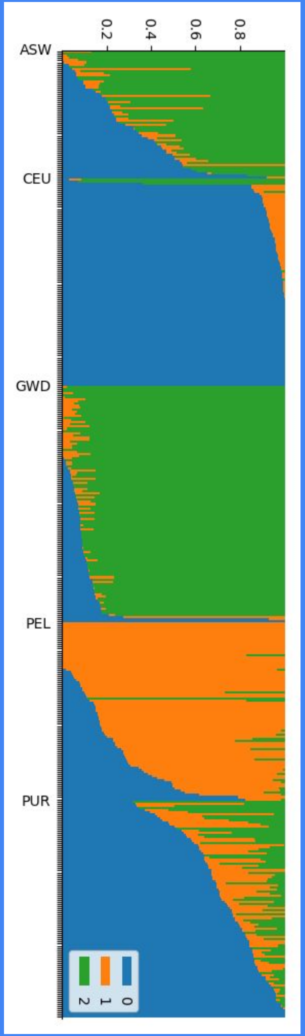
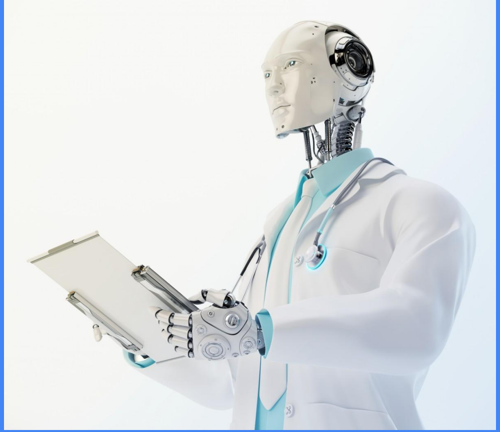
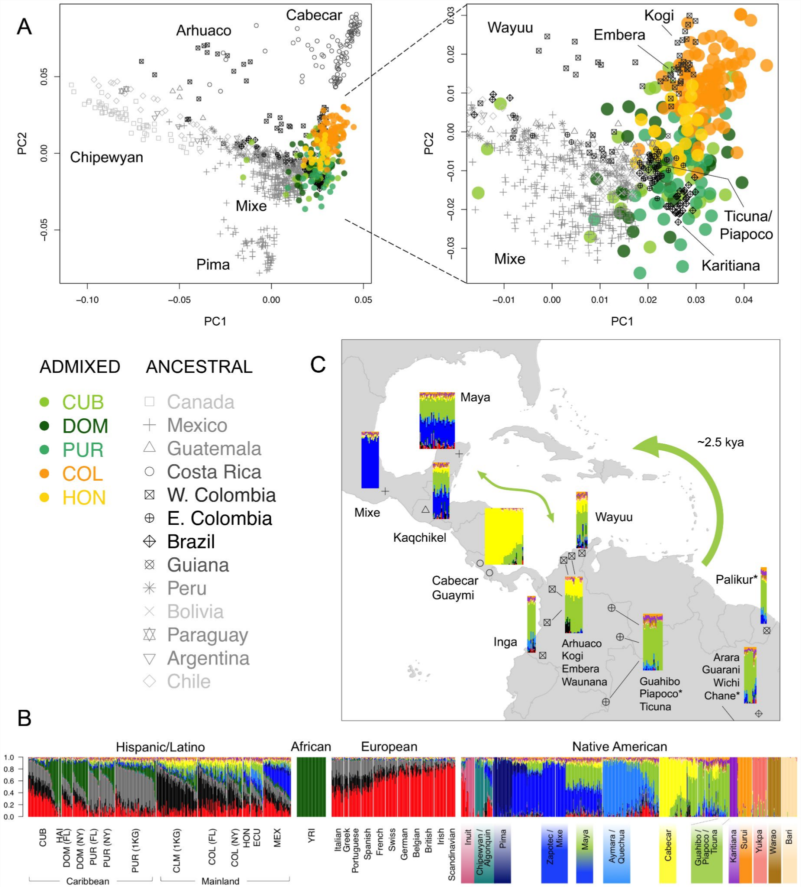
wf-admixture

