G	–	T	-	G	*Dis*.
G	–	A	-	G	*Nor*.
C	–	A	-	A	*Nor*.
C	–	A	-	A	*Nor*.
C	–	A	-	G	*Nor*.

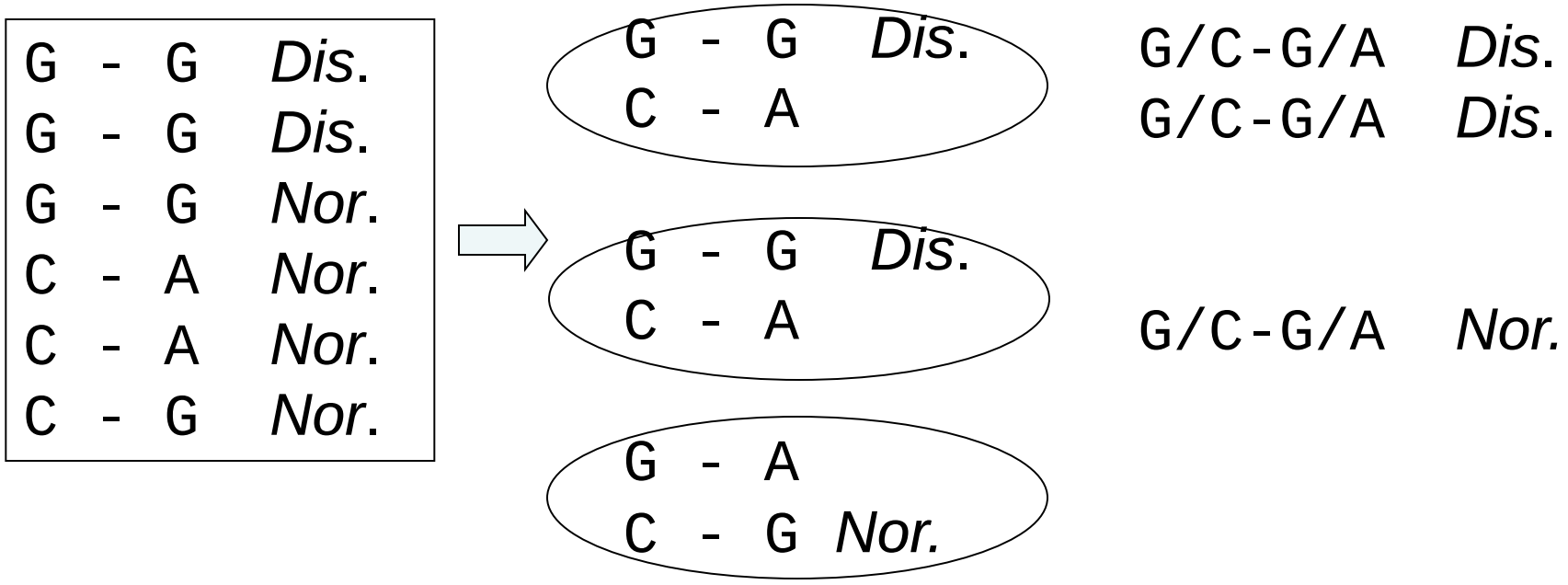
- A haplotype refers to the combination of allelic values on a single chromosome
- Without the causal SNP (A ->T), the haplotype G-G correlates with occurrence of disease, while other haplotypes do not

Humans are diploid



- Humans have two copies of each chromosome
  - Inherited from mother and father
- Genotyping technologies do not maintain the phase

Genotype disease association



- Diploidy and genotyping reduce the power of association!

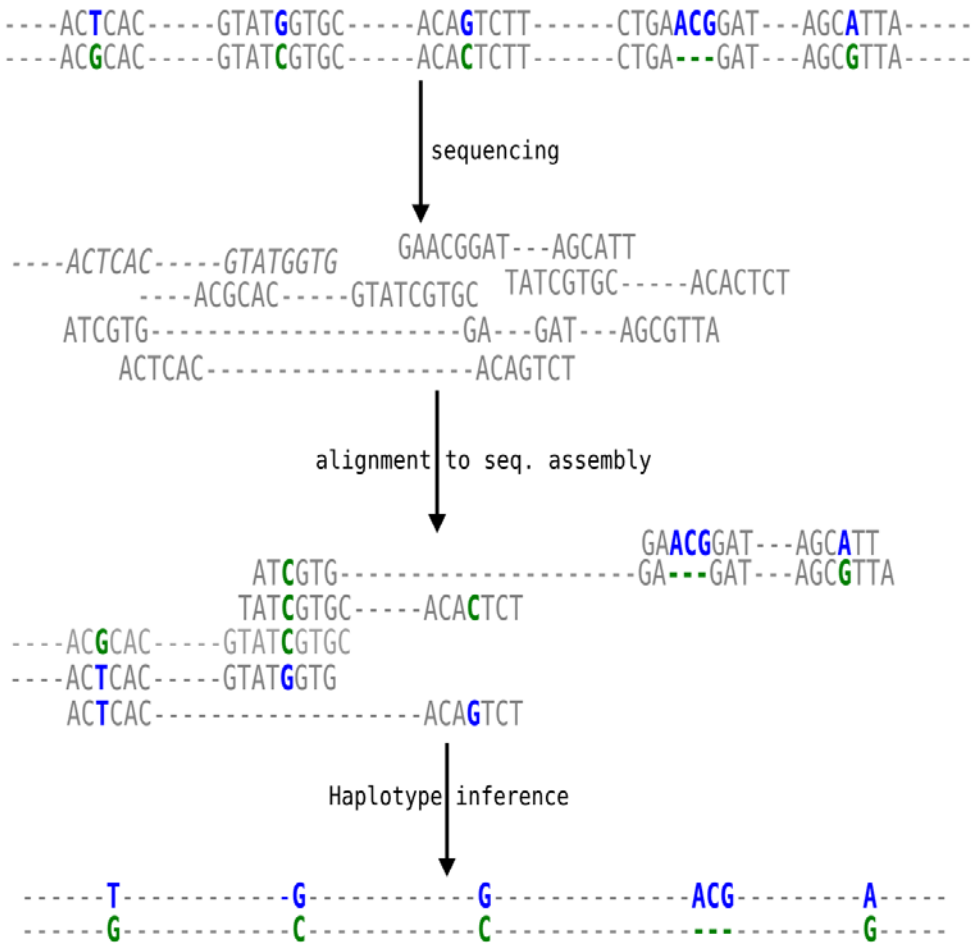
Haplotypes from Genotypes

- The goal of haplotype phasing is to reconstruct the haplotypes from the genotypes
- Haplotypes reconstructed from population genotype data by using correlation between alleles in LD blocks
- Accuracy of haplotypes limited by length of LD blocks (~ 20-50 kb)
- Family genotype data can be used to obtain reliable haplotypes

