Introduction to expression analysis
(RNA-seq)

Transcript quantification using Salmon and differential expression analysis using baySeq

Philippine Genome Center
University of the Philippines

Prepared by
Nelzo C. Ereful
National Institute of Agricultural Botany
Cambridge, UK

General information

The following standard icons are used in the hands-on exercises to help you locate:

ℹ	Important Information
📄	General information / notes
👣	Follow the following steps
Q	Questions to be answered
❗	Warning – PLEASE take care and read carefully
🟩	Optional Bonus exercise
🟥	Optional Bonus exercise for a champion

Resources used

Salmon: not the fish but … https://combine-lab.github.io/salmon/

baySeq: http://bioconductor.org/packages/release/bioc/html/baySeq.html

Reference used

Most of these notes are lifted from these references:

Thomas J. Hardcastle. 2015/2016. baySeq: Empirical Bayesian analysis of patterns of differential expression in count data. From www.bioconductor.org (vignette)

Thomas J Hardcastle and Krystyna A Kelly. 2010. baySeq: Empirical Bayesian methods for identifying differential expression in sequence count data. BMC Bioinformatics.

In this module, we will use Salmon to quantify transcript expression. We will then calculate differential expression using baySeq between two *Claviceps* samples – Control and Germinating – each with three replicates. Ideally, we need biological replicates (two or more for RNA-seq) to improve statistical power. For demo purposes and for the interest of time, we will quantify transcript expression in only one replicate.

Before performing differential expression, we need to estimate transcript expression for each gene or isoform of each sample replicate. One of the tools to perform this function is Salmon.
Salmon gets the best results in 'read' mode where you give it your fastq files. First you build an index of your cDNA reference transcriptome and then align the reads and get expression quantification estimates. For more information on this mode, go to the link above.
In our case, since we already obtained the alignment outputs, we will use the alignment-based mode of salmon.
Using the **bam** output you generated from the bowtie alignment (remapping), run the command below. The arguments in this command include:

--alignments	input alignment (BAM) file(s)
--**targets**	FASTA format file containing target transcripts
--**biasCorrect**	Perform sequence-specific bias correction

```
salmon quant --libType U \
     --alignments alignments.bam \
     --targets /path/to/ Claviceps.new.output.fa \
     --threads 2 \
     --biasCorrect \
     --useErrorModel \
     --output C.purpurea.salmon
```

To open the output, on the command line type in:

```
$ cd C.purpurea.salmon
$ less quant.sf
```

To extract the NumReads column, use the following command:

cut –f1,4 quant.sf > NumRead.sf
where 1 and 4 are columns corresponding to the annotation and the read counts, respectively. These values depend on the version of salmon you are using.

Stop. Instructions will be given after you generated the results.

You will be given a CSV file.
Using R, you can manipulate the data using the following commands:

Open R (or R in Linux)

Load the baySeq package

- library(baySeq)

Because high-throughput sequencing (RNA-seq) experiments are usually massive, we will use parallel processing as implemented by the snow package. If parallel is not present we can proceed with NULL cluster

- if(require("parallel")) cl <- makeCluster(2) else cl <- NULL

#data frame

- df <- read.csv("CvW_ReadCount.csv", header=TRUE)
- df[1:5,]

  #sample view

	Name	C1	C2	C3	W2	W4	W5
#1	Traes_7DS_FFE9ACDAB.2	246	32	703	128	361	137
#2	Traes_7DS_FFA36F6DA.1	12	7	18	12	6	5
#3	Traes_7DS_FF9F1CF23.1	0	0	0	0	0	0
#4	Traes_7DS_FF911FA4A.1	15	14	20	8	29	6
#5	Traes_7DS_FF7C9C6FD.1	0	0	2	1	2	0

We assume that the read counts show differential expression between the first three libraries (replicates) and the last three libraries. Our replicate structure, used to estimate the prior distributions on the data, can thus be defined as

- replicates <- c("C1", "C2","C3", "W2","W4","W5")

We expect that some tags will be equivalently expressed in all libraries (this corresponds to NDE model); some tags will show differential expression (DE)

- groups <- list(NDE=c(1,1,1,1,1,1), DE = c(1,1,1,2,2,2))

Combine the count data

- CD <- new("countData", data = as.matrix(df[,2:7]), replicates = replicates, groups = groups)

We can optionally add annotation

- CD@annotation <- data.frame(name = df[1])

