MSCBIO 2070/02-710: Computational Genomics, Spring 2017

HW2: Gene expression analysis

Due: 23:59 EST, Mar 21, 201 by autolab

Your goals in this assignment are to

1. Understand the basics of multiple hypothesis testing
2. Explore properties of linear models
3. Investigate moderated T statistics and compute permutation FDR
4. Build HMMs for lncRNA detection

What to hand in. We will assume that programming exercises are implemented in Python, however this assignment can be easily completed in `R` or `MATLAB`. Feel free to use those instead but still include the code. We will not be autograding code this time but it should be included as text in your report.

- One report (in pdf format) addressing each of following questions including the figures generated when appropriate.
- All source code for the programming exercises included as text in the report. We should be able to run the source code and produce the result and figures requested.

1. (a) **[10 points] Basic multiple hypothesis testing**

- Suppose we are performing a gene expression experiment looking at the effect of a drug and we have found an interesting gene, geneX, that has a *uncorrected* T-test p-value of 10^{-5}. What is the corresponding Bonferroni adjusted p-value if we started with 20,000 genes? Suppose I have a hypothesis that the drug affects genes in the glycolysis pathway (which geneX is a member of) and I will only consider those genes for hypothesis testing. What is the Bonferroni adjusted p-value if there are 500 genes in the glycolysis pathway?
- I have also computed the Benjamini-Hochberg (BH) FDR correction for this experiment and it is 0.01 at the p-value $\leq 10^{-5}$ threshold, when considering all 20,000 genes. Recall that to compute the BH correction we can compute

  $$p_k^* = \frac{mp_k}{k}$$

  Where m is the number of hypothesis and k is the rank of each p-value when sorted in increasing order. We also require that p^* be monotone non-decreasing in k so the final p_k^* is set to $min_{i \geq k} p_i^*$. Suppose that my hypothesis is correct and among the glycolysis pathway genes p-values $\leq 10^{-5}$ are 10 times more frequent the the genome-wide rate for all 20,000 genes. What can you say about the BH FDR for the same threshold computed on the glycolysis restricted geneset. We will assume that all p-values are unique and that $p_k^* = \frac{mp_k}{k}$ is already non-decreasing. Hint: consider the relative rank k/m for a gene with p-value 10^{-5} and a genome-wide BH FDR correction 0.01.
- Unfortunately, it turns out that I am wrong and the mechanism of action is not related to the glycolysis pathway. What can you say about the FDR on the glycolysis restricted gene set in this case. We are now assuming that the distribution of p-values is identical for glycolysis pathway and non glycolysis pathway genes.

(b) **[10 points] Multiple hypothesis testing with dependent tests**

Recall that if R is the number of rejected hypothesis and V is the number of falsely rejected hypothesis we can define $Q = V/R$ for $R > 0$ and $Q = 0$ otherwise. The FDR then is defined as $E(Q)$ where the expectation is taken over hypothetical repeated experiments where the data is drawn from the same multivariate distribution, representing some mixture of null and non-null hypothesis. (Of course, in order to calculate this exactly we need to agree on a data generating model.) Ideally we want to estimate Q for our specific experiment but we can only get a bound on $E(Q)$ and the expected value and the "typical" case can indeed be very different when random variables are not independent. We will now explore a toy example of how FDR behaves under dependence.

Consider the following gene expression experiment. For simplicity the data is reported in terms of sample-wise gene ranks. In this toy system genes 1,2,3 and genes 4,5,6 represent co-regulated clusters and one of these two clusters always occupies the top 3 ranks of each sample (columns of the table).

	Group1			Group2			Test-statistic	Permutation p-value
gene1	1	2	3	4	5	6	?	?
gene2	2	3	1	5	6	4	?	?
gene3	3	1	2	6	4	5	?	?
gene4	4	5	6	1	2	3	?	?
gene5	5	6	4	2	3	1	?	?
gene6	6	4	5	3	1	2	?	?

i. We are interested in contrasting Group1 and Group2. The data are not normally distributed so our test statistic will be the absolute value of the mean difference between Group1 and Group2. If x_i is the value of a gene in sample i the test statistic is simply

$$\frac{|\sum_{i\in\{1,2,3\}} x_i - \sum_{i\in\{4,5,6\}} x_i|}{3}$$

What is the test statistic for each gene?

ii. We will now compute a permutation based p-value. How many ways are there to split 6 samples into two equal groups? Please give the value up to Group1-Group2 symmetry so that a ((1,2,3),(4,5,6)) split is the same as ((4,5,6), (1,2,3)). What fraction of those permutations would give you a test statistic equal to or greater than what you got in i. above–this is the permutation p-value. Note, the p-value cannot be 0 so we will consider sample grouping given in the table as one of the permutation possibilities. What is the p-value for each gene?

iii. Perform the Benjamini-Hochberg FDR adjustment on these p-values. Applying the definition here as a little tricky since all the p-values are the same however it still works out. We can rank the p-values in arbitrary order and apply the procedure recalling that the adjusted p-value must be monotone non-decreasing. Since the p-values are all equal and there is a single possible rejection threshold the BH FDR will also be equal for all genes.