A global ancestry tool by Willard Ford





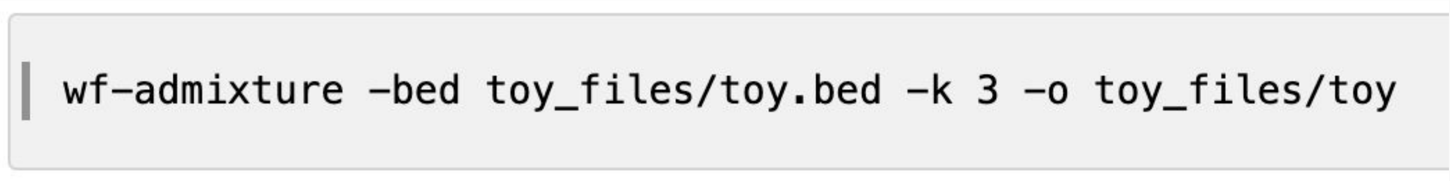


Admixture helps reveals historical migrations

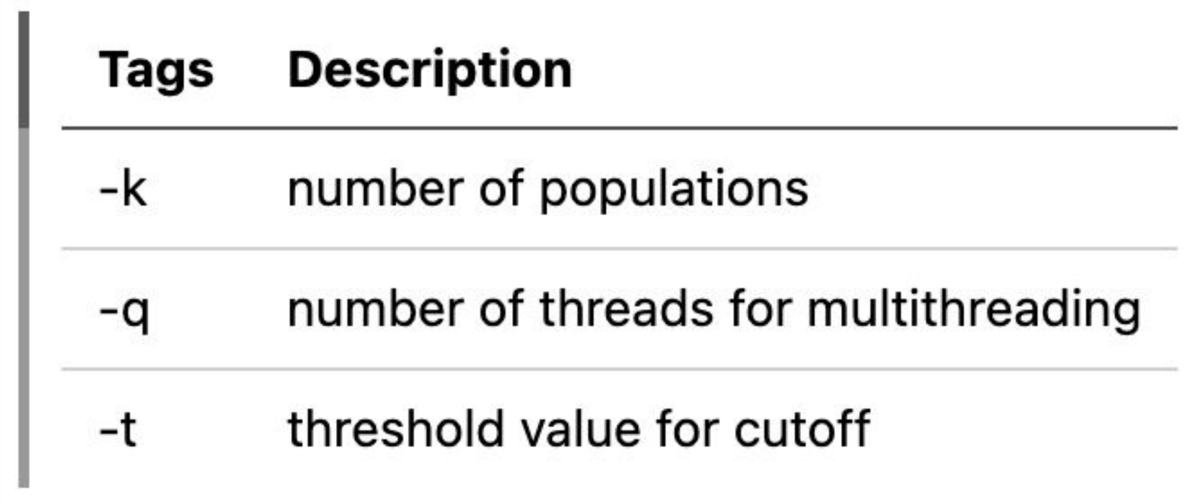
Reference: https://journals.plos.org/plosgenetics/article%3Fid=10.1371%2Fjournal.pgen.1003925

Usage

Input: LD filtered *.bed*, *.bim*, and *.fam* files from plink.



```
wf-admixture -bed toy_files/toy.bed -k 3 -o toy_files/toy
```

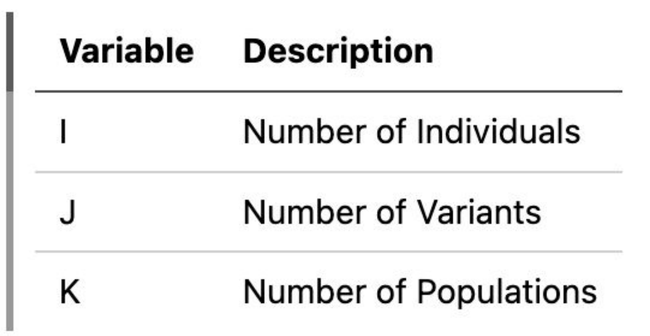


Tags	Description
-k	number of populations
-q	number of threads for multithreading
-t	threshold value for cutoff

