Relatedness Estimator

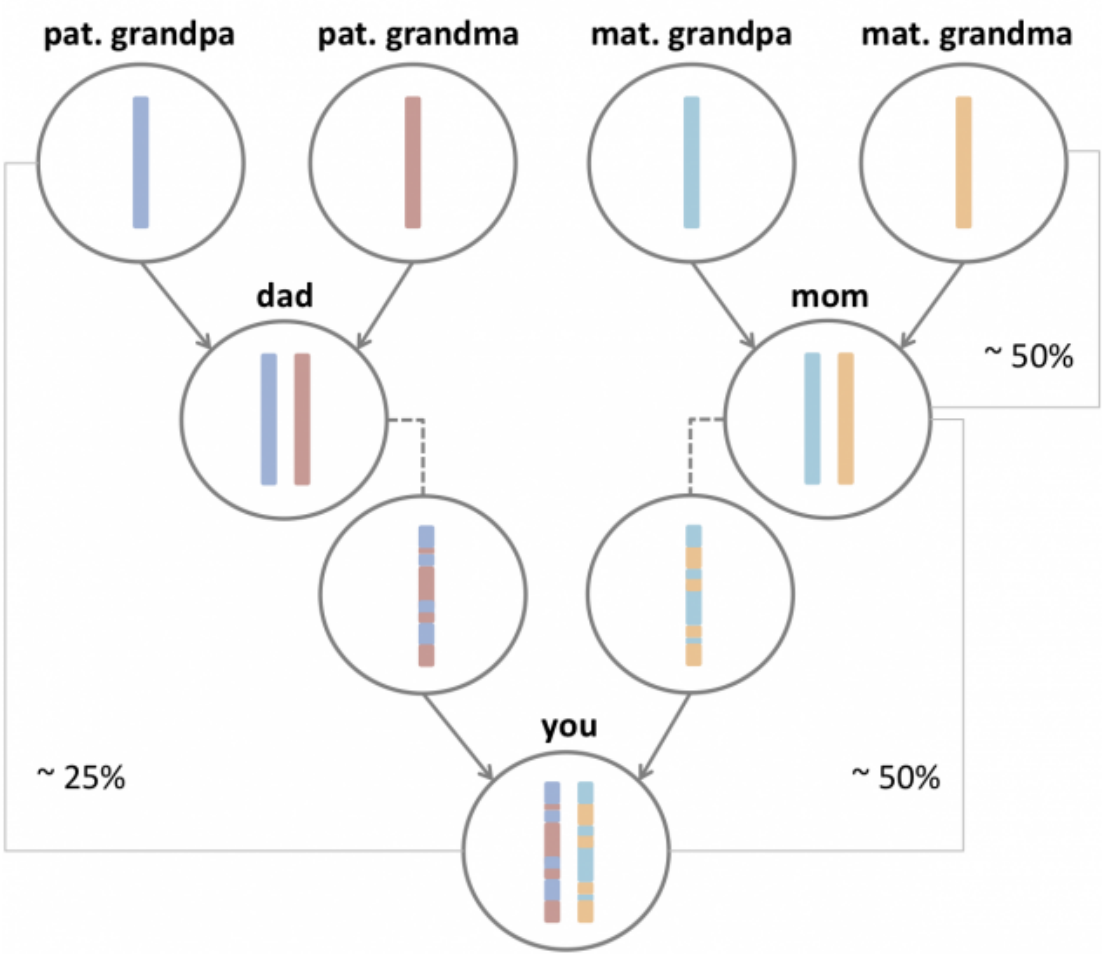
Zheng Sun // UID 204016261

Introduction

- How do we tell if two individuals are related, why do we care?
  - They look alike, same surname, family registry
  - Important for paternity testing, screening for diseases etc.
- What if those factors are uncertain?
- We can turn this into a computational problem and solve it algorithmically

Biological Problem

- The chromosome of an individual is a combination of parents' chromosomes
  - Siblings on average share ~50% DNA
- However, unrelated individuals can also share DNA by chance
- How can we tell if two individuals are siblings based on their DNA?


pat. grandpa
pat. grandma
mat. grandpa
mat. grandma
dad
mom
~ 50%
you
~ 25%
~ 50%

Experiment Setup

- Sample data is a set of genotypes with some related and others unrelated
  - Algorithm doesn’t know which are related, its job is to figure it out
- To evaluate our algorithms, we use the following metrics:
  1. Correctness
  2. Time complexity
  3. Space complexity

Sample Data

- To create sample data:
  a. generate haplotypes for parents with MAF p
     - MAF = Minor Allele Frequency
  b. generate genotypes for children by breeding using haplotypes from each parent
  c. to create siblings, breed two children using same pair of parents
  d. to create unrelated individuals, breed using two different pairs of parents
  e. all genotypes are stored in arrays

Baseline Method

- On average, siblings share ~50% DNA.
- We can do a naive comparison of genotypes
  - Compare each SNP between the two individuals
  - If they share > 50% similarities, then related
  - Else, the individuals are not related
- In our sample data
  - 0 is homogenous minor, 1 is homogenous major, 2 is heterogenous

