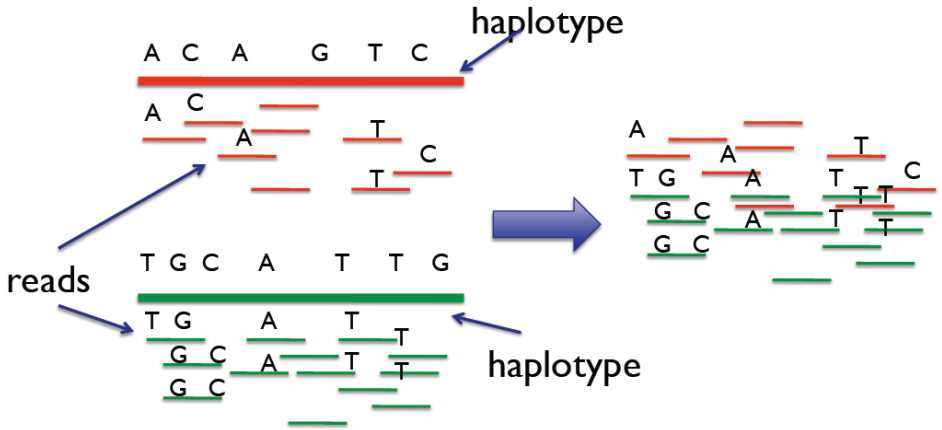
Haplotype Assembly

BY PATRICK TAN

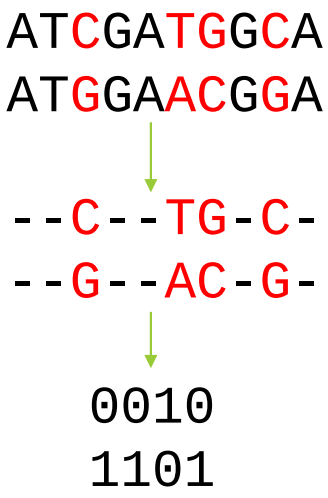
Background and Motivation

- We want to read the haplotypes from a part of the genome
  - Haplotypes can be used for association studies, investigating diseases
- We take multiple reads of parts of the haplotype
  - Only read short parts at a time (cheaper)



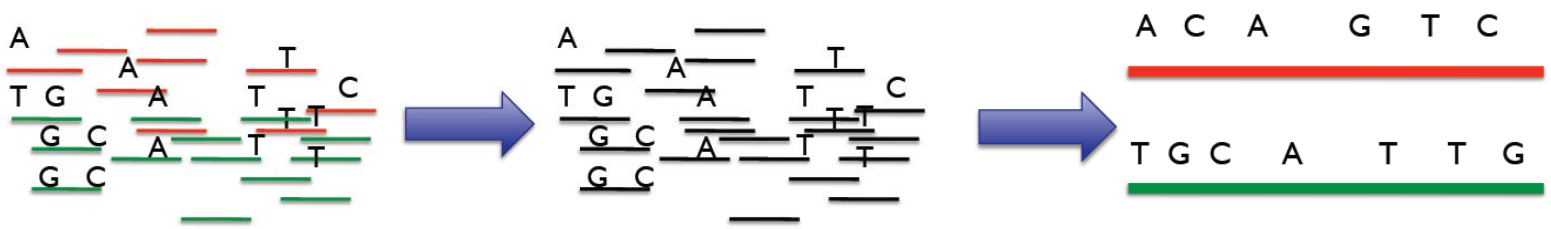
Example Read

- Only need to keep heterozygous SNPs
- We can represent these SNPs as binary strings of 0s and 1s



The Problem

- However, there are two haplotypes per chromosome and each reads can only be taken on one haplotype
- We don't know which reads correspond to which haplotypes
- We want to tile the reads together to recreate the two haplotypes



Computational Problem

- Input: Given n reads for m SNPs (n x m matrix), where each row is a read from one of the two haplotypes
- Output: Two complementary haplotypes of lengths m SNPs that match the reads given in the input.
- Benchmarks
  - Time: Speed of algorithm
    - Expect O(n*2^m) for brute force and O(n) for greedy
  - Accuracy: Errors are counted by the number of switches made between the solution haplotypes and the original haplotypes
    - Since we have no sequencing errors, errors will be very uncommon

Haplotype Example Read Matrix (Input)

