Relatedness Estimator

Project 3 (Easy)

Andrew Kuang

Motivations

“Why do we care about determining relatedness?”

- ★ Due to the nature of passing down traits (and diseases), knowing one’s relation to others helps determine his/her chances of inheriting certain traits/diseases
- ★ Sometimes **phenotypic** features are not enough to determine relatedness, requires **genetic** analysis.
- ★ Currently, methods such as the **degree of kinship** can be used, but they require knowledge of full family trees in order to accurately determine the coefficient of relationship between two individuals.

Computational Problem

“What are we trying to do?”

- ★ Given two individuals’ SNPs, accurately determine whether or not the two individuals are related (siblings).
- ★ Observe the fluctuations in accuracy of algorithm in relation to SNP size, trial size, and MAF*.

  * MAF = Minor Allele Frequency

  0: minor allele; 1: major allele
- ★ What determines a “good” algorithm?
  1. Speed (how does the algorithm handle a large SNP size/number of trials?)
  2. Accuracy (how well can the algorithm correctly predict the relatedness?)
  3. Memory (how much space does the algorithm use to run trials?)

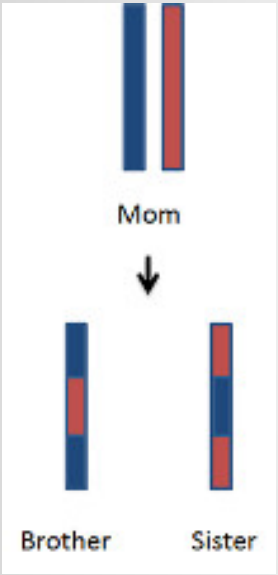
Baseline Method

“A straightforward approach to the computational problem.”

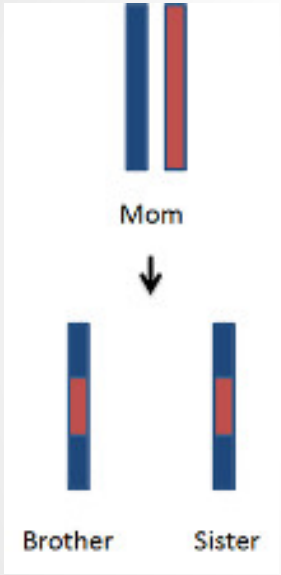
- ★ Idea: The more related two individuals are, the more likely their SNPs are similar.
- ★ Problem: How related are siblings? What percentage of their SNPs should match?
- ★ Everyone is ~50% related to each parent, but can be related anywhere between 0~100% to a sibling

Baseline Method Pt. II

“A straightforward approach to the computational problem.”



0% related



100% related

★ Although siblings’ relatedness can range from 0~100%, the average relatedness falls around 50%

★ As such, we use a cutoff of 50% SNP match for our baseline method.

Baseline Method Pt. III

"A straightforward approach to the computational problem."

SNP_A = 0 1 1 0 2 1 0 2 1 0 1 0 1 1 2 1 0 0

SNP_B = 0 2 0 1 2 0 0 1 1 0 2 0 0 1 2 1 1 0

Number of Matches: 10

SNP Size: 18

SNP Match: 10/18 = 0.5555… = 55.5%

Relatedness Check: 55.55% > 50%?

YES, the two siblings are **related**.

My Method

"My approach to the computational problem."

- ★ Idea: Rather than looking at the two individuals' number of matches (risky for false positives), predict the probability that the parents are the same (aka siblings).
- ★ How to do this?
  - Using **Bayes' Theorem of Total Probability**, we can calculate the likelihood that the parents are the same given the individuals' SNPs.
  - This gives us the probability that the two individuals are related (aka our related matrix).

My Method Pt. II

"My approach to the computational problem."

★ Calculating the unrelated 3x3 matrix is trivial: simply calculate the probabilities through multiplication (Assumption: the probability of a 0 or 1 is independent for every allele).

★ $P_u(0,0) = P(0)*P(0)*P(0)*P(0) = maf^4$
★ $P_u(0,1) = P(0)*P(0)*P(0)*P(1) = maf^3*(1-maf)$
★ $P_u(0,2) = P(0)*P(0)*P(1)*P(1) = maf^2(1-maf)^2$

My Method Pt. III

"My approach to the computational problem."