Examples

A	0	0	1	2	1	0	0	1	1	2
B	0	1	1	2	1	0	0	1	1	0

Number of SNPs = 10

Number of matches = 8

Percentage of matches = 80%

80% > 50% => **A** and **B** are related

Examples

A	0	1	1	2	1	1	0	1	0	1
B	0	0	1	1	1	0	1	1	1	2

Number of SNPs = 10

Number of matches = 4

Percentage of matches = 40%

40% > 50% => **A** and **B** are unrelated

Improved Method

- We can use the MAF to calculate the related and unrelated probabilities for all the haplotype combinations
- This leads to two different 3x3 matrices, one for unrelated probabilities and one for related

Unrelated Matrix

- To calculate unrelated probabilities, we multiply the maf based on parents' 0/1 haplotypes for 0/1/2 haplotypes
- Example
  - $P_u(0, 0)$ => both sets of parents passed 0's => p^4
  - P_u => both A's parents supplied 1's and B's parents either supplied 0 and 1 or 1 and 0 => P(0, 0) = p * p * 2(p * (1-p))

Related Matrix

- To calculate the related matrix, we must consider the different possibilities of how two parents may pass down different haplotypes to children with different genotypes
- Example: $P_r(0, 2)$

	0	1
0	0	2
0	0	2

	0	1
0	0	2
1	2	1

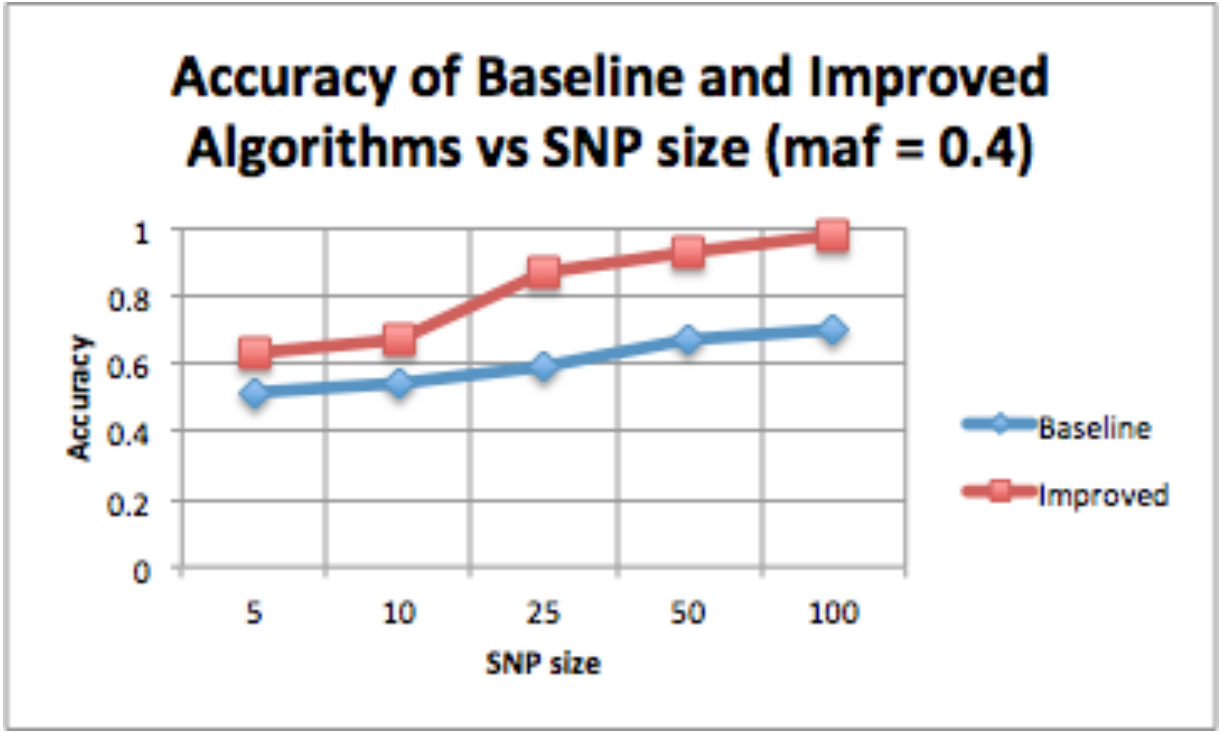
	0	0
0	0	0
1	2	2

$$P_r(0, 2) = 0.5 * 0.5 * P_u(0, 1) + 0.5 * 0.5 * P_u(1, 0) + 0.5 * 0.25 * P_u(0, 2)$$

Improved Method

- Given the unrelated and related matrices, I iterate through the two genotypes
  - I instancialize a variable to keep track of the total probability for related and unrelated cases
  - For each pair of SNPs from the genotypes, I take the unrelated and related probabilities of that pair and multiply the values to my total probability
- Finally, I compare the two total probabilities and use the higher one as my result

Metric: Correctness



Metric: Time Complexity

- For both algorithms, we only need to traverse through the genotype a single time, therefore both are linear time algorithms
- The improved method has the negligible overhead of calculating the two matrices

Metric: Space Complexity

- Assuming the genotype arrays with pass by reference, both algorithms exhibit constant space complexity
- Baseline array only needs to keep track of the tallying sum of the same/different SNPs
- Once again, the improved method has a negligible increase because it must keep track of two 3x3 arrays in addition to the total

Conclusion

- Overall, our improved algorithm exhibits greater accuracy while still keeping linear time and constant space
- Further improvements would be to extend this algorithm more than just siblings (parents, grandparents, etc.)