Reconstucting Haplotypes from sequencing data

- Reads that cover multiple variant sites provide local haplotype information

- Haplotype assembly: use overlap between reads to infer two haplotypes for an individual



Haplotype assembly: Formulation

```
A G A G C T A G C A T G A
C T T T T G G T T C G C G

A G A G - - - - - - - - -
C T T - - - - - - - - - -
- - A G T - - - - - - - -
- - A - - T
- - - G - - - - - - G G -
- - - T C - - - - - - - -
- - - - - T A G - - - - -
- - - - - - A T - A T - -
- - - - - - - G C A - - -
- - - - - - - - - A T G -
- - - - - - - - - - T G A
- - - - - - - - T - - - G
```

- The fragments are aligned to the unphased reference
- Uninformative fragments and columns are removed

Haplotype assembly: Formulation

```
A G A G C T A G C A T G A
C T T T T G G T T C G C G
0 0 0 0 - - - - - - - - -
1 1 1 - - - - - - - - - -
- - 0 0 1 - - - - - - - -
- - 0 - - 0
- - - 0 - - - - - - 1 0 -
- - - 1 0 - - - - - - - -
- - - - - 0 0 0 - - - - -
- - - - - - 0 1 - 0 0 - -
- - - - - - - 0 0 0 - - -
- - - - - - - - - 0 0 0 -
- - - - - - - - - - 0 0 0
- - - - - - - - 1 - - - 1
```

- The fragments are aligned to the unphased reference
- Uninformative fragments and columns are removed
- Relabel the two alleles using 0/1

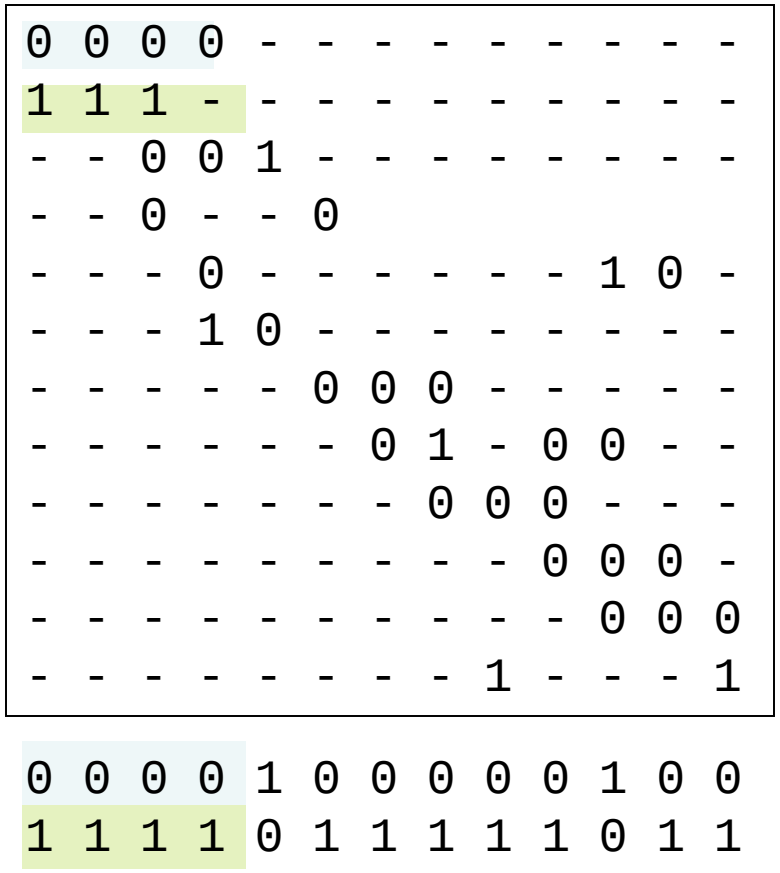
Haplotype assembly: Formulation

```
A G A G C T A G C A T G A
C T T T T G G T T C G C G

0 0 0 0 - - - - - - - - -
1 1 1 - - - - - - - - - -
- - 0 0 1 - - - - - - - -
- - 0 - - 0
- - - 0 - - - - - - 1 0 -
- - - 1 0 - - - - - - - -
- - - - - 0 0 0 - - - - -
- - - - - - 0 1 - 0 0 - -
- - - - - - - 0 0 0 - - -
- - - - - - - - - 0 0 0 -
- - - - - - - - - - 0 0 0
- - - - - - - - 1 - - - 1

0 0 0 0 1 0 0 0 0 0 1 0 0
1 1 1 1 0 1 1 1 1 1 0 1 1
```

- The fragments are aligned to the unphased reference
- Uninformative fragments and columns are removed
- Relabel the two alleles using 0/1
- Goal: Reconstruct the binary string, and its complement, given substrings