Haplotype1	0	0	1	1	0	1	0	0	1
Haplotype2	1	1	0	1	0	1	0	0	1

reads	0	1	2	3	4	5	6	7	8
read1	0	0	-	-	-	-	-	-	-
read2	1	1	-	-	-	-	-	-	-
read3	-	0	1	1	-	-	-	-	-
read4	-	-	0	0	1	-	-	-	-
read5	-	-	-	1	0	-	-	-	-
read6	-	-	-	-	0	1	0	-	-
read7	-	-	-	-	1	0	-	-	-
read8	-	-	-	-	-	1	0	0	-
read9	-	-	-	-	-	-	1	1	0
read10	-	-	-	-	-	-	-	0	1

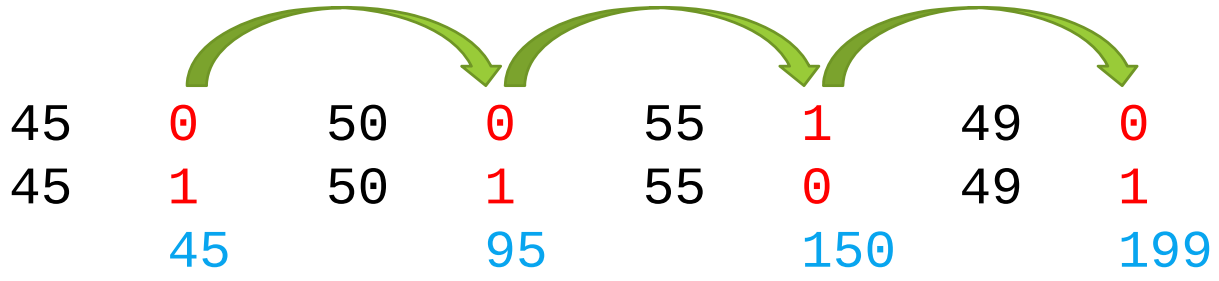
Haplotype Example Read Matrix (Input)

```
Haplotype1    0    0    1    1    0    1    0    0    1
Haplotype2    1    1    0    1    0    1    0    0    1
```

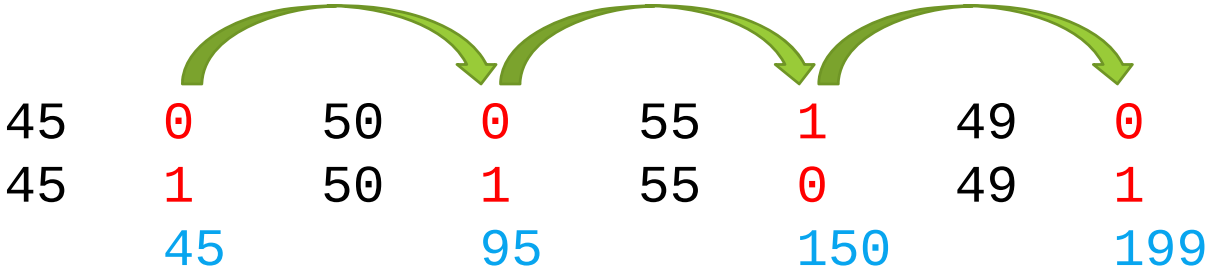
reads	0	1	2	3	4	5	6	7	8
read1	0	0	-	-	-	-	-	-	-
read2	1	1	-	-	-	-	-	-	-
read3	-	0	1	1	-	-	-	-	-
read4	-	-	0	0	1	-	-	-	-
read5	-	-	-	1	0	-	-	-	-
read6	-	-	-	-	0	1	0	-	-
read7	-	-	-	-	1	0	-	-	-
read8	-	-	-	-	-	1	0	0	-
read9	-	-	-	-	-	-	1	1	0
read10	-	-	-	-	-	-	-	0	1

Creating the Read Matrix (Simulation)

- We first generate a haplotype with simulated random distances between each SNP
  - We set a probability that each SNP will be heterozygous, and run geometric distributions for the number of homogenous SNPs until the heterozygous SNP.
  - For each heterozygous SNP, we then randomly assign 0 or 1.
- Then, we create the complement haplotype by switching 0s and 1s (keep the same location distances)



- Reads are taken by choosing random locations in the SNP and reading SNPs within a given read length.
- These reads are stored into the read matrix (read matrix does not need to store location anymore, just needs to store what number SNP it is)



- For example, a read from 60 to 160 would return either:

-01-
-10-

Baseline Solution: Brute Force (Exhaustive)

- Try every possible combination of 0s and 1s of length n SNPs
  - We do not take into account sequencing errors, so it is almost always possible to get 0 errors
  - Therefore, just stop once a combination is found that satisfies all reads.
- Given that: n is the number of reads and m is the number of SNPs
  - There are 2^m possible combinations
  - Each combination can be tested against up to all n reads
  - Time Complexity is therefore O(n*2^m)