iv. Let's now assume that there are in fact no differences between the two groups and the 6 gene-level hypothesis are all null. Given this information we can calculate $E(Q)$ directly where the expectation will be taken over all the possible splits into two equal groups (as in ii. above). We consider a hypothesis rejected if the test statistic (absolute value of the difference in means) is greater to or equal to that in i. above. While calculating $E(Q)$ in a general setting is difficult in this case Q can only take on 2 values $\{0, 1\}$ so we just need to figure out the probability of each. What is $E(Q)$?
Consider an alternative situation where gene1 and gene4 are not null. What is $E(Q)$ now? (Hint: Q can still only take on 2 values). How do these explicit calculations compare with the BH correction?

v. Extrapolate your findings to a realistic differential expression scenario. Suppose I performed a differential expression analysis comparing blood from a sample of people that sit on the left side of the classroom to a sample of people that sit on the right side of the classroom and found 200 out of 20,000 genes to be differentially expressed at a BH FDR of 0.1. I conclude that there are real molecular differences associated with classroom side. Are you convinced? What additional information about the experiment, the data, and/or the analysis would be helpful for judging the validity of this conclusion (several possible answers here)?

2. **Linear models**

- **[10 points] Model matrix equivalence in linear regression** Recall from lecture that given

  a gene expression experiment with two groups of two samples each, we can specify a linear model for the expression of a single gene as,

$$y = X\beta + \epsilon$$

$$y = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} + \epsilon$$

Using the formula for the solution to least squares regression show that if we rewrite this as

$$y = X'\beta' + \epsilon'$$

$$y = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \beta'_1 \\ \beta'_2 \end{pmatrix} + \epsilon'$$

we have that $\epsilon = \epsilon'$. Start by writing X' as XA. What is A? What is the relationship between β and β'? Formulate and prove a general statement/theorem about the equivalence of different model specifications? In the case the two models are X and X' with $X' = XA$. The statement should be in the form: given a model X and a matrix A (with some conditions) the following relationships hold for ϵ, β and ϵ', β' for the corresponding least squares fits ...

- **[20 points] Deriving Ridge Regression**

  Given a gene expression experiment, a common task is to infer the transcriptional regulatory network. That is, we want to learn a directed acyclic graph in which each transcript is connected to its regulators by an edge. An interesting method for this is neighborhood regression - for each gene, we fit a linear model which predicts the gene expression value from all other expression values. Highly predictive genes are then selected to be the parents of the gene in the learned graph. Here, we will investigate a few forms of this linear model.

  Recall from lecture that we can specify a linear model by

  $$Y_i = X_i\beta + \epsilon$$

  where X is a matrix of size $n \times p$, β is a vector of length p and ϵ is a noise term. In class, we assumed that $n \gg p$, but with gene expression experiments, we often have thousands of expression values with only a few hundred samples. In this setting, the least squares estimator is likely to not have a unique solution. To fix this problem, we can add a penalty term which encourages small effect sizes. With this penalty, we would like to solve the following problem:

  $$\hat{\beta}_{RR} = min_\beta \sum_i (y_i - X_i\beta)^2 + \lambda \sum_j \beta_j^2$$

  where λ is a fixed positive number. This problem is known as Ridge Regression.

  Similarly to the formula for least squares ($\hat{\beta}_{LS} = (X^TX)^{-1}X^TY$), derive the closed form estimator for Ridge Regression. Like regular least squares linear regression, this is a quadratic function of β and the minumum is found by setting the derivitive to 0. Hint: $\frac{d}{dX}(X^T) = \frac{d}{dX}(X)^T$.

  Fill out the stub code in regression.py (if using python) to compute Ridge Regression and Least Squares estimates for the provided gene expression assays from The Cancer Genome Atlas. The data is located in the file regression_data.tsv, which forms a matrix of size 99x501. The first 500 features form the X matrix, and the final value gives the Y vector. Set the hyperparameter λ to 1. If using Python, these details are handled for you in the regression.py script.

  Qualitatively, what is the difference between the Ridge Regression and Least Squares estimates? Is one more sparse than the other?

3. **Moderated T statistic [25 points total]**

Here we will write some custom code to calculate moderated T statistics. Begin by writing a simple function to calculate a T statistic using equal variance assumption. Your function should take 2 inputs: a vector of data and a vector specifying the group membership numerically as $1, 2$.