#calculate the number of entries or rows

- length(CD@data[,1])

#determine summary statistics

- summary(CD@data)

#generate histogram – distribution curve

- hist(log(CD@data[,1]))

#generating figures (histogram, MA, scatterplots)

```
jpeg("Histogram_count.jpg")
hist(log(CD@data[,1]))
dev.off()
```

pdf("Histogram_count.pdf")
hist(log(CD1@data[,1]))
dev.off()

jpeg("Histogram_plotMA.jpg")
plotMA.CD(CD, samplesA = "C", samplesB = "B", col = c(rep("red", 10000), rep("black", 90000)))
dev.off()
#compute correlation between samples or replicates

- cor(CD@data[,1],CD@data[,2])

#Pre-filtering

- CD1 <- CD[rowSums(CD@data)>0,]

#Threshold value for pre-filtering (the number of columns of the data will be the threshold value)

- CD1 <- CD[rowSums(CD@data) > ncol(CD),]

Open R (if you haven't done so)

#Libray sizes can be defined from the data

- libsizes(CD1) <- getLibsizes(CD1)

#Calculate priors

- CD1 <- getPriors.NB(CD1, cl = cl)

#Acquire posterior likelihoods

- CD1 <- getLikelihoods(CD1, cl = cl)

After performing differential expression analysis, the following script will list all DE genes with FDR lower than 0.05:

- topCounts(CD1, group="DE", FDR = 0.05, normaliseData=TRUE)

If you want to save the output into a file, use:

- table = topCounts(CD1, group="DE", number=nrow(df),
  + normaliseData=TRUE)
- write.table(table, file = "CvW_result_DE.txt", col.names = T, row.names +
  = T, sep = "\t")

The preceding approach allows you to execute commands one at a time. What if you want to execute all commands in one run without typing in each command in the command line successively? Fortunately, we can do this by using Rscript.

Put the following commands in an Rscript command file:

```
$ nano CvW_DE_script

#alternatively, you can use vi, instead of nano
#the content of the script should be:

library(baySeq)
if(require("parallel")) cl <- makeCluster(2) else cl <- NULL
df <- read.csv("WvC_ReadCount2Final.csv", header=TRUE)
replicates <- c("C", "C","C", "W","W","W")
groups <- list(NDE=c(1,1,1,1,1,1), DE = c(1,1,1,2,2,2))
CD <- new("countData", data = as.matrix(df[,2:7]), replicates = replicates,
+       groups = groups)
CD@annotation <- data.frame(name = df[1])
CD1 <- CD[rowSums(CD@data)>6,]
libsizes(CD1) <- getLibsizes(CD1)
CD1 <- getPriors.NB(CD1, cl = cl)
CD1 <- getLikelihoods(CD1, cl = cl)
table = topCounts(CD1, group="DE", number=nrow(df), normaliseData=TRUE)
write.table(table, file = "CvW_result_DE.txt", col.names = T,
+       row.names = T, sep = "\t")


#end
```

Check if you have the right parameter

```
#save your script using Control + O
#exit using Control + X
```

Create a bash file which runs the script using Rscript and an email message that tells you that the script is done

```
$ nano bash_Rscript.sh

#Rscript contains the file name of your script above
#Optionally, provide your email address so you will receive a message when analysis is
#done

Rscript MBvNB_DE_script

mail -s "subject:bayseq" example@gmail.com <<< "content: run is finished"

#save your script using Control + O
#exit using Control + X
```

Now, change the bash script to 'executable'

```
$ chmod +x bash_Rscript.sh
```

Run the executable bash script

```
$ nohup ./bash_Rscript &
```

With 'nohup' you can shut down your machine, have a break, check your FB account and wait until the analysis is done. You should receive an email message telling you that the "run is finished".

Downstream analysis

Manipulating output

1. Computing for Expression Ratio, fold change or Log2FC.
2. Identify repressed and induced genes.

Create a Heatmap to visualise changes in gene expression.

Note: Reference materials used in this program are attributed to the original authors.

Sequencing reads in this exercise are used with permission.

Because the read files are proprietary (owned by the National Institute of Agricultural Botany - Cambridge), kindly erase them including intermediate files generated if no longer in use. This will be made publicly available on a database (e.g. NCBI-SRA).

Thank you and we hope you enjoyed the exercise.

Don't hesitate to ask any questions and feel free to contact us any time (email address: nelzo.ereful@niab.com).