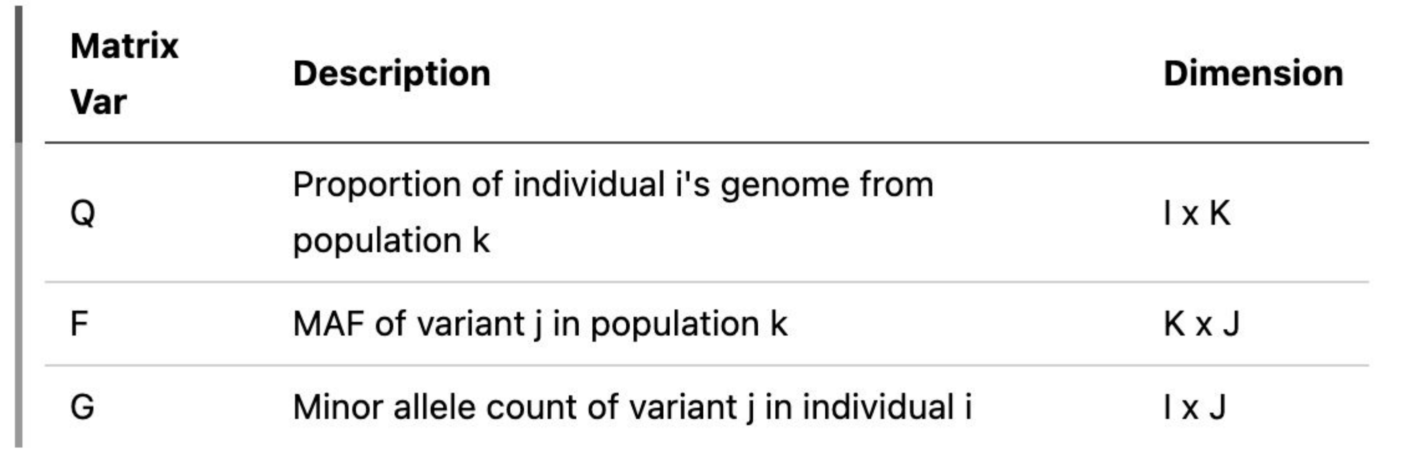
Problem Setup

Goal: Get admixture of each individual

Requires iteratively calculating Q and F



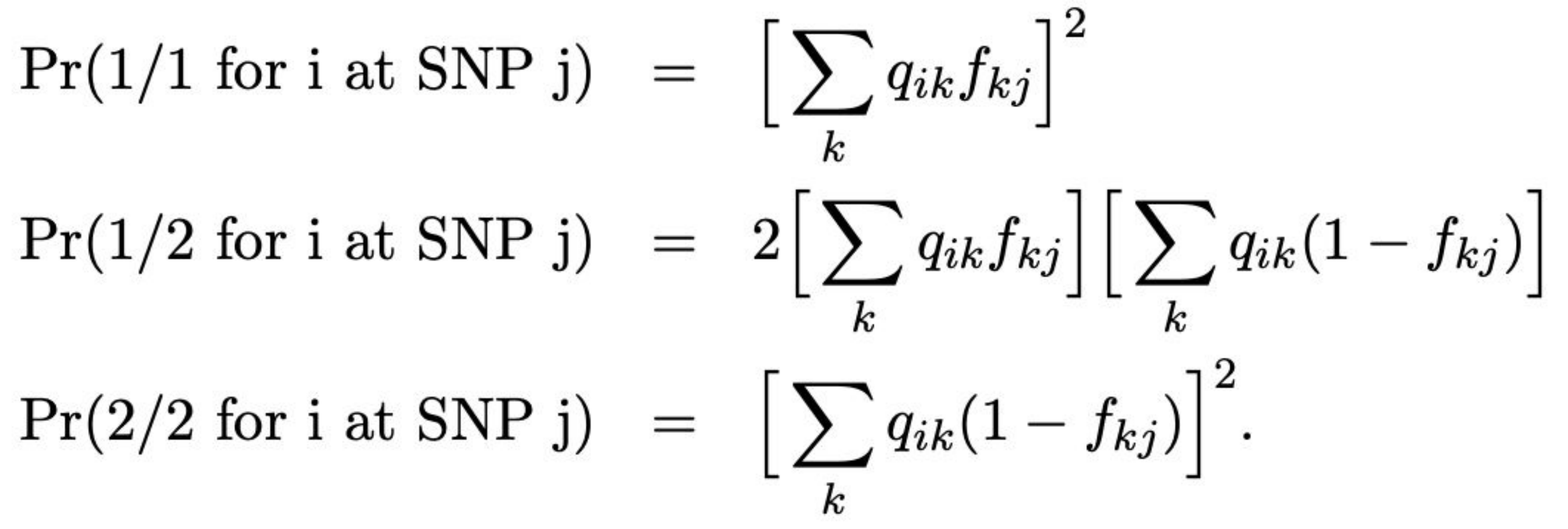
Variable	Description
I	Number of Individuals
J	Number of Variants
K	Number of Populations