★ Calculating the related 3x3 matrix is a bit harder, as we need to consider all possible cases. For example:

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

	0	1
0	00	01
0	00	01

	0	1
0	00	01
1	01	11

	0	0
0	00	00
1	01	01

*P(1,0) = P(one sibling is a 1 (alleles: 01/10), one sibling is a 0 (00))

My Method Pt. IV

"My approach to the computational problem."

★ After obtaining the related and unrelated matrices, I iterated through the two SNPs, calculating the probability of (un)relatedness based on every SNP entry

SNP_A = 0 1 1 0 2 1 0 2 1 0 1 0 1 1 2 1 0 0

SNP_B = 0 2 0 1 2 0 0 1 1 0 2 0 0 1 2 1 1 0

P(A, B related) = $P_r(0,0) * P_r(2,2) * P_r(0,0) * P_r(1,1) * P_r(0,0) * P_r(0,0) * \ldots$

P(A, B unrelated) = $P_u(0,0) * P_u(2,2) * P_u(0,0) * P_u(1,1) * P_u(0,0) * P_u(0,0) * \ldots$

Results

“How did the two algorithms compare against each other?”

★ Three criteria: speed, accuracy, and memory* (space).

★ Memory:

Baseline: constant space

- only needs to store the two individuals’ SNPs
- a single counter which tracked # of matches

My Algorithm: constant space

- only needs to store the two individuals’ SNPs
- two doubles that kept track of current running prob. (related/unrelated)

*memory of the algorithm only, not generation of data

Results

“How did the two algorithms compare against each other?”

★ Speed

Baseline:

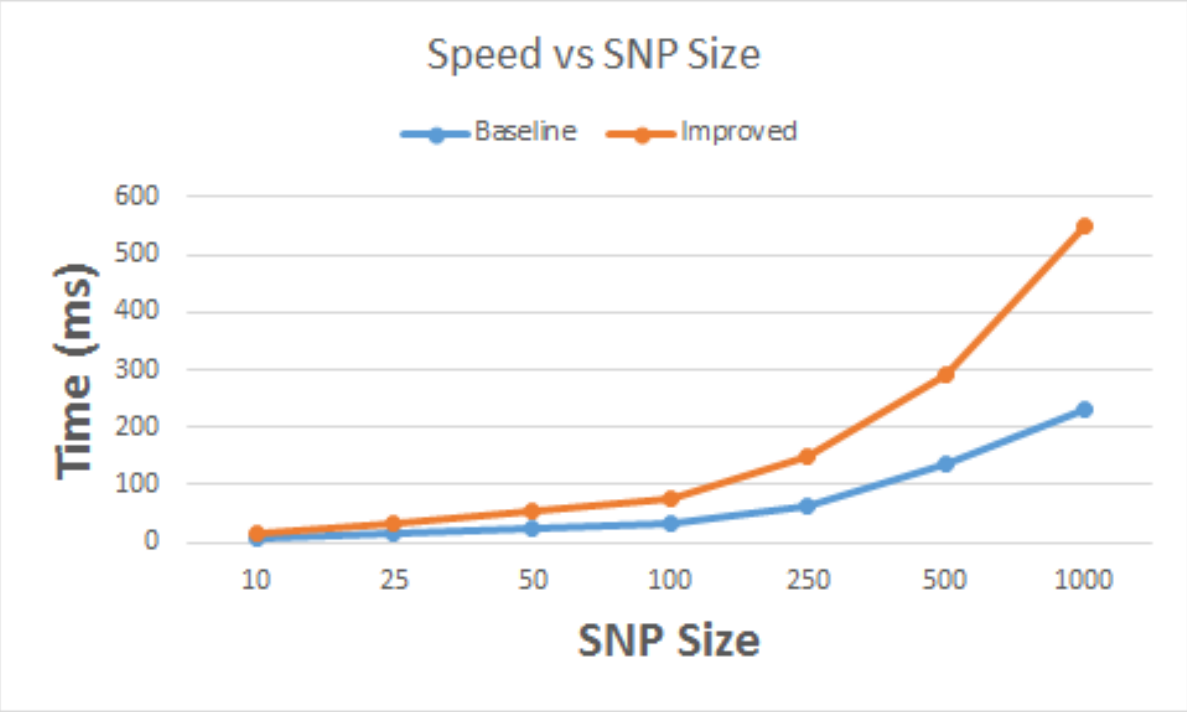
- only needs to iterate through SNPs and compare value at position
- increment counter if match found

My Algorithm:

- create the probability matrices (unrelated and related) given the MAF
- iterate through SNPs and, at every position:
    1. access the unrelated/related matrices
    2. multiply matrix value with running probability

Results

“How did the two algorithms compare against each other?”

