Exercise 3. Statistical Analysis of RNA-seq Data

Part 1. Most the statistical analyses in this document are done with R. You can install R and do all the analyses on your own computer, instead of using the BioHPC computers. Here are the steps to install R for this project. You can skip this section if you will log on to BioHPC computers to do analysis.

- Install R by following instructions on this web site: https://cran.r-project.org/
- Install EdgeR on your computer. Start the R software, and in R console window, type the following commands:

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

biocLite("edgeR")
```

- If you will use your computer for data analysis, you will also need to download all data files to your laptop.

Part 2. Using EdgeR to identify differentially expressed genes.

DESeq and EdgeR are two commonly used statistics packages for analyzing RNA-seq data. In this section, we will use DESeq to detect differentially expressed genes.

The instructions here assume that you use the BioHPC computer. If you use your own computer, you might need to set the R working directory (From R menu File->Change Dir) to point to where the data files are.

We will use the `gene_count.txt` created from the first exercise of the RNA-seq workshop. From the ssh terminal, type "R" and press return. Now, you are in R console.

Run the following commands in R console, which include the following steps:

1. Load the EdgeR libraries;
2. Read in the gene_count.txt file, and add the column header;
3. Specify sample groups;

4. Normalization;
5. Remove genes with no expressions;
6. Identify differentially expressed genes.

```
library("edgeR")

x <- read.delim("gene_count.txt", header=F, row.names=1)

colnames(x)<-c("WTa","WTb","MUa","MUb")

group <- factor(c(1,1,2,2))

y <- DGEList(counts=x,group=group)

y <- calcNormFactors(y)

# only keep genes with cpm value greater than 1 in at least 2
samples

keep <-rowSums(cpm(y)>=1) >=2

y<-y[keep,]

# write the CPM values into a tab-delimited text file

# cpm stands for counts per million.

d <- cpm(y)

write.table(d, "CPM.txt", sep="\t")

design<-model.matrix(~0+group)

y <- estimateGLMCommonDisp(y,design)

y <- estimateGLMTrendedDisp(y,design)

y <- estimateGLMTagwiseDisp(y,design)

fit<-glmFit(y,design)

lrt.2vs1 <- glmLRT(fit, contrast=c(-1,1))

top2v1 <- topTags(lrt.2vs1, n=5000)

write.table(top2v1, "diff2-1.txt", sep="\t")
```

Part 3. Clustering

1. Install the following software on your laptop computer. The two software are available for both Windows and Mac version.
   Cluster 3.0: http://bonsai.hgc.jp/~mdehoon/software/cluster/software.htm
   Java TreeView: http://sourceforge.net/projects/jtreeview/files/jtreeview/

2. Prepare the data files.
   - In this project, RNA-seq experiments were done on samples representing four different maize leaf development stages, from young to mature leaf tissues (Sample names are s1, s2, s3, and s4. *Nat Genet.* 2010, 42:1060-7). Using FileZilla to download the files from the BioHPC server, in the directory "/shared_data/RNAseq/exercise3/". There are two files in the directory: "genes.txt" and "maize.annot". The "genes.txt" file is a table of normalized expression levels.
   - You can examine the file "genes.txt" with Excel. When working with your own data, this file can be created with Excel.
   - Add +1 to all values in the excel table, so that there is no "0" in the table. This step is necessary as we will do log transformation later.
   - Save the Excel file as a "Text (tab delimited)" file.
3. Run Hierarchical Cluster.
   - Open the data file in Cluster.
   - Do Log transformation: click "Adjust Data" tab; check "Log transform data"; click "Apply".
   - Filter Data: click "Filter Data" tab; check the filter item 3 and 4; change "observations with abs(Val)>=" to 5; change "MaxVal-MinVal>=" to 2.0. Click "Apply filter", followed by "Accept Filter". As the filter is applied on log transformed data, this setting will keep the genes with FPKM value above 2^5 in at least one of the samples, and fold change above 2^2 between highest and lowest samples. The filtering is arbitrary, we would like to keep the filtered gene number below 10,000 after filtering.

- Center genes: click the “Adjust Data” tab; uncheck the “Log transform data” box if it is still checked, as we do not want to do log transformation twice; check “Center genes” and “Median”; Do NOT check “Normalize genes” as the FPKM values are already normalized. Click “Apply”.
- Run Hierarchical clustering: check “Cluster” both under Genes and Arrays; click “Average linkage”.

4. Run K-Means cluster
    - Click the “K-Means” tab.
    - Check “Organize genes”. Change “number of clusters(k)” to 12. Click “Execute”. There is a new “.kgg” file is created, you can open this file in Excel. This file has two columns: gene name and cluster ID. In this file, all the genes in your data set were separated into 12 clusters based on their expression patterns across the 4 samples. The “cluster ID” column indicates what cluster each gene is in.
5. Visualize the hierarchical clustering results.
    - Open the “.CDT” file in TreeView. The manual is available at http://jtreeview.sourceforge.net/manual.html.