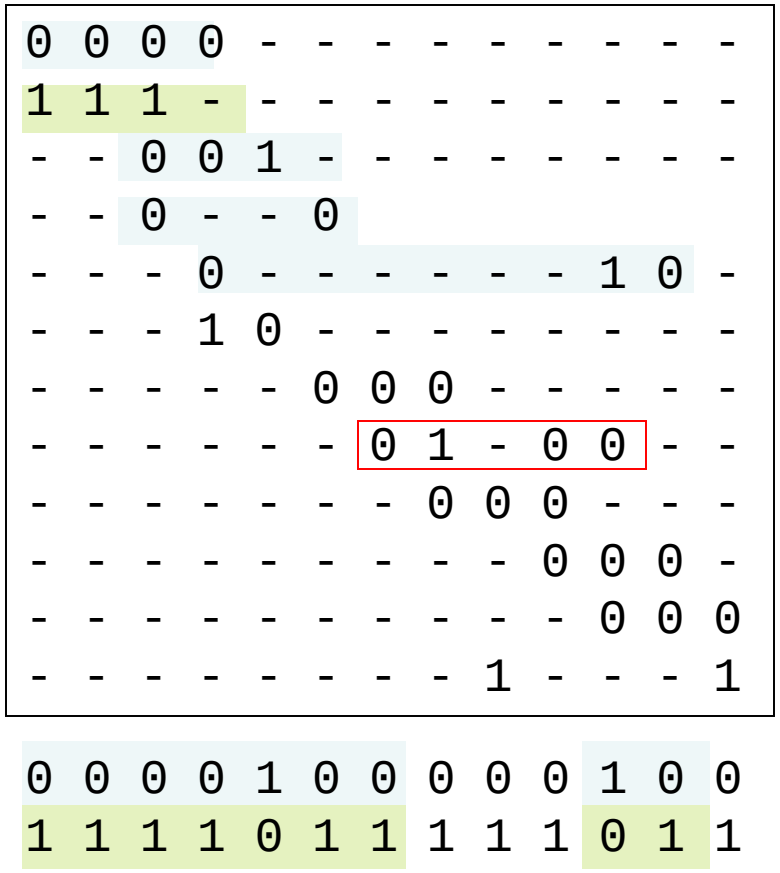
A simple greedy approach

- Greedily select a fragment that extends the current haplotype



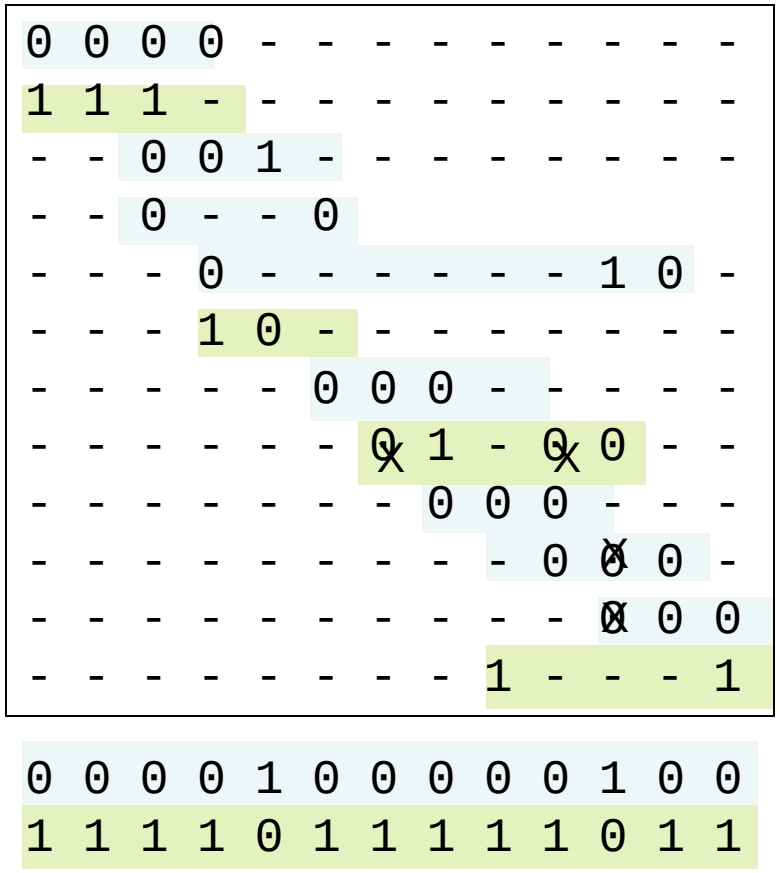
A simple greedy approach

- Some fragments will not match without error
- These are ‘assigned & corrected’ greedily



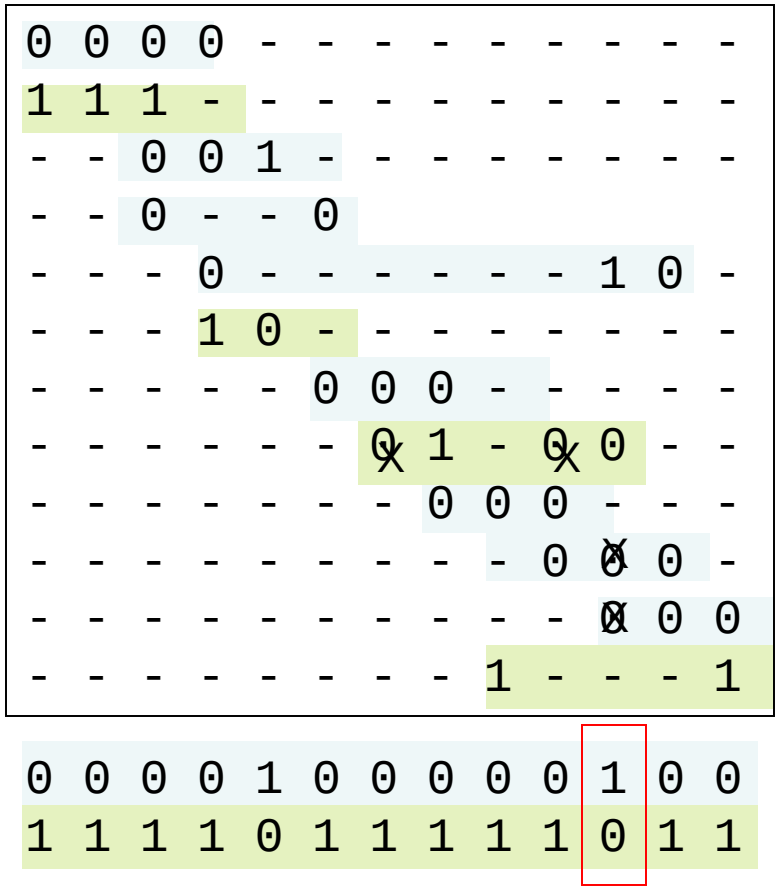
Greedy haplotype assembly

- Minimum Error Correction (MEC): minimum number of variant calls that need to be flipped so that every fragment matches one of the two haplotypes
- MEC is NP-hard



