6.047/6.878/HSPH IMI.231/HST.507 Fall 2015
Problem Set 4: Alleles and Arrays

Due Thursday, November 12 at 8pm (submit on the course website)

Submit a zip file of a directory named `Lastname_Firstname` containing:

- A PDF file named `Lastname_Firstname.pdf` with your written answers, which should include all plots you are referencing.
- A directory named `code` with all the code you are submitting

In your answers to the questions please refer to the appropriate file name where your code for that problem is located. Unless skeleton code has been provided, feel free to use any programming language you are comfortable with, as long as you structure and comment your code to make it concise and legible.

1 Generalized suffix trees (10pts)

In this problem, we will study some generalizations of suffix trees which allow searching multiple strings and approximate string matching.

(a) Describe a modification to the suffix tree data structure which will allow queries on multiple strings. For example, we may want to search for occurrences of a particular query sequence in multiple reference genomes.

(b) Recall that in the case of a suffix tree on one string, we can construct an equivalent suffix array which will require less space to store. Can your generalized suffix tree be transformed into a suffix array? If so, give an algorithm to do so. Is it possible to directly use a suffix array to solve this problem?

(c) Suppose we are instead interested in allowing only certain mismatches in certain positions (e.g., looking for motif instances). Describe how to build a suffix tree which can handle these queries. Can this tree be transformed into a suffix array?

(d) Suppose we want to search for approximate occurrences of a query string within *Hamming distance* k (number of mismatches at most k). Describe an algorithm to perform this query on a suffix tree.

Extra credit: Describe an algorithm to perform this query on a suffix array.

2 Finding eQTLs (20pts)

In this problem, we will examine the sources of variation in gene expression that partition a population into sub-populations. You will find the datasets used in this question in the `eQTLs` folder available though the problem set folder on the course website.

(a) In the file `ExpData.txt`, you will find log-normalized RNA-seq expression data from our population of 1000 samples, with 5000 genes profiled for each sample. Do a principal components analysis on this dataset to find the clusters of samples that have similar patters of gene expression. Plot the output of your analysis, and describe the patterns that you observe. What is the structure inherent in this population?

For PCA, we recommend you use the `princomp` function in the `stats` package available by default in `R`. However, many other languages such as MATLAB and python have analogous functions; you should use whatever you are most comfortable with. In your plots, be sure the axes are labeled with the components you are displaying in each plot. Also make sure that at least one of your plots colours the points corresponding

to the samples with the sub-population that you think they should belong to. (Hint: You can re-use your k-means code from Pset 3 to find these sub-populations!).

Hand in your write-up and the code you used for plotting and assigning samples to sub-populations.

(b) In the file `SnpData.txt`, you will find genotyping data for the same 1000 samples across 500 SNPs. Each SNP's genotype has been called with reference to the same reference genotype; "0" thus represents the reference allele, "2" represents the non-reference allele, and "1" represents a different allele on each strand.

You will find that some of the SNPs (more than 5, less than 100) are eQTLs, that is, they have an effect on the expression of one or more of the genes we collected expression data for. Using whatever model you see fit, search for these eQTLs using the genotyping data and the expression data. You may not have the computational resources to test all combinations of SNPs and genes, so you should think about smart ways to choose subsets of each to find some eQTLs - you don't have to find all of them!

For three of the eQTLs you found, present the evidence you have for why you think it is an eQTL, and not just associated with the expression of a gene by chance alone. Be sure to include plots in your analysis to support your hypothesis, and to thoroughly explain the method you used to find eQTLs. You can assume that the association between genotype and expression is linear for eQTLs. Don't forget that you should be correcting for the fact that you are performing multiple significance tests.

Hand in your write-up as well as the code you used to look for eQTLs in the two datasets provided.

(c) In the above analysis, we were forced to consider all pairs of SNPs and genes to identify eQTLs. What sources of data that have not been provided as part of this problem would have been useful in constraining the amount of such pairs you had to test? For at least two sources, give a description of what the dataset would look like (what are the rows and columns of the data matrix? what kinds of values are stored in the matrix?) and explain how you would use it to filter out pairs of SNPs and genes that are unlikely to be associated with one another.

3 Coalescent simulation (6.878 only, 10pts)

In this problem, we will simulate the coalescent process. Recall this is the time-reverse of the Wright–Fisher process.

(a) Write a program to simulate the coalescent process on a population of N alleles. Track the times of coalescent events starting from the initial generation until all alleles coalesce to a single ancestor. If we are tracking k lineages, you should report $k - 1$ coalescent events.

Recall that the Wright–Fisher process assumes each allele in the next generation is sampled independently from all alleles in the current generation. We are now interested in the reverse, so we instead need to sample parents in the previous generation uniformly at random with replacement. Note we are interested in the identities of the parents and not their ancestral alleles.

Run 1,000 trials with a population size of $N = 500$. Report the mean and standard deviation of the number of generations between coalescent events of $k = 2$, 3, and 4 lineages.

(b) Recall the waiting time between coalescent events is approximately exponentially distributed with parameter λ. For each value of k, what is the value of λ given $N = 500$?

Given this distribution, the mean waiting time and its standard deviation are both $1/\lambda$. How do these expected values compare to your observed values? If your observed values are different, give an explanation of what could have caused the differences.

(c) Extend your simulator to model sexual reproduction.

Assume a fixed number of females F (and therefore $M = N - F$ males) in each generation and that each chromosome in the next generation is selected in the following way: sample a male and female to mate uniformly at random, then sample one of the two alleles uniformly at random. Your simulation should do the reverse: sample a father and mother and then pick one at random as the ancestor for each allele.

Run 1,000 trials with $F = 100$ and $M = 400$. Do your results agree with the coalescent approximation? Justify your answer as in (b).

Can you extend the coalescent approximation to more accurately reflect this model of sexual reproduction? Do your results agree with this new approximation?

MIT OpenCourseWare
http://ocw.mit.edu

6.047 / 6.878 / HST.507 Computational Biology
Fall 2015

For information about citing these materials or our Terms of Use, visit: http://ocw.mit.edu/terms.