HW1: Sequencing, Statistical Concepts, and Population Genetics

Due: Friday September 24th, 2021 at 5 pm.

Submission Guidelines

Please read the instructions carefully! Name all files using the following convention:

- lastName_firstName_shortDescriptionOfFile.extension.

You must submit **one** pdf file and **one** file contating your code. If using R this can either mean including:

- one pdf file and one .R file **or**
- one pdf file and one .Rmd (R markdown) file.

1 Sequencing and Assembly

1. What is shotgun sequencing and how has this technique significantly contributed to the acquisition of the human genome?
2. What are the relative advantages and disadvantages of Sanger sequencing vs. next generation (Illumina) sequencing?
3. What is paired-end sequencing? Give a cartoon example (i.e., a sample set of reads and/or a piece of a genome) where paired-end sequencing leads to improved assemblies.

2 Computing q-values from p-values

Here we're going to analyze a file that contains a table with two columns:

- `pval`: p-values from simulated data (t-test on counts from Poisson model with random λ values; details aren't important).
- `null`: has a value of 1 if the p-value is for a condition-negative case (where the null hypothesis holds) or 0 otherwise (the alternative hypothesis holds).

Note the second column has information that would be unknown in practice, but we'll use it in this thought experiment to understand how q-value estimation works. Write an R script that does the following:

1. Load the file **sim-pvals.txt**. Hint: use the R function `read.table`, though other functions can be used too.
2. What is π_0 (the proportion of condition-negatives or null hypotheses) in this data?
3. Plot the p-value distribution of the null and alternative p-values separately. Hint: use the R function `hist`. Why do these distributions have the observed shapes?
4. Now plot the combined p-value distribution. Please use the `freq=FALSE` option if you're using `hist`, so the y-axis shows the density rather than the frequency of each bin. Draw a horizontal line with a height of π_0 on this density. What does π_0 correspond to in this figure?
5. Based on the last figure, how do you suggest estimating π_0 from the combined p-value distribution when the true condition of each case is unknown? No equations are necessary, a one-sentence summary of most important part of the strategy suffices.
6. Calculate q-values manually:
   - For each p-value p, start by using this formula that estimates the FDR:
   $$q'(p) = \frac{pm\pi_0}{\#\{p' \leq p\}},$$
   where m is the number of p-values, π_0 is the true value of the parameter you calculated earlier from the null column, and $\#\{p' \leq p\}$ is the number of p-values in the whole list that are smaller or equal to the given p.
   - The next step is to ensure monotonicity. Typically the FDR increases as the p-value increase. When the FDR estimate actually get smaller for a larger p-value, you should accept the lowest FDR you can that includes your given p-value: $q(p) = \min_{p' \geq p} q'(p')$
   - Lastly, cap q-value estimates to 1 as needed.
7. Explain why $q'(p)$ above estimates the FDR. In particular, explain what $\#\{p' \leq p\}$ is, what $pm\pi_0$ estimates, and why their ratio estimates the FDR.
8. Plot q-values against p-values and describe the relationship. What is the maximum q-value and why?
9. Compute q-values with the R `qvalue` package.
   (a) **If you are programming in R**, install the R `qvalue` package. To do this you need to use BiocManager from bioconductor by running the following in the R console:

   ```
   if (!requireNamespace("BiocManager", quietly = TRUE))
       install.packages("BiocManager")

   BiocManager::install("qvalue")
   ```

   If you run into any problems this might mean you a version of R < 3.5 (double check by typing version in the R console), then type the following in the R console:

```
source("https://bioconductor.org/biocLite.R")
biocLite("qvalue")
```

To compute q-values with the package, use the function `qvalue` of that same package.

(b) **If you are programming in another language**, you can use the `qvalue` web server instead (http://qvalue.princeton.edu/). Upload the `sim-pvals.txt` file and change the "Separator" option to "Tab." Output > Summary will show the results and estimates of the run, and if you scroll to the bottom, the "Download Output" button will give you a data file you can import for plotting.

Use the original p-values and nothing else as input, with default options. Report the π_0 estimate of the package.

10. Show a plot that compares our manual q-value estimates with the q-value estimates of the package, and the $y = x$ line for comparison. Do they differ? Why?

3 Precision-recall Analysis

For this problem, you will need to examine a ranked list of genes and determine whether or not the ranking is biologically relevant. Specifically, you will ask whether genes, known to be associated with a particular biological function, preferentially occur at the top of your experimental list or, alternatively, are randomly scattered throughout the list.

Imagine that you have performed an experiment in which you have treated yeast *Saccharomyces cerevisiae* with a chemical compound. To determine which yeast genes are likely responsible for protecting the cell against chemical stress, you have measured the expression of all genes before and after the treatment. After performing proper corrections and normalizations, you came up with a list of genes that are differentially expressed upon treatment and ranked the genes based on the extent and confidence of their differential expression.

3.1 Data import and cross-checks

To work on this problem you will need to download the following datasets and import them into your working environment. The R function `read.table` might be useful here, although definitely not the only way to import your data.

1. The list of genes that are differentially expressed in your experiment: **differentially_expressed_orfs.txt**
2. The list of genes known to be involved in a specific biological process: **positive_orfs.txt**

To verify that your data import was successful, perform the following cross-checks:

1. Is your list of differentially expressed genes unique? If not, which genes are repeated? How many times are they repeated? Remove the repeated ORFs, if any, keeping only their first occurrence. Hint: the R functions `length`, `unique`, and `table` might be useful.
2. How long is the list of known genes? How many of them appear in your differentially expressed list?

3.2 Precision-recall analysis

Assume that the order of genes in the **differentially_expressed_orfs.txt** file reflects their experimental ranking, which is based on the extent and confidence of their differential expression. At each position in this ranked list, calculate the precision and the recall of the data relative to the list of known genes. Plot the corresponding precision-recall plot. Assume that every gene in the genome was tested for differential expression, even those that do not appear in the list of differentially-expressed genes.

1. What is the precision of these data at 10% recall?
2. Do you think it is reasonable to conclude that your experiment is recapitulating well previously known biology? Motivate your answer.

4 Population Genetics

The files **pop1.txt** and **pop2.txt** contain genetic data from a 10 kb region sequenced in two populations. Each biallelic segregating site (S) contains two alleles, labeled 0 or 1. For each population, genetic data from 100 chromosomes was obtained, where each line represents genetic data from a different chromosome (another way to describe the data is that each line represents a different haplotype; alleles at loci that occur on the same chromosome). For example, "0100" means there are four segregating sites in this region and this particular chromosome has allele 0 at the first segregating site, allele 1 at the second segregating site, allele 0 at the third segregating site, and allele 0 at the fourth segregating site.

1. Write code to determine the number of segregating sites, S, in populations 1 and 2. Hint: the R function `readLines` can help read in the data files.

2. For each population, calculate a per nucleotide estimate of θ_W and π (i.e., divide θ_W and π by the number of bps sequenced in the region).

3. Calculate the statistic $D = \pi - \theta_W$ for each population.

4. Use your estimates of θ_W to estimate N_e in each population. Note that for this problem, you can assume that the mutation rate μ is equal to 1×10^{-8} per site per generation.

5. Based on the data in 1-4, do you think N_e is the same or different between population's 1 and 2? What aspects of the data are most informative in arriving at your answer?