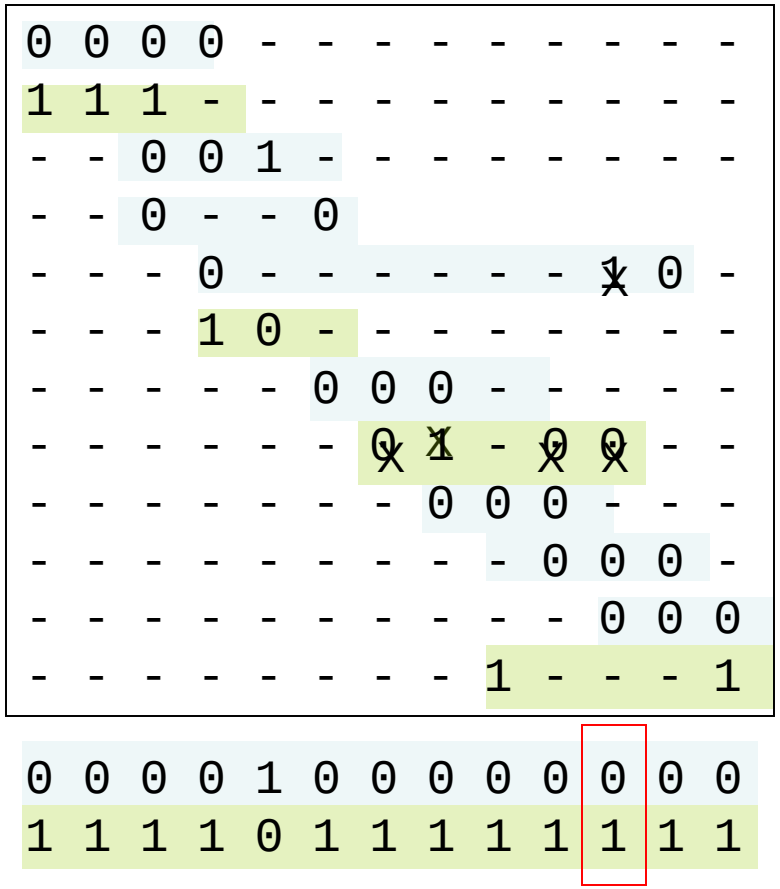
Modifying the haplotypes

- The Greedy approach often leads to suboptimal solutions
- A local flipping of the current haplotype might improve the MEC



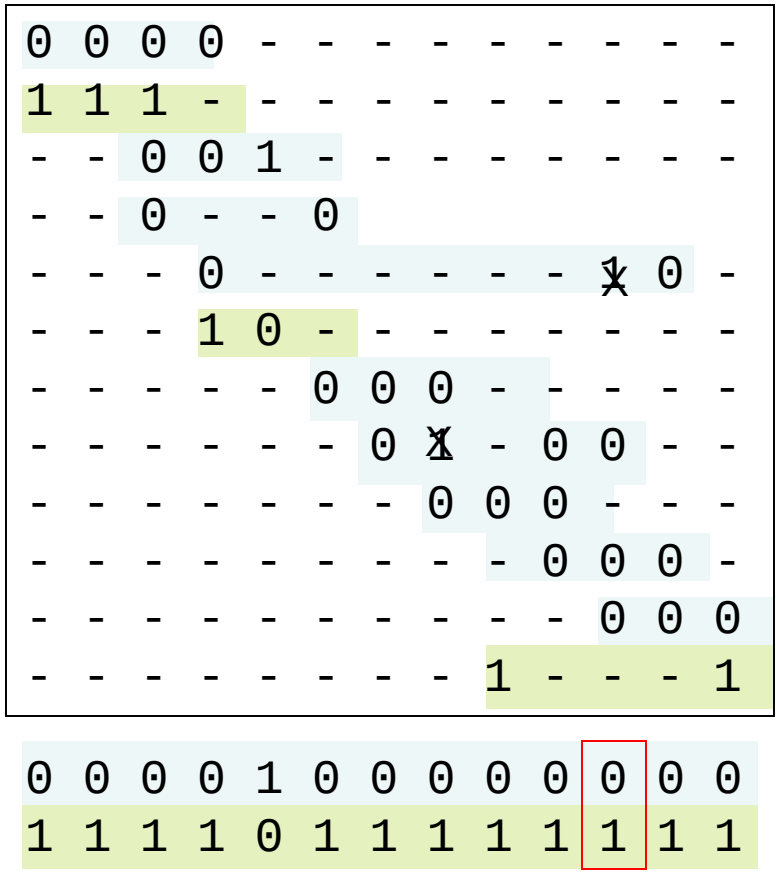
Modifying the haplotypes

- The Greedy approach often leads to suboptimal solutions
- A local flipping of the current haplotype might improve the MEC



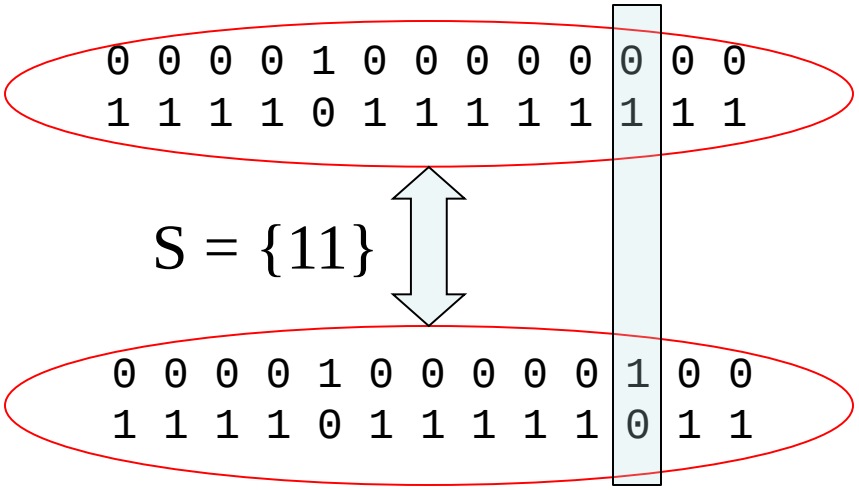
Haplotype to Haplotype

- The haplotype change also involves a reassignment of fragments
- The MEC error reduces to 2
- This suggests a generic strategy
- Start with a haplotype, and move to a new one if it can improve the MEC



Haplotype to Haplotype

- A simple neighborhood is defined by flipping one column at a time (Ex: col. 11)

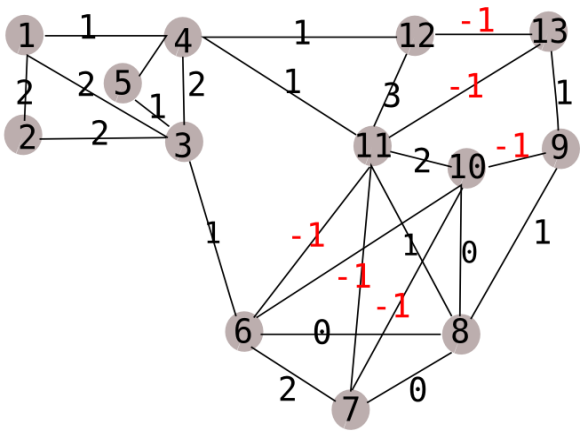


- It is difficult to get out of local minima using single flips
- The “right” move cannot be chosen independently of the fragment matrix and the current solution
- We use the graph structure of the fragment matrix to determine the transitions

