Gene expression analyses from RNA-Seq data

Overview

In this exercise we will use the count data - the number of reads that map uniquely to each contig or gene - from the alignment files that you generated for each sample in the alignment exercise. The count data will serve as a proxy for the magnitude of gene expression since transcripts of greater abundance in the cell will have more reads generated from libraries prepared from RNA. Gene expression may vary among samples or individuals in your study due to your experimental design, for example control versus heat shock or high versus low intertidal zones. The biological questions you can address using the analyses below include: How many genes, if any, are differentially expressed between my treatments? Are differentially expressed genes concentrated in specific functional types of genes or are they randomly distributed with respect to function across the transcriptome?

In the analyses described below, we consider only the reads that map to only one place across the entire reference. There are a number of reasons a read may map to multiple reference sequences such as sequence similarity due to gene duplications or homologous regions across gene families, or errors in the generation of the reference sequences considering one gene as two genes. It is not possible to distinguish between the possibilities, so the most conservative approach is to only consider the reads that map to one contig or gene.

For this exercise, we will work locally, so the first step will be to copy the counts files to your own computer.

Objectives

The objectives of this section are to 1) make a combined counts data file for all individuals in the gene expression study (a column for each sample and a row for each gene), 4) normalize across individuals and identify significantly differentially expressed genes using the program DESeq implemented in R, and 5) use p-values from DESeq to identify 'transcriptome'-wide patterns of enrichment for functional classes of proteins using the software ErmineJ.

Resources

DESeq: Gene expression data analysis program. There is a very useful manual available on the website. http://www-huber.embl.de/users/anders/DESeq/

Anders, S, Huber, W. 2010. Differential expression analysis for sequence count data, *Genome Biology* 11: R106

ErmineJ: http://www.bioinformatics.ubc.ca/ermineJ/

The website is very informative. The link below describes the four input file formats.

http://www.bioinformatics.ubc.ca/ermineJ/ermineJ-help/classScore/html/formats/

The website called Quick-R (http://www.statmethods.net/) provides basic information for learning how to code in the R environment.

In this pipeline, we use the program DESeq (Anders 2010) to normalize counts and test for differences in gene expression. However, there are many other programs that perform similar functions. (e.g. EdgeR and BaySeq (Hardcastle and Kelly 2010, Robinson et al. 2010)). The package can be easily installed in R by typing:

```
source("http://www.bioconductor.org/biocLite.R")
biocLite("DESeq")
```

in the R prompt. If that does not work, a copy of the package is also located in the "R_scripts" folder on the server.

Hardcastle, T, Kelly, K. 2010. baySeq: Empirical Bayesian methods for identifying differential expression in sequence count data. *BMC bioinformatics* 11: 422.

Robinson MD, McCarthy DJ, Smyth GK. 2010. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* 26: 139.

Wang Z, Gerstein M, Snyder M. 2009. RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics* 10: 57-63.

Exercise

1) Copy the counts files from all the samples to your local drive.

a. Create a folder called "DGE_analysis" on your Desktop.

b. Copy the R scripts called DESeqScript.R and Heatmapping.R from the R_scripts folder to your local computer.

2) Extract the second column of data (number of unique reads mapped to each contig) from the _counts.txt file from each individual (generated yesterday).

a. To do this automatically, we can execute the ParseExpression2BigTable.py script while logged in to Albiorix. Note that the wildcard "*" will tell the script to process all files in your current working directory whose name ends with "counts.txt".

```
../scripts/ParseExpression2BigTable.py ContigNames.txt
CombinedCounts.txt *counts.txt
```

The ContigNames.txt file is just a list of all contig names in your assembly. To create it, you can copy and paste out the first column of any of your counts files into a new document. For today's exercise, there is already a ContigNames.txt file located in your data folder.

(It is also possible to open each counts file and copy and paste the second column into a file called CombinedCounts.txt using Excel. For a limited number of samples, like we have, it may be faster to use Excel, but if you have more data, the script becomes very useful).

b. Copy CombinedCounts.txt to your local computer drive in order to work with it in the R environment.

3) Identify differentially expressed genes using the program DESeq. In this section you will work in the R programming environment executing sequential parts of a script file. It is useful to move through the script a few lines at a time in order to learn about your data and to realize what each step of the script does.

a. Open R and open the DESeqScript.R script file from inside R.
b. In the DESeqScript.R script, change the working directory (the drag and drop trick works in R, too), enter your input file name (likely CombinedCounts.txt if you successfully used the script from step 2), enter your conditions (e.g. before and after, or whatever your conditions are for each of your samples). There should be a named condition for each of the samples or columns of data in your file, i.e. each condition corresponds to a column of data in your input file (they need to be in the same order).
c. Run the script through the line `head(countsTable)` by highlighting the lines and entering apple+return. You should see the first 6 rows for each of your columns in the table populated by your input file.
d. Run the script through `sizeFactors(cds)`. The calculated size factors are factors such that values in a column can be brought to a common scale by dividing by the corresponding size factor. Ideally you want all of the factors near 1. If you see a factor much less than 1, then there were many fewer singly mapped reads for that sample and likely fewer reads for that sample, and visa versa.
e. Run the script through `head(res)`. You should see the output columns from the results table.
f. Run the script through counting the number of significantly differentially expressed genes (the line that begins with the `nrow` function). You should see how many differentially expressed genes you have in your data set based on these conditions and at the $p < 0.05$ and $p < 0.01$ significance levels.
g. Run a few more lines of script to filter out the contigs where the average number of counts is less than 5. This filter may affect the number and fraction of significantly differentially expressed genes.
h. Now filter your results excluding contigs that have high variance (down to line ~73). Step through the lines stopping at the head and dim functions to follow how the data are being processed.
i. Now you can change the file names and generate various output items to be used for further analyses (lines ~76-110).
j. You can also make a heat map of your significantly differentially expressed genes using the `heatmap.2` function from the `gplots` library in R. You can install gplots from within R: Go to “Packages & Data,” “Package Installer,” search for gplots, select it and “install dependencies,”

then click “Install Selected.” Note that you need to be connected to the internet. To make the plot you should normalize each gene to itself, i.e. divide the counts for each individual by the average counts across all individuals – essentially relative fold difference. This will make all genes visible on a single plot even if some have two orders of magnitude higher counts, for example. You can make this new data matrix in R or Excel and save it as a .txt file. Now open the R script Heatmapping.R. Run the lines of the script sequentially as you did in the script above. Note that you will need to change the value of n in line 12. You can save the heat map image from Quartz in R though the quality is poor or you can press apple+shift+4 to capture a nice image of your heat map (it will save to the desktop by default). Or you can write a script in R to save a higher quality image.

Summary

In this section you went from individual mapped reads files (SAM) to hopefully understanding the variability in gene expression among your samples at the individual gene level and at the broader functional categorization level. You did this by counting the number of reads that mapped uniquely to each contig or gene in your reference, combining those into a single data file, then analyzing those gene expression data for differences among your treatments using DEseq.