My Solution: Greedy

- Loop from first SNP read to last SNP read (must sort the SNPs in order)
- First read can be assigned to either haplotype
- From there, if the read overlaps with known haplotype, assign each read to its corresponding haplotype (match overlaps)
- If unknown (no overlaps), guess
  - This is the only cause of error since we do not have read errors
  - Generally, this only occurs when a distance between reads is longer than the read length
  - Note: it can also occur if we simply do not take enough reads

Example Run

```
Haplotype1    ?    ?    ?    ?    ?    ?    ?
Haplotype2    ?    ?    ?    ?    ?    ?    ?
```

reads	0	1	2	3	4	5	6
read1	0	0	-	-	-	-	-
read2	-	0	1	1	-	-	-
read3	-	-	0	0	1	-	-
read4	-	-	-	-	0	1	0

read1

Haplotype1	0	0	?	?	?	?	?
Haplotype2	1	1	?	?	?	?	?

reads	0	1	2	3	4	5	6
read1	0	0	-	-	-	-	-
read2	-	0	1	1	-	-	-
read3	-	-	0	0	1	-	-
read4	-	-	-	-	0	1	0

read2

```
Haplotype1    0    0    1    1    ?    ?    ?
Haplotype2    1    1    0    0    ?    ?    ?
```

reads	0	1	2	3	4	5	6
read1	0	0	-	-	-	-	-
read2	-	0	1	1	-	-	-
read3	-	-	0	0	1	-	-
read4	-	-	-	-	0	1	0

read3

```
Haplotype1    0    0    1    1    0    ?    ?
Haplotype2    1    1    0    0    1    ?    ?
```

reads	0	1	2	3	4	5	6
read1	0	0	-	-	-	-	-
read2	-	0	1	1	-	-	-
read3	-	-	0	0	1	-	-
read4	-	-	-	-	0	1	0

read4

Haplotype1	0	0	1	1	0	1	0
Haplotype2	1	1	0	0	1	0	1

reads	0	1	2	3	4	5	6
read1	0	0	-	-	-	-	-
read2	-	0	1	1	-	-	-
read3	-	-	0	0	1	-	-
read4	-	-	-	-	0	1	0

Errors

```
Haplotype1    0    ?    ?    ?    ?    ?    ?
Haplotype2    0    ?    ?    ?    ?    ?    ?
```

reads	0	1	2	3	4	5	6
read1	0	-	-	-	-	-	-
read2	-	0	1	1	-	-	-
read3	-	-	0	0	1	-	-
read4	-	-	-	-	0	1	0