Read-haplotype consistency graph

1	2	3	4	5	6	7	8	9	10	11	12	13
A/C	G/T	A/T	G/T	C/T	T/G	A/G	G/T	C/T	A/C	T/G	G/C	A/G
0	0	0	0	-	-	-	-	-	-	-	-	-
1	1	1	-	-	-	-	-	-	-	-	-	-
-	-	0	0	1	-	-	-	-	-	-	-	-
-	-	0	-	-	0	-	-	-	-	-	-	-
-	-	-	0	-	-	-	-	-	-	1	1	-
-	-	-	1	0	-	-	-	-	-	-	-	-
-	-	-	-	-	0	0	0	-	-	-	-	-
-	-	-	-	-	0	0	1	-	0	0	-	-
-	-	-	-	-	-	-	0	0	0	-	-	-
-	-	-	-	-	-	-	-	-	0	0	0	
-	-	-	-	-	-	-	-	-	-	0	0	0
-	-	-	-	-	-	-	-	1	-	-	-	1

0	0	0	0	1	0	0	0	0	1	1	1	0	H
1	1	1	1	0	1	1	1	1	0	0	0	1	

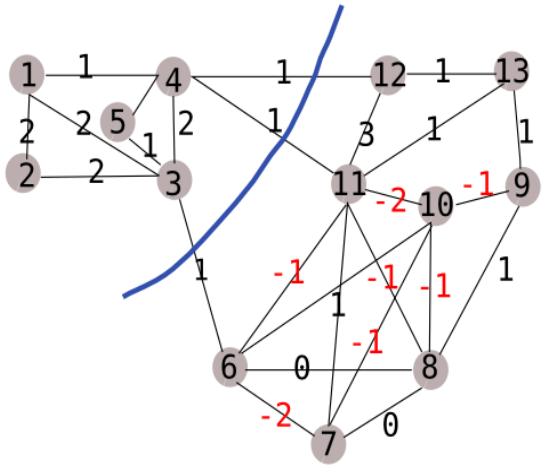


- Each column is a node
- (x,y) is an edge if there is a fragment ‘touching’ columns x and y
- w(i,j) = # fragments matching 'phase' of H - # fragments mismatching 'phase' of H

Cuts

1	2	3	4	5	6	7	8	9	10	11	12	13
A/C	G/T	A/T	G/T	C/T	T/G	A/G	G/T	C/T	A/C	T/G	G/C	A/G
0	0	0	0	-	-	-	-	-	-	-	-	-
1	1	1	-	-	-	-	-	-	-	-	-	-
-	-	0	0	1	-	-	-	-	-	-	-	-
-	-	0	-	-	0	-	-	-	-	-	-	-
-	-	-	0	-	-	-	-	-	-	1	1	-
-	-	-	1	0	-	-	-	-	-	-	-	-
-	-	-	-	-	0	0	0	-	-	-	-	-
-	-	-	-	-	0	0	1	-	0	0	-	-
-	-	-	-	-	-	-	0	0	0	-	-	-
-	-	-	-	-	-	-	-	-	0	0	0	
-	-	-	-	-	-	-	-	-	-	0	0	0
-	-	-	-	-	-	-	-	1	-	-	-	1

0	0	0	0	1	0	1	1	1	0	1	1	1	H
1	1	1	1	0	1	0	0	0	1	0	0	0	

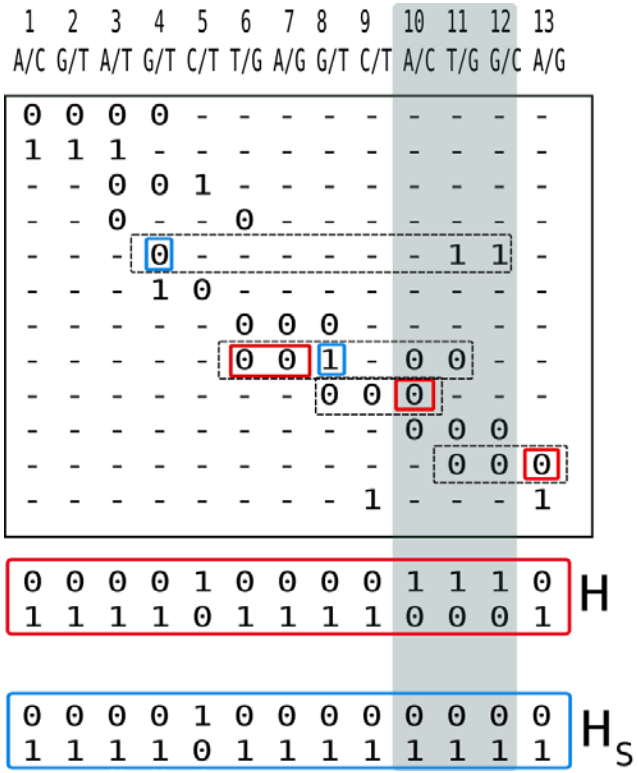


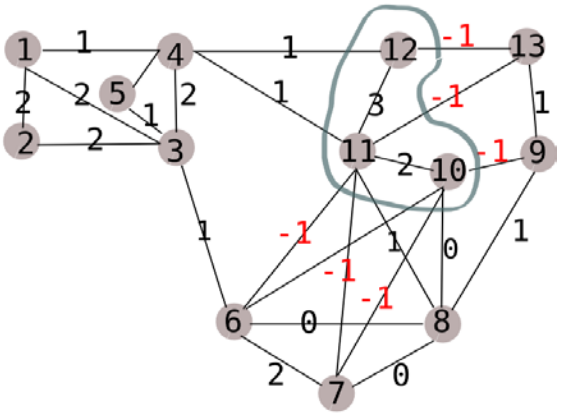
S = { 1,2,3,4,5}

W(S) = 1 + 1 + 1 = 3

- A Cut is a bipartition of the vertices

Negative weighted cuts are ‘good’





- If fragments 'leaving' S are inconsistent with H, the cut S has negative weight
- Such cuts correspond to transitions that reduce the MEC score

A combinatorial scheme

Algorithm HapCUT:

Initialization: Choose an initial haplotype configuration H^1 arbitrarily.

