Computational Biology Final Project

October 29, 2020

Write a report analyzing the RNA-seq expression patterns in the provided data set—focusing on your particular assigned gene set—as described in the next couple of pages. Your assignment submission will consist of 2 or 3 distinct files:

- The report itself, which may be in either PDF or HTML format (exported from word processor or generated from a .Rmd script or a .tex file) or a Jupyter notebook.
  - If your report is in Jupyter notebook format: make sure the notebook has been saved with all required graphics generated and included in the notebook.
- If the report is not a Jupyter notebook, then you must also upload a code file in either plain R (.R or .r file) or R markdown (.Rmd or .rmd file).
  - if your report is in Jupyter notebook format, just upload the same .ipynb file for both the report and the script files.
- A tab-delimited text file (.tsv) containing the ids, names, and descriptions of the genes in your assigned gene set generated as specified in item **4** below.

In interpreting the RNA-seq data set, you should note the following descriptions of the different sample genotypes:

Col wild-type *Arabidopsis thaliana*.

14B a mutant lacking two genes for a plant specific translation initiation factor, eIFiso4G1 (AT5g57870) and eIFiso4G2 (At2g24050).

4G a mutant lacking the one gene for eIF4G (AT3g60240), a translation factor that is more similar to those of other eukaryotes.

Also note that the samples were collected from two different time points/light conditions: "End day" (light) and "Ex dark" (dark).

In this project, we are primarily interested in understanding how the differences in gene expression between the light and dark conditions vary by genotype. To this end, we will use `DESeq` to look for genes with significant *interaction* terms between genotype and timepoint. The model used by `DESeq` treats such interactions in essentially the same manner as a two-way ANOVA model, such as those you would have dealt with in biostatistics, deals with interaction terms. As a quick review of this idea, you might read through:

`https://statisticsbyjim.com/regression/interaction-effects/`

*Assigment continues on **next page**...*

You can get started by downloading the files:

- rna_counts.tsv.gz
- rna_sample_annotation.tsv
- arabidopsis_thaliana_gene_names.tsv.gz
- gene_sets.tsv.gz
- bio321g_rnaseq_utils.R

Your script file should include the line `source("bio321g_rnaseq_utils.R")`, which will use the provided R script file to load in the provided data files along with a couple other objects and a function which you will need in your analyses. (Take a look at this file so you know what R objects it creates for you to use.)

Items to include in your code and report

1. Construct a `DESeqDataSet` object using `DESeqDataSetFromMatrix` setting the `design` argument to the function `DESeqDataSetFromMatrix` as follows:

   - `design = ~ time + genotype + time:genotype`

   Then use the `DESeq` function on the output from `DESeqDataSetFromMatrix` together with the additional arguments:

   - `test = "LRT"`
   - `reduced = ~ time + genotype`

   in order to perform a hypothesis test specifically for the significance of the interaction term `time:genotype` in the model. Finally, use the `results` function from the `DESeq2` package to extract a table of results for each gene from the `DESeqDataSet` object.

   - For how many genes from the full data set would we conclude that there was evidence of a significant time:genotype interaction term if we wanted to keep the false discovery rate (FDR) ≤ 0.10 (or 10%)?
   - What would we be the expected number of false positive discoveries out of the list of genes with evidence of significant interaction terms determined this way?

2. Extract the normalized counts from the `DESeqDataSet` object you created above and define `lgNorm` to be the result of first adding 1 to the normalized counts and then $\log_2$ transforming them. In your report, state that data was normalized with `DESeq` and log-transformed with an offset of 1.

3. Generate (and include in your report) a principal components analysis (PCA) plot (using `ggplot2`) for the data in `lgNorm`. Indicate in your report what samples are most separated from each other along the PC1 direction in the plot and, similarly, what samples are most separated from each other along the PC2 direction in the plot.

   - Color the points in the plot according to which `group` they are assigned according to the file `rna_sample_annotation.tsv`. Add `scale_color_manual(values=groupColors)` to your plot object to use the colors encoded in the `groupColors` loaded in when you `source("bio321g_rnaseq_utils.R")`.

*Items continue on **next page**...*

4. Use R to extract a vector containing the ids of the genes associated with your assigned gene set from the provided file `gene_sets.tsv`.

   Make a `data.frame` with three columns: (1) gene id, (2) gene name/symbol, and (3) gene description; the table should have one row for each gene in your assigned gene set. Save this `data.frame` to a tab-delimited text file to include with your assignment. Indicate that this file has been provided and state what name you gave the file in your report.

5. Make a new `data.frame` by filtering `lgNorm` to retain only those rows of that `data.frame` corresponding to genes in your assigned gene set. Assign this new `data.frame` to the variable `lgGo`. (Only need to address this item in code file.)

6. Generate and include a principal components analysis (PCA) plot for the data in `lgGo`. For the report: in your own words, compare the positions of the samples from the various groups to the positions of the same samples in the full-gene-set PCA plot.

   - Color the points in the plot according to which `group` they are assigned according to the file `rna_sample_annotation.tsv`. Add `scale_color_manual(values=groupColors)` to your plot object to use the colors encoded in the `groupColors` loaded in when you `source("bio321g_rnaseq_utils.R")`.

7. For the gene set represented in `lgGo`, use `pheatmap` to generate and include a clustered heatmap of the differences in each gene's (log-transformed normalized) expression level from it's (the gene's, that is) mean (save these differences in expression levels from their gene-wise means to a new `data.frame` named `heatData`). Limit the dynamic range exhibited in the heatmap by setting any expression level difference greater than 2 $\log_2$ units greater than the mean to +2 and any expression level difference less than 2 $\log_2$ units less than the mean to -2.

   - Supply the object `heatPalette` created when you ran `source("bio321g_rnaseq_utils.R")` as the second argument to `pheatmap` in order to reset the color palette used in the heatmap.
   - Also include the named argument: `labels_row=geneNamesAndDescriptions[rownames(heatData), "symbol"]` in your `pheatmap` call to use the gene names/symbols as the names of the rows of the heatmap instead of the gene ids.

   Include the heatmap in your report and discuss how the samples cluster: What are the two largest clusters before everything is joined together in the dendrogram? Any other interesting clustering results, either for samples or genes?

8. Filter the expression data in `lgGo` down to just the 9 genes in your assigned gene set with the lowest `time:genotype` interaction p-values according to `DESeq`. Use the function `stripchart321g` from the provided R source file `bio321g_rnaseq_utils.R` to generate and include a stripchart showing the expression levels of these genes by time and genotype. Include the plot in your report and briefly describe in your own words how the expression levels of these genes vary across timepoint and genotype. Remember that main goal of analysis is to see how differences in expression between timepoints vary between genotypes.