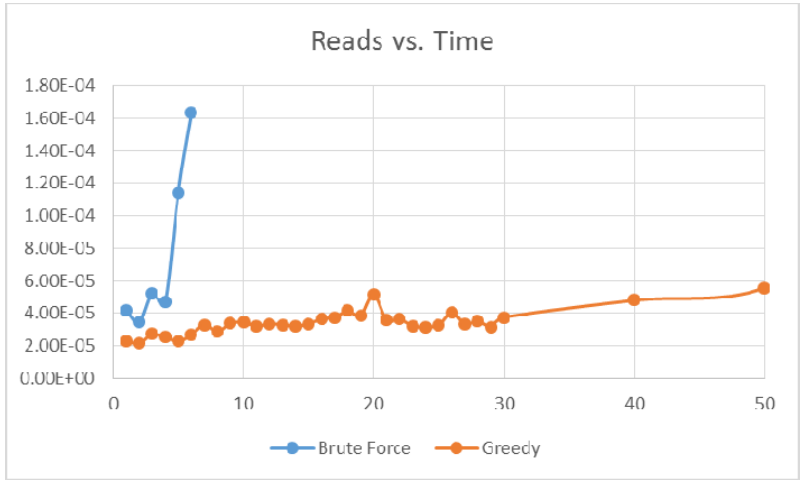
Errors

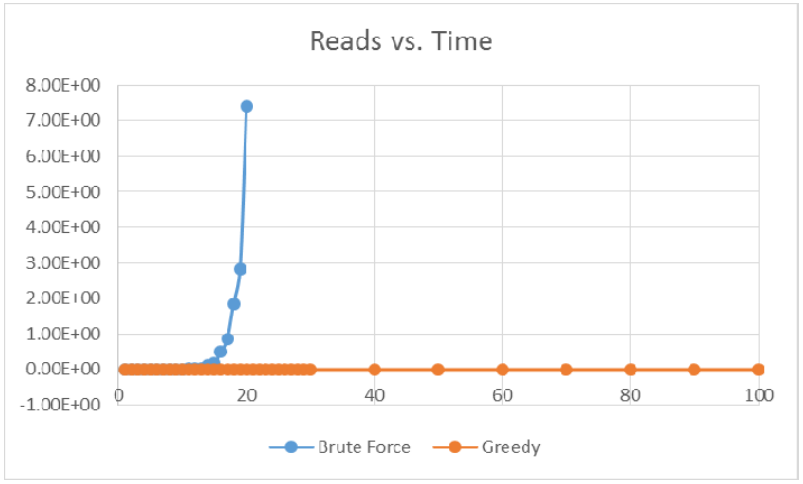
```
Haplotype1    0    ?    ?    ?    ?    ?    ?
Haplotype2    0    ?    ?    ?    ?    ?    ?
```

reads	0	1	2	3	4	5	6
read1	0	-	-	-	-	-	-
read2	-	0	1	1	-	-	-
read3	-	-	0	0	1	-	-
read4	-	-	-	-	0	1	0

Performance

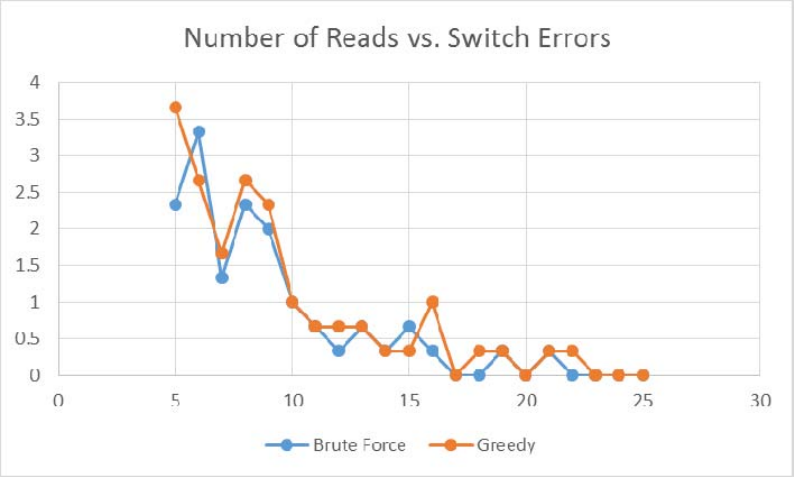
For 20 SNPs, 5% chance of heterozygous, read length of 100 (3 trials each)





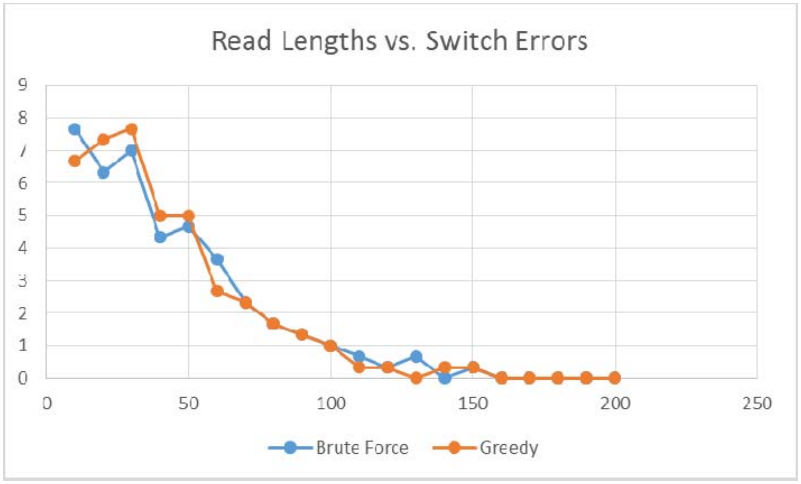
Error Performance

- For 20 SNPs, 5% chance of heterozygous, read length of 100 (3 trials each)



Error Performance

- For 20 SNPs, 5% chance of heterozygous, 15 reads (3 trials each)



Results and Observations

- Both solutions have the same problems with errors (SNPs with no overlaps)
- However, the greedy solution speed scales much better as the number of reads increases
  - O(n) vs O(n*2^m)
  - Our greedy solution takes < 1 second up to approximately 750,000 reads
- Accuracy depends on number of reads, read length (and heterozygous chance)
  - If the distance between SNPs is small enough (high chance of any SNP being heterozygous), there will be no errors since all heterozygous SNPs will likely have some overlap
- If any distance between two SNPs is larger than our read length, we will not have overlap at the SNP. Then, we have a 50% chance of guessing the incorrect haplotype, causing a switch error

Questions