# PSET 4 - 6.047/6.878/HST.507 - Fall 2018
# Alleles and Arrays

Due Monday, November 5 at 11:59pm (submit via Stellar)

Please submit a zip file of your solution file via Stellar. Make sure you review the Stellar announcement regarding the submission format.

## 1 Simulated Genome-wide Association Studies

In this problem, we will use a very simple model to simulate genome-wide association studies. In general, we often think of binary phenotypes like the presence of a disease as a cutoff applied to an underlying continuous phenotype. We're going to represent a person's genotype as a list of $m$ snps, $k$ of which relate to a particular disease. The underlying continuous phenotype will be modeled as

$$y = \sum_i \beta_i * x_i + \epsilon$$

$\beta_i$ represents the effect size of snp $x_i$. For the sake of simplicity, we will assume that each snp **can only take the values 0 or 1**. $\epsilon$ is an error term, which we will model as a normal distribution with mean 0 and standard deviation 1. We will say that a person has the disease if the y-value is greater than 2.

(a) What is the relationship between $\beta_i$ and a snp's odds ratio? Express your answer in terms of $\Phi(x)$ the cumulative distribution function of a normal distribution.

(b) If we assume that each of the $k$ disease-related snps have beta values distributed according to a normal distribution with mean $0$ and standard deviation $s$, we can generate a beta value for each of these $k$ snps. Now, assuming that each of the $m$ snps occurs in the population with a probability of $0.05$, we can generate $n$ people's genotypes, again remembering that each of the $m$ snps can only take the values 0 or 1. For each person, we can then randomly generate an $\epsilon$ value, and compute $y$. Finally, we can then use our cutoff of 2 to determine if the person has the disease. Write a program which can do generate the genotypes and $y$ values for a population for arbitrary $k$, $n$, $m$, and $s$.

(c) For each snp, we can calculate a chi-squared statistic and p-value for it based on our simulated population. In this particular case, we compare it to a chi-squared value with 1 degree of freedom. For reference: https://www.youtube.com/watch?v=hpWdDmgsIRE

(d) In terms of m, what statistical p value is needed to achieve significance up to a Bonferonni correction? Why is a Bonferonni correction needed with GWAS testing?

(e) Run your program with the following sets of hyperparameters, and report the accuracy and precision:

- n = 10000, k = 100, m = 1000, s = 0.25
- n = 1000, k = 100, m = 1000, s = 0.25
- n = 100000, k = 100, m = 1000, s = 0.25
- n = 10000, k = 100, m = 10000, s = 0.25
- n = 10000, k = 100, m = 1000, s = 0.5
- n = 10000, k = 100, m = 1000, s = 0.1

What patterns do you notice in these results? Describe the effects of changing each of the three variables.

(f) (6.878 only) Now we make another simplifying assumption. There are 1000 snps, of which only one snp is associated with the disease. Create a simulation which empirically estimates the probability of this snp having a chi-squared score which is statistically significant. Run this simulation for a number of different betas, and convert each of these betas to an odds ratio. Now, make a plot of the probability versus odds ratio for various odds ratios across a number of different population sizes.

## 2 Finding eQTLs

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 Stellar 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 [1] on this dataset to find the clusters of samples that have similar patters of gene expression. Plot the output of your analysis. 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!)

Describe the patterns that you observe. What is the structure inherent in this population?

Be sure to include 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:

i. Give a description of what the dataset would look like (i.e. what are the rows and columns of the data matrix? what kinds of values are stored in the matrix?).

ii. Explain how you would use it to filter out pairs of SNPs and genes that are unlikely to be associated with one another.

[1] 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.

## 3 Convolutional Neural Networks

Convolutional Neural Networks are a very popular way of classifying images, achieving state of the art performances on a number of different tasks. In this problem we will use them to classify synthetic sequence data. To do this, we will use the machine learning package Keras.

To install, use the command:

$$pip\ install\ keras$$

In the problem set folder are two different datasets, one entitled positivedata.txt and one entitled negativedata.txt. The positive model was generated from a set of motifs, while the negative model has randomly generated sequences.

(a) We've provided skeleton code to load and format the data into a one-hot encoding. Now, build a convolutional neural network with the following layers:

- convolutional layer (pooling size 4,6)
- max pooling layer
- flatten layer
- dense layer with relu activation
- dense layer with softmax activation

When you finally fit your model leave 10 percent of the data as test data. In the writeup, paste your code and report the accuracy on training and test data. (A clarification, training data is used to train the weights of the network, test data is used to determine the accuracy of your model.)

(b) What are the dimensions of the data at each of these different layers? Briefly explain what each of the layers do and how they collectively help classify the sequences.

(c) Experiment with making at least 3 huge changes to some of the hyperparameters in your architecture. Describe if you notice any changes to the convergence rate and test/training accuracies. Why do you think the changes you made to the hyperparameters brought about the observed effects?

(d) Now experiment with completely different architectures than the one in part a. Report on what different architectures you tried, as well as what the training and test accuracies were for these architectures. Again, try to explain why the new architecture performed the way it did. Consider using dropout regularization or adding more dense layers.