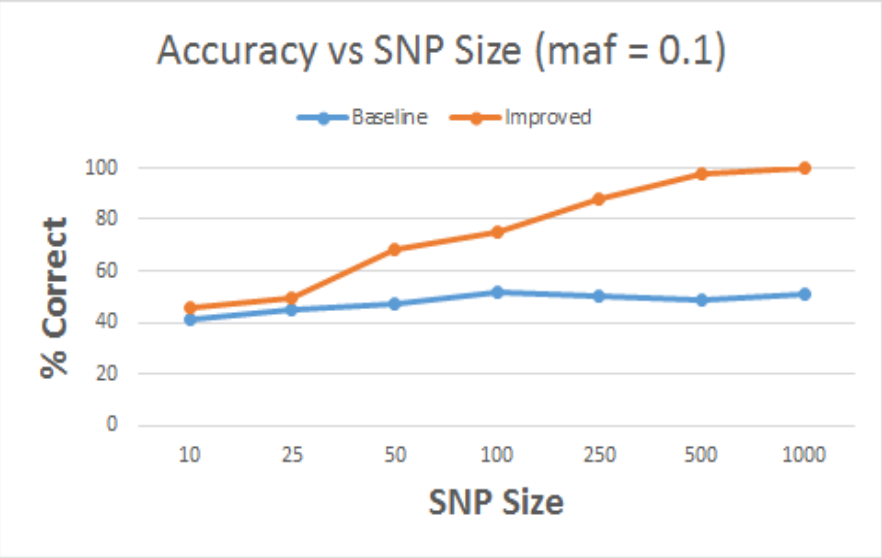


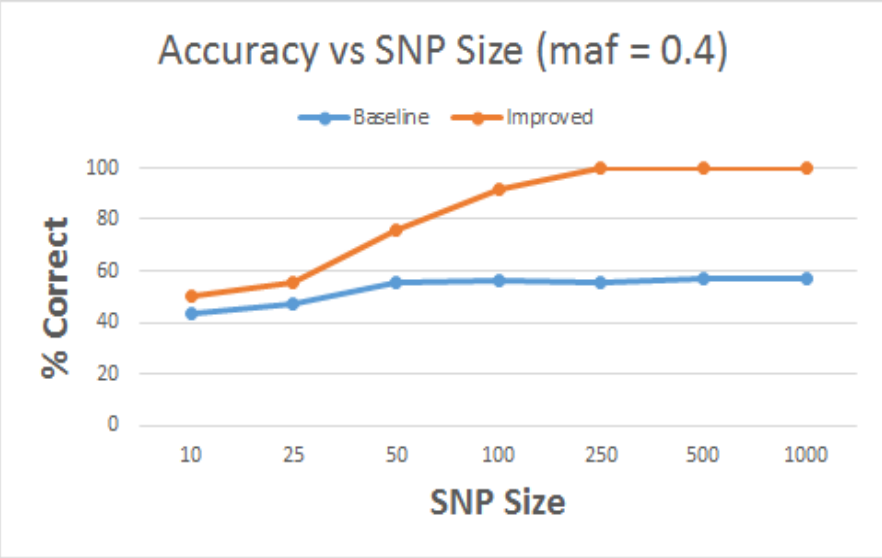
* results taken from the average of 50 simulations

Results

“How did the two algorithms compare against each other?”

★ Accuracy (averaged over 50 Simulations per SNP Size)





Conclusion

“Final thoughts and improvements.”

- Speed-wise and memory-wise, my algorithm is slightly worse (although still linear). Accuracy-wise, my algorithm shows more promise.
- **Improvement:** Try to increase the generation depth more (simulating parents’ parents, etc.) to see if there is a more accurate algorithm that can detect more false positives for lower MAFs.
- **Further Steps:** Study more on how the fluctuation in MAFs can affect the probability of false positives (in baseline vs my algorithm).