Matrix Var	Description	Dimension
Q	Proportion of individual i's genome from population k	I x K
F	MAF of variant j in population k	K x J
G	Minor allele count of variant j in individual i	I x J

Hardy Weinberg Equilibrium

Assume all individuals are independent combinations of the previous generation's gametes.



$$\begin{aligned}
\Pr(1/1 \text{ for i at SNP j}) &= \Big[\sum_k q_{ik} f_{kj}\Big]^2 \\
\Pr(1/2 \text{ for i at SNP j}) &= 2\Big[\sum_k q_{ik} f_{kj}\Big]\Big[\sum_k q_{ik}(1 - f_{kj})\Big] \\
\Pr(2/2 \text{ for i at SNP j}) &= \Big[\sum_k q_{ik}(1 - f_{kj})\Big]^2.
\end{aligned}$$

MLE: Log Likelihood Function

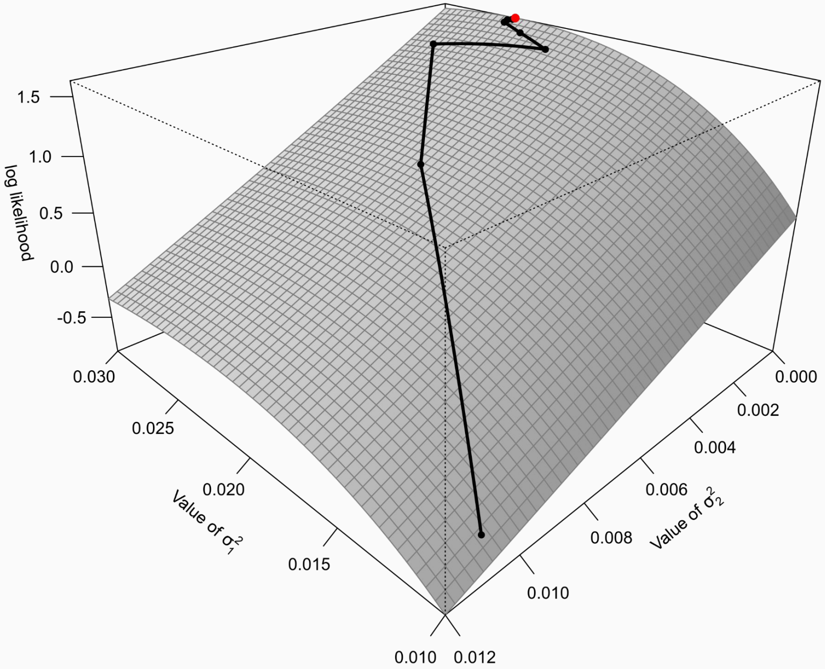
$$L(Q,F) = \sum_i \sum_j \left\{ g_{ij} \ln \left[\sum_k q_{ik} f_{kj} \right] + (2 - g_{ij}) \ln \left[\sum_k q_{ik}(1 - f_{kj}) \right] \right\},$$

Which parameters were most likely to achieve our inputs?