Iteration: For $t = 1, 2, \ldots$

1. Construct the read-haplotype consistency graph $G(H^t)$
2. Compute a cut S in $G(H^t)$ such that $W(S) < 0$
3. If $\text{MEC}(H^t_S) \leq \text{MEC}(H^t)$, $H^{t+1} = H^t_S$
4. Else $H^{t+1} = H^t$

Final: Return H^t

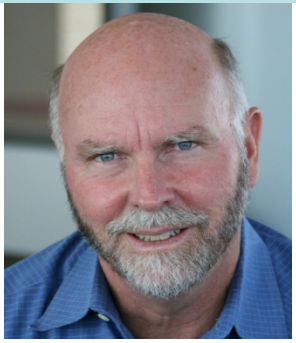
- Cuts computed using a greedy max-cut heuristic
- Stop if no improvement in MEC score for 10 iterations

HapCUT versus sampling

- The HapCUT algorithm uses cut computations in the read-haplotype consistency graph to 'greedily' move towards haplotypes with low MEC
- It can be modified to sample from the haplotype space, instead of searching for haplotypes with lowest MEC

  HASH - “An MCMC algorithm for haplotype assembly from whole-genome sequence data” (Bansal et al. Genome Research, Aug. 2008)

Haplotype assembly for HuRef



OPEN ACCESS Freely available online PLOS BIOLOGY

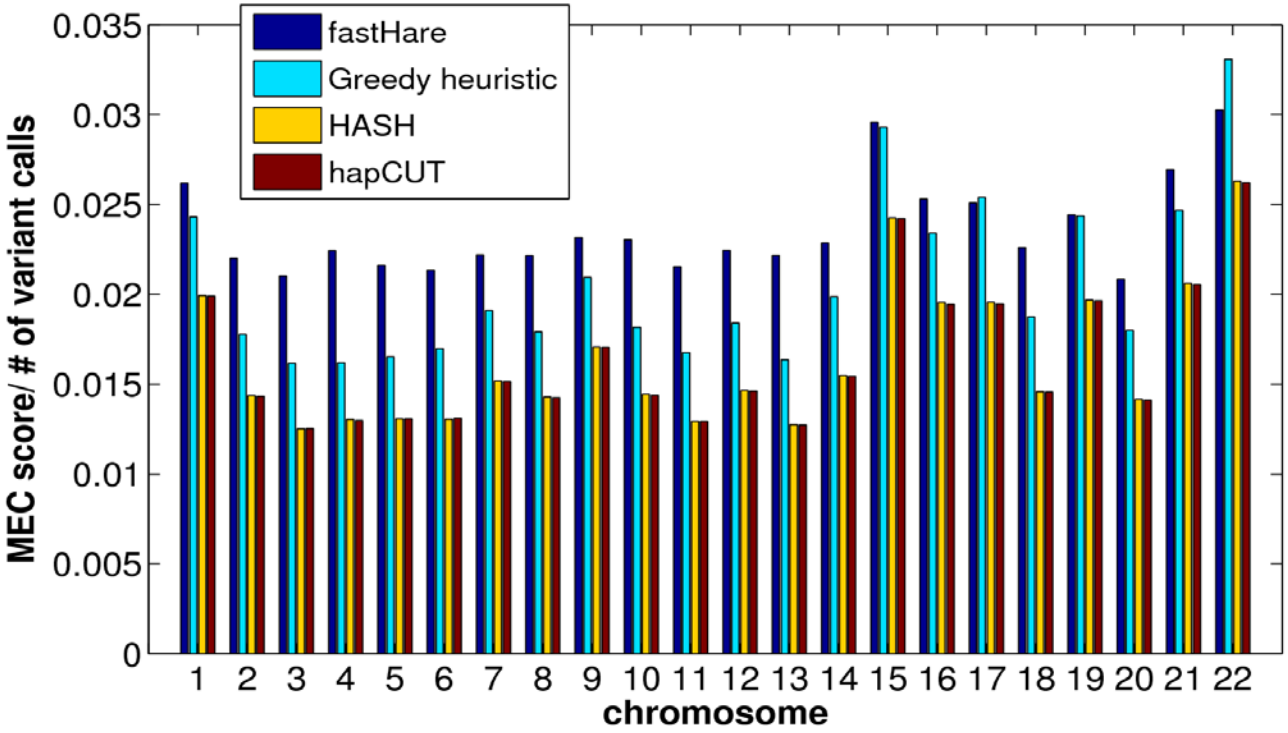
The Diploid Genome Sequence of an Individual Human

Samuel Levy[1*], Granger Sutton[1], Pauline C. Ng[1], Lars Feuk[2], Aaron L. Halpern[1], Brian P. Walenz[1], Nelson Axelrod[1], Jiaqi Huang[1], Ewen F. Kirkness[1], Gennady Denisov[1], Yuan Lin[1], Jeffrey R. MacDonald[2], Andy Wing Chun Pang[2], Mary Shago[2], Timothy B. Stockwell[1], Alexia Tsiamouri[1], Vineet Bafna[3], Vikas Bansal[3], Saul A. Kravitz[1], Dana A. Busam[1], Karen Y. Beeson[1], Tina C. McIntosh[1], Karin A. Remington[1], Josep F. Abril[4], John Gill[1], Jon Borman[1], Yu-Hui Rogers[1], Marvin E. Frazier[1], Stephen W. Scherer[2], Robert L. Strausberg[1], J. Craig Venter[1]

1 J. Craig Venter Institute, Rockville, Maryland, United States of America, 2 Program in Genetics and Genomic Biology, The Hospital for Sick Children, and Molecular and Medical Genetics, University of Toronto, Toronto, Ontario, Canada, 3 Department of Computer Science and Engineering, University of California San Diego, La Jolla, California, United States of America, 4 Genetics Department, Facultat de Biologia, Universitat de Barcelona, Barcelona, Catalonia, Spain

- 1.856M variants used for haplotype assembly of HuRef (Craig Venter's genome)
- Chromosome 22 stats:
  - 25K variant sites, 53K ‘useful’ fragments (rows)
  - ~7 fragments per variant
  - 609 disjoint haplotype blocks (largest contains 1008 variants)
  - 50% of the variant sites lie in haplotypes 350kb or greater (N50 haplotype length)

