CM124 CM224: Machine Learning Applications in Genetics **Winter 2020**

Class Project

You can work alone or in group of 2 people.

Grading

- 50 points for any code (that runs) and a report.
- +0 point if accuracy is 50% or less.
- +4 points for each percent gain in accuracy from 50-55%.
- +2 points for each percent gain in accuracy from 55-65%.
- +1 points for each percent gain in accuracy from 65-75%.
- 100 points for accuracy over 75%.
- +5 extra points for ranking 10th to 4th among the groups.
- +10 extra points for ranking in the top 3 groups.

Submit a .zip file, and name it as YourStudentId.YourPartnerId.zip. This file needs to have

- Your code. If you have many files (like header files in C++), put all files into one folder named *code*.
- Your report. This report can be short (1-2 pages) explaining your solution, and can be in docx or pdf format.
- A file named test_data_sol.txt with your predicted haplotypes for the individuals; make sure you output the correct format.
- A readme.txt explaining how to execute your code.

Haplotype phasing from genotypes

The genotype for a person is given as a string containing the values $\{0, 1, 2\}$, indicating the number of copies of the reference allele at each SNP. In this assignment, your task is the following: conditioned on the genotypes in the file test_data_masked.txt, determine the haplotype phase for each person. For a genotype of 0, the phase will be (0, 0) and for a genotype of 2, the phase will be (1, 1), so the only question is how to phase the heterozygous SNPs (e.g. when the genotype is 1). These individuals are from an admixed population, but we do not know the number of ancestral populations or the number of states for each ancestral population. You can use any external data resources to infer these ancestral populations. You are not required to infer these populations, but you may do so if it helps you.

Sequencing technology is imperfect, and will produce missing (or "masked") values for some positions of the genome. These missing values are denoted by the symbol * (as in lectures). In this project, only homozygous genotypes are missing, so * represents either 0 or 2. So, these masked values must either be imputed (e.g. guessed based on surrounding data), or taken into account by the likelihood function of your method (if you use a model with a likelihood function).

You can apply any method to the file test_data_masked.txt to determine the haplotype phase for these individuals. You are allowed to use methods taught in this class and/or your own ideas. Examples include Clark's Algorithm, EM-based methods, HMM-based methods, or others. You can code in Python, R, Java, or C/C++. You are allowed to use any published methods or your own approaches to impute the missing * values. Cite all external resources in your report if you use any. You are **not allowed** to use published methods specifically designed for determining haplotypes from genotypes.

Datasets and submission

You are given the following files:

(a) EXAMPLE GENOTYPES. Two example datasets to build and test your model. Each file contains the genotypes for the individuals, where each row is SNP and each column is an individual.

   i. example_data_1_masked.txt

   ii. example_data_2_masked.txt

(b) EXAMPLE UNMASKED GENOTYPES. These correspond to the masked genotypes, but without the missing data. If you are performing imputation, you can use these to see how well you are doing.

   i. example_data_1.txt

   ii. example_data_2.txt

(c) EXAMPLE HAPLOTYPES. These files contain the true haplotypes for two genotype datasets above. You can consider these files as your ground-truth labels. Each row in the file is a SNP, each pair of consecutive columns in the file is an individual (i.e. for an individual's genotype column i, the corresponding haplotypes are located at column $2i$ and $2i + 1$).

   i. example_data_1_sol.txt

   ii. example_data_2_sol.txt

(d) TEST GENOTYPES. This file contains the genotypes on which we will evaluate your final method. Once you have built your model, run it on this test genotypes and output your estimated haplotypes. Submit these haplotypes for final grading.

   i. test_data_masked.txt

(e) GENOTYPES POSITIONS. These files provide the physical positions of SNPs on the chromosome. You can use this extra information in your model, but you are not required to use it.

   i. example_data_1_geno_positions.txt

   ii. example_data_2_geno_positions.txt

   iii. test_data_geno_positions.txt

Your submission must contain the haplotypes for all the individuals in the test_data_masked.txt. Name your prediction file test_data_sol.txt and use the exact the same format as the example haplotype example_data_1_sol.txt.

The example datasets are provided to you as some samples of possible input and output. These example datasets do not necessarily represent the final test data. You should not think of the example datasets as training datasets; rather, think of these datasets as practice datasets to get a sense of how well the method is doing. Chances are that, if your method performs well on the example datasets, then it will do well on the final test data.

Your method **must** run within 4 hours (on a modern single-core machine) and use a maximum of 16GB of memory. This is to ensure the feasibility of grading the assignments at the end of the quarter. You cannot receive any points other than for the code and report if your code does not fit within these constraints.

Evaluation metric

The metric to measure the accuracy is the *switch accuracy*, which is defined as $\frac{(n-1-sw)}{(n-1)}$, where n denotes the number of heterozygous sites and sw is the number of switches between neighboring heterozygous sites needed in the computer-phased haplotype to recover the original haplotype sequence, see Figure 4 in Lin et al. for a pictorial explanation.

We have a script that you can use to evaluate your model. This same R script will be used to grade your final submission. This script requires the *R* software, and can be run by the command

```
Rscript calculate_switch_accuracy.R predicted_haplotype_file true_haplotype_file
```