The formula for an equal-variance T statistic is

$$t = \frac{\bar{X}_1 - \bar{X}_2}{s_{X_1X_2} \cdot \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

where

$$s_{X_1X_2} = \sqrt{\frac{(n_1 - 1)s_{X_1}^2 + (n_2 - 1)s_{X_2}^2}{n_1 + n_2 - 2}}$$

Here s_{x_1} and s_{x_2} refer to the standard deviations calculated within the two groups respectively and n_1 and n_2 is the number of samples. We have provided some simulated data in `simData.tsv`. The file contains a "gene expression matrix" with 5000 genes and 40 samples. The sample groups that we wish to contast are the first and second 20 columns. The data was simulated as follows for each gene the variance is drawn for a scaled inverse χ^2 distribution with degrees of freedom = 3 and scaling factor = 5 as in

$$s_i^2 \sim \frac{5}{\chi_3^2}$$

The values are drawn from a normal distribution as follows

	samples $i = 1 \ldots 20$	samples $i = 21 \ldots 40$
genes $i = 1 \ldots 500$	$\mathcal{N}(1, s_i)$	$\mathcal{N}(0, s_i)$
genes $i = 501 \ldots 5000$	$\mathcal{N}(0, s_i)$	$\mathcal{N}(0, s_i)$

So the first 500 genes are up-regulated in the first 20 samples. Check that your T-statistic code works by verifying that for the first gene it comes out to about 2.927.

Let your function take an additional parameter s_0 to be added to the denominator of the T statistic equation

$$t = \frac{\bar{X}_1 - \bar{X}_2}{s_0 + s_{X_1X_2} \cdot \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

a value of 0 should leave the result unchanged. This is the "fudge factor" in SAM analysis.

Complete your T statistic function by allowing `x` to be a matrix and computing a single T statistic per row.

Simulated Data [10 points]

Given that we know that the first 500 genes are simulated to have different means we can test the performance of various statistics at distinguishing these genes from the rest. Write a function that computes the Area Under Receiver Operating Characteristic (ROC) Curve (AUC). You function should take 2 inputs: the statistics for each gene and the labels (whether or not the gene is differentially expressed).

Since in a real application we do not known if genes of interest are up- or down- regulated we will use **the absolute value of the T statistic above** as our readout. Feel free to use any libraries here though it may just be easier and faster to do this from scratch. If we defined t_i^r to be the rank of the i'th T statistic (in increasing order of absolute value) than the AUC is.

$$\frac{\sum_{i \in P} t_i^r}{n_p n_n}$$

It is the sum of the ranks of the positive set P (here the first 500 genes) divided by the product of the number of positive and negative genes. You should be getting values very close to 1.

- Now we will see if the moderated T statistic gives better performance. Compute T statistics using at least 20 equally spaced s_0 values in the range $[0, 1]$ and plot the AUC for the results relative to the value of s_0. Which value achieved the best performance?

Real Data [15 points]

Now we will apply this to real data. A gene expression dataset of two subtypes of leukemia is included in the file `cancerData.tsv` and the group identity is specified in `simDataGrp.tsv`. Note that the groups are not of equal size.

For this datasets we do not know which genes are differentially expressed therefore we will use the permutation based emprical FDR strategy to optimize $s0$

Write a function that generates the distribution of the moderated T statistic under group permutation. As input it should take the gene expression data, the group identity, and the number of permutations to perform as in `permuteTstat(data, grp, nperm)`. It is important to remember that the groups are permuted once per permutation instance and all the genes are tested against the same group permutation. The output of this function should be a vector of T statistics that has length (number of genes) × (number of permutations).

Write a function that takes in the real T statistics and the permutated T statistics and calculates the FDR for each value of the real T statistic. Again we will compare the absolute values here. The function should return a vector of false discovery rates at each value of the real T statistic.

Since the true differential expression status of the genes is unknown we will use the number of genes with empirical FDR of <0.1 as a performance metric. Using this metric make a plot of moderated T statistic performance for different values of s_0 (20 equally spaced s_0 values in the range $[0, 0.3]$). The moderated T statistic will have a different distribution so it is necessary to rerun the permutation analysis every time. Perform 20 permutations for each value of s_0. In order to facilitate comparisons across different values of s_0 keep the permutations identical. This is accomplished by setting the random seed for each set of permutations as in `np.random.seed(123)` in Python or `set.seed(123)` in `R`. When the permutations are identical across s_0 values the resulting plot of FDR against s_0 will look reasonably smooth with a clear peak even for 20 permutations. Be aware that running the permutations will still take a some time.

- **[15 points]** Plot the performance against the s_0 value and find the optimal constant for this dataset.

4. **[10 points] lncRNA detection** Recall the lncRNAs often have poor base-level conservation across genomes and often cannot be found using BLASTn however in many cases they can be reliably detected using HMMs. Show an HMM structure for detecting lncRNAs with the following sequence properties. We consider two different lncRNAs which will be recognized by two different HMMs. Only show hidden nodes but specify if they output single or multiple characters. Show start and end states. Specify parameters only if explicitly asked.

   - lncRNA is determined by a series of semi-regularly spaced AAGGC repeats, like the *Megamind* gene from lecture notes. The inter-repeat length has a mean of 5 and a variance of 10. In order to get the correct distribution for the inter-repeat length use the parameterization of the Negative Binomial distribution and refer to slides on duration modeling. Give only those transition probabilities that are uniquely specified. The region recognized should start with a repeat.
   - lncRNA is determined by a series of tandem repeats of variable length with variable linker length such as the *Xist* gene. The repeat sequences are CCAC, TTGGG, and TAGTAG. No need to specify parameters. The region recognized should start with the first repeat.