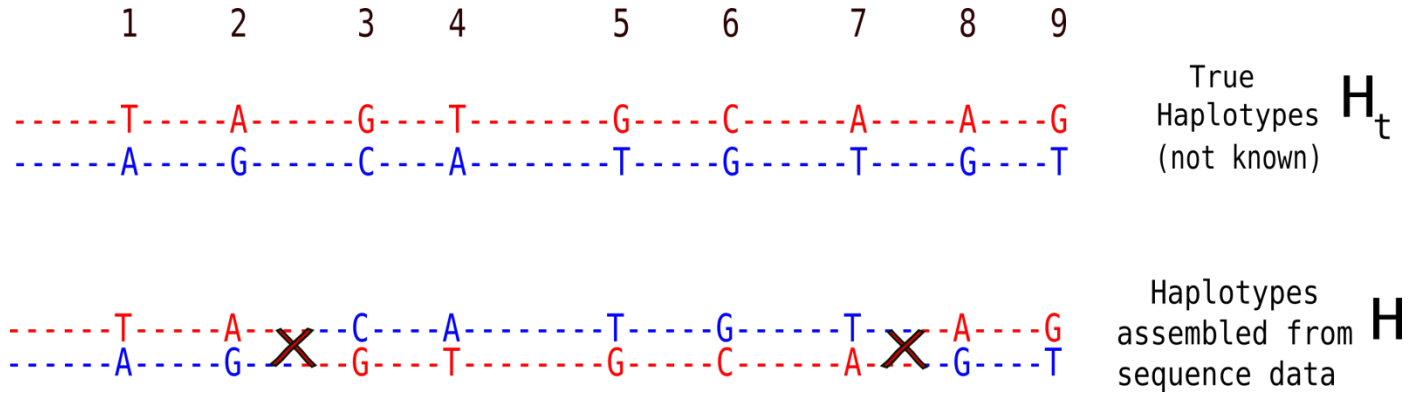
Performance on HuRef



- HapCUT and HASH have nearly identical accuracy
- Both offer > 20% improvement over previous methods
- HapCUT is an order of magnitude faster than HASH

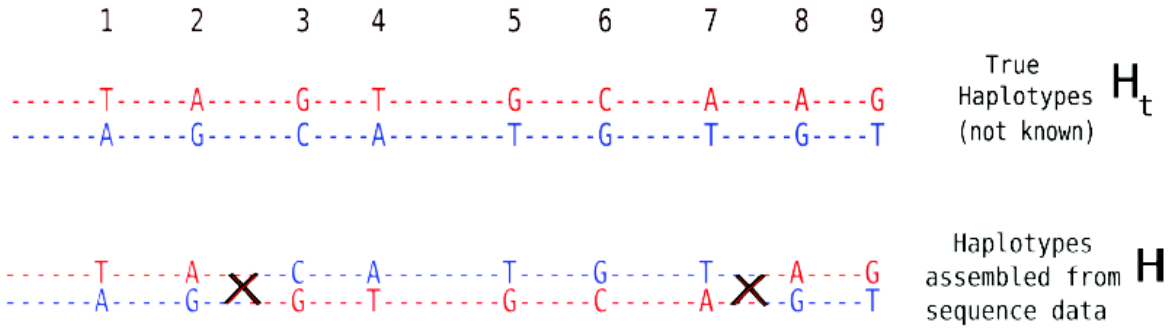
Switch error rate in reconstruction

- MEC error rate measures consistency of haplotypes with the sequenced fragments
- Switch error rate measures absolute accuracy of haplotypes



Switch error rate = 2/8 = 0.25

Using HapMap to estimate switch error rate

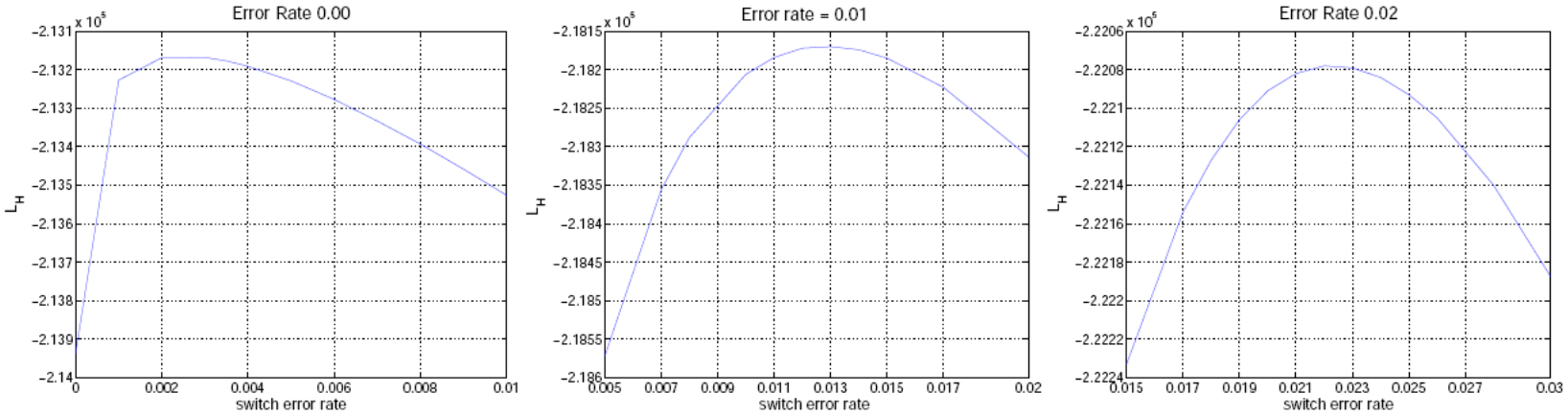


$$L_H(1,2) = (1-\varepsilon_s)\cdot \Pr(H_{12}|H_p) + \varepsilon_s \cdot (1-\Pr(H_{12}|H_p))$$

- For a pair of adjacent SNPs, define a likelihood for the haplotype assembly 'H' conditional on the HapMap haplotypes 'H_p'
- L_H computed as product of pairwise likelihoods