Maximize the Likelihood function.

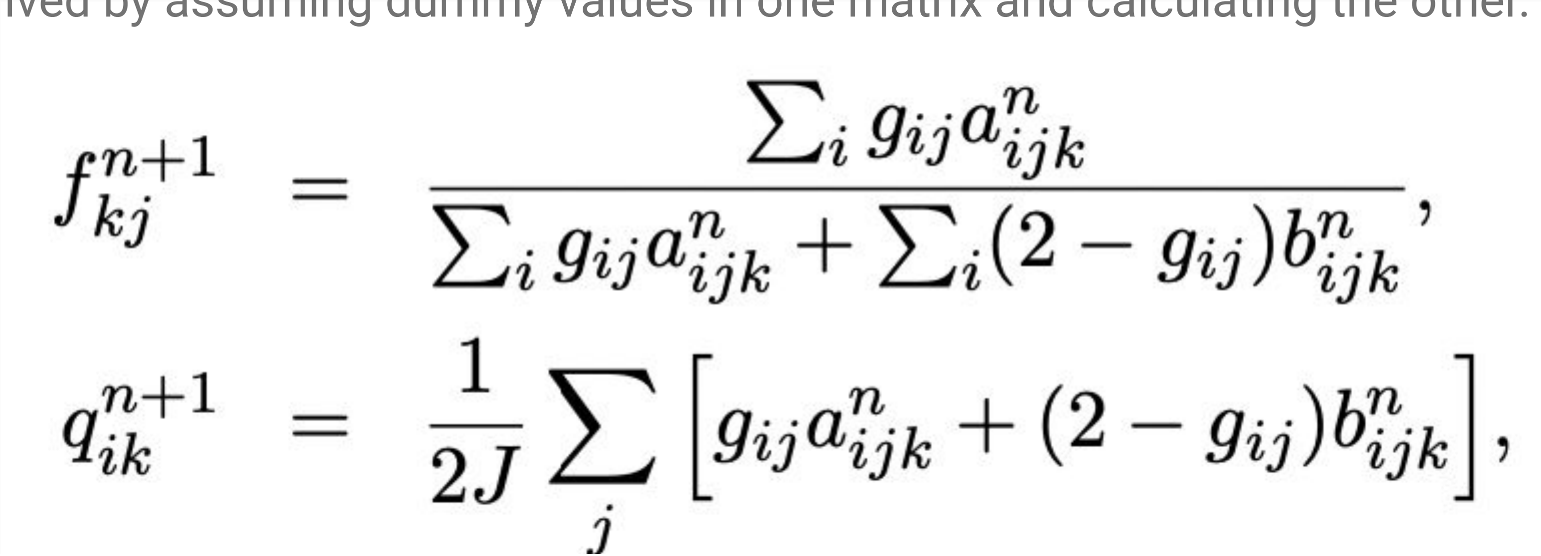
Numerical optimization, no analytical solution.

Stop when subsequent likelihoods are below some threshold.



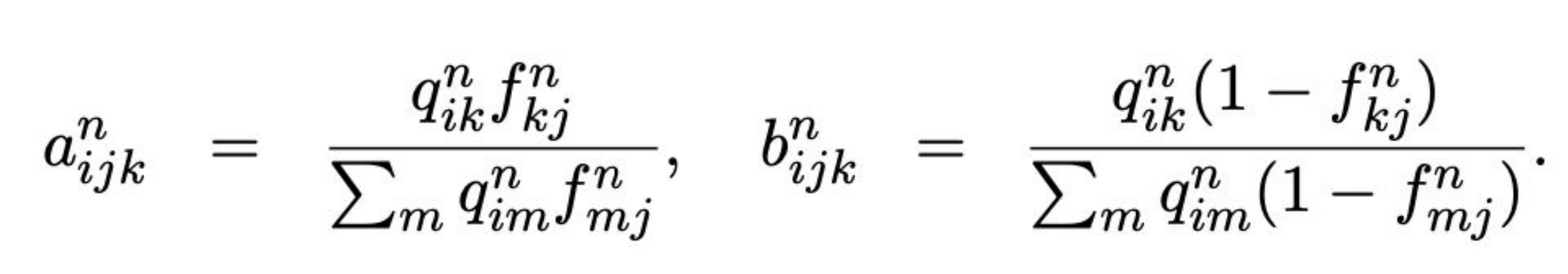
Frappe Expectation Maximization Algorithm