Using HapMap to estimate switch error rate

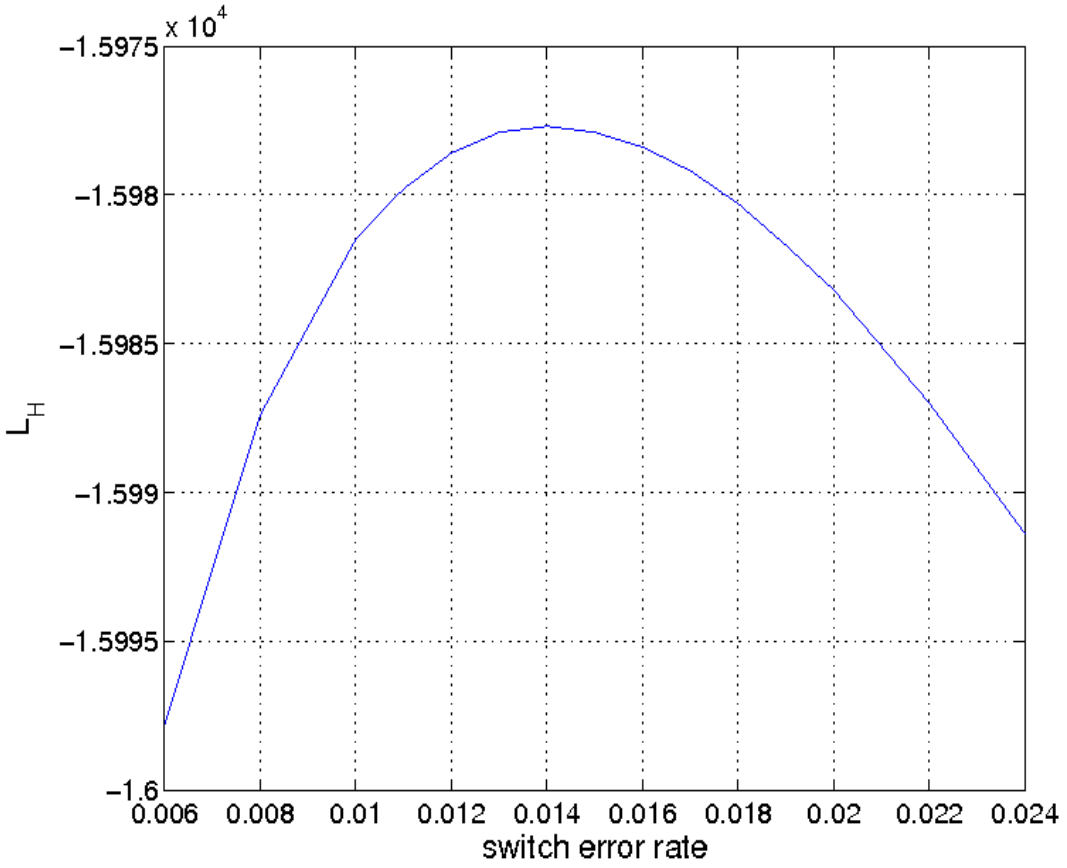
- L_H is a function of Linkage Disequilibrium (LD) in HapMap data and switch error rate ε_s
- *Intuition*: Maximum likelihood value of L_H should track the switch error rate ε_s



- ML estimator works well in simulations (switch errors distributed randomly)

Switch error rate for HuRef haplotypes

- Switch error rate for HapCUT: 0.014
- Switch error rate for greedy heuristic: 0.03
- HapMap switch error rate is 0.005 (CEU) to 0.02 (YRI) even with trios
- Without trios, HapMap error rate is > 0.05



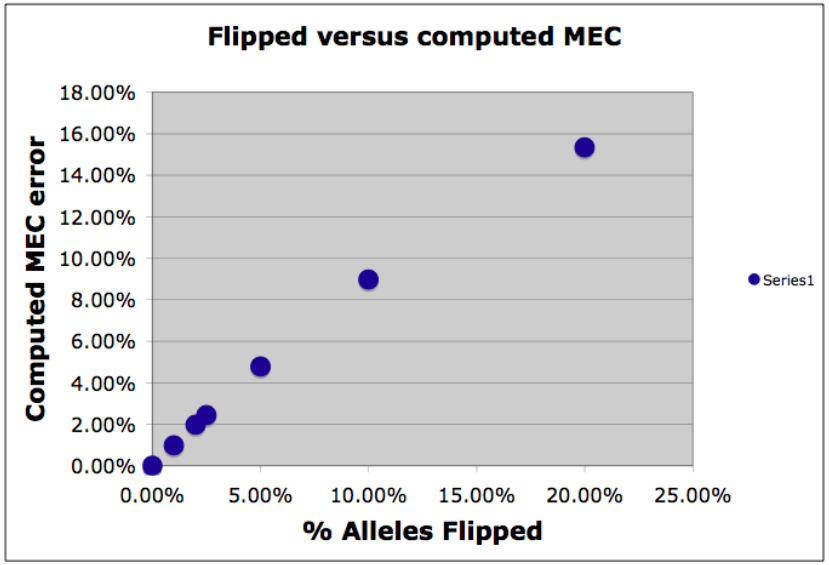
Conclusions

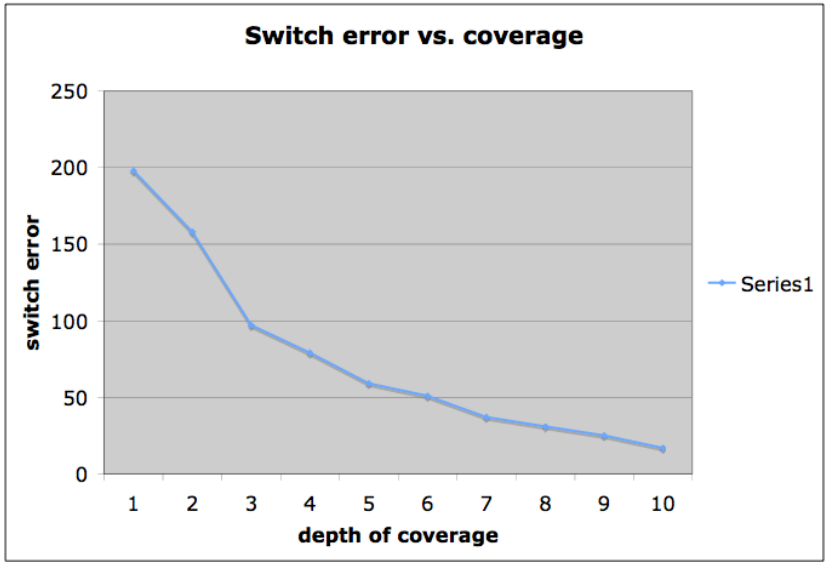
- Haplotype assembly is a feasible approach to haplotype inference, with increasing applicability
- A combinatorial algorithm HapCUT for haplotype assembly with good performance on real data
- Highly accurate haplotypes with low switch error rates based on comparison to HapMap haplotypes

Acknowledgements

- Aaron Halpern
- Sam Levy
- JCV Institute

Simulating errors





- Errors were simulated on HuRef sequences
- The computed MEC error tracks simulated errors
- Switch error in reconstruction is low, and decreases with increasing depth of coverage