Derived by assuming dummy values in one matrix and calculating the other.



$$f_{kj}^{n+1} = \frac{\sum_i g_{ij} a_{ijk}^n}{\sum_i g_{ij} a_{ijk}^n + \sum_i (2 - g_{ij}) b_{ijk}^n},$$

$$q_{ik}^{n+1} = \frac{1}{2J} \sum_j \left[g_{ij} a_{ijk}^n + (2 - g_{ij}) b_{ijk}^n \right],$$



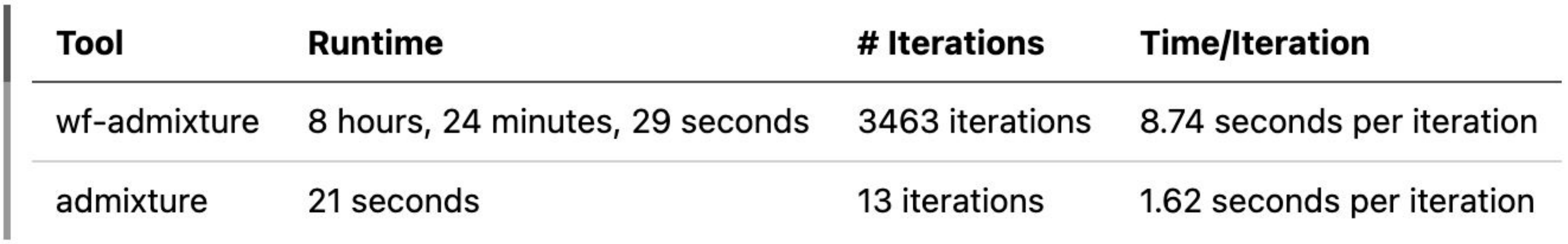
$$a_{ijk}^n = \frac{q_{ik}^n f_{kj}^n}{\sum_m q_{im}^n f_{mj}^n}, \quad b_{ijk}^n = \frac{q_{ik}^n (1 - f_{kj}^n)}{\sum_m q_{im}^n (1 - f_{mj}^n)}.$$

Benchmarking against admixture

I = 462, J = 1000, K = 3. (No multithreading either tool)

Orders of magnitude slower in number of iterations.

Time per iteration comparable.

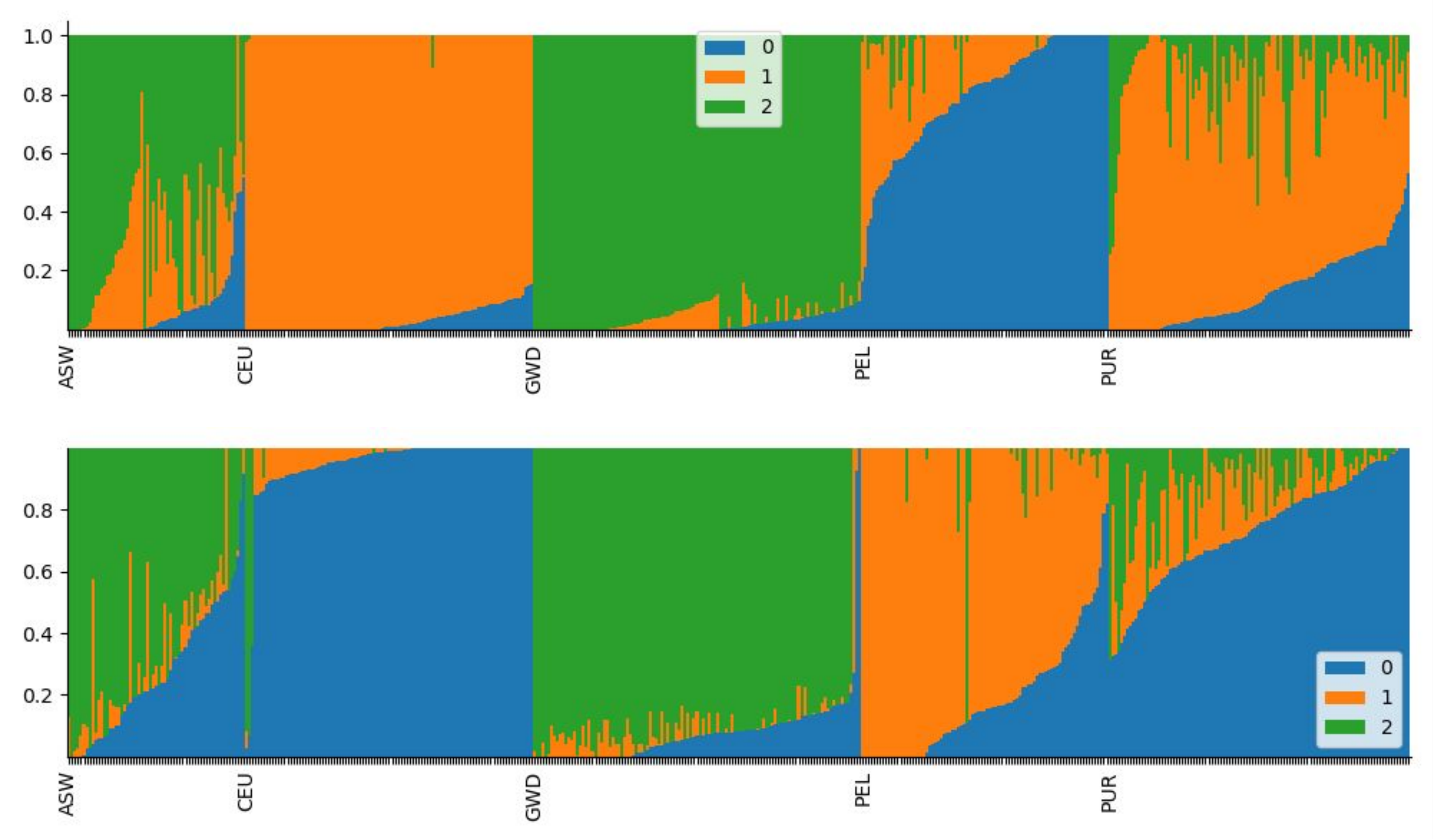


Tool	Runtime	# Iterations	Time/Iteration
wf-admixture	8 hours, 24 minutes, 29 seconds	3463 iterations	8.74 seconds per iteration
admixture	21 seconds	13 iterations	1.62 seconds per iteration

Comparison to admixture on 1000 Genomes

admixture

wf-admixture



Challenges

1. First Time Linear Programmer:
    a. Testing Large Datasets.
    b. Wrapping my head around theory
2. Q Initialization:
    a. uniform
    b. discrete
    c. mixed
3. Unique file formats:
    a. .bed, .fam, and .bim unique formats
    b. Read and write .bed binary file efficiently bitwise

Next Steps

1. Add a block relaxation algorithm
2. Parallelize more steps and more efficiently
    a. No copying data between threads
3. Add acceleration
4. Add confidence levels or some way quantitative way to qualify our results
    a. The math here gets very complex
5. Rewrite in not-python

Sources

https://journals.plos.org/plosgenetics/article%3Fid=10.1371%2Fjournal.pgen.1003925

https://dalexander.github.io/admixture/admixture-manual.pdf

https://genome.cshlp.org/content/19/9/1655

https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-12-246