Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

Introductions to Bioconductor:

Bioconductor is a project based on R, whose aim is to develop and support open source software in bioinformatics. It has defined clear workflow for performing multiple common biological data analysis tasks. It also includes many specialized packages based to process biological information efficiently compared to traditional R programming.

Install Bioconductor

- Bioconductor 3.16, core packages attached

```
if (!require("BiocManager", quietly = TRUE))
install.packages("BiocManager")
BiocManager::install(version = "3.16")
```

- Install other packages in Bioconductor

```
BiocManager::install(c("pkg1","pkg2”,...))
```

RNA-seq Analysis

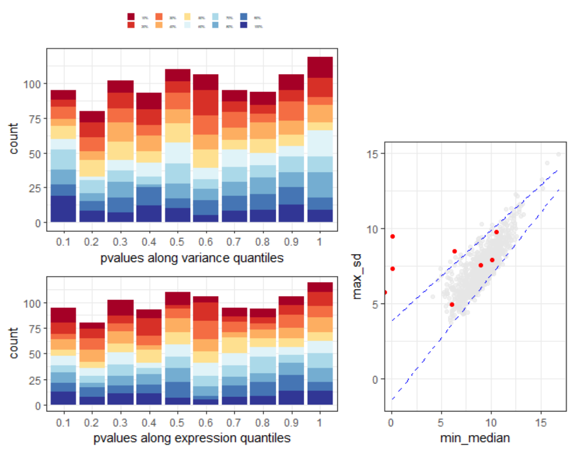
Introduction: RNA-seq adopts next generation sequencing technique to present the frequency of gene expression by measuring the quantity of RNA in a biological sample. It generates a large number of sequencing data. We can use packages of Bioconductor to perform sequencing file download, quality control, gene filter, expression pattern detection, and part of them can be visualized.

Download sequencing files (.fastq) with GEO accession number

```
library(GEOfastq)
library(R.utils)
#Download RNA seq sample with GEO accession number
gse_name <- 'GSE133758'
gse_text <- crawl_gse(gse_name)
gsm_names <- extract_gsms(gse_text)
#file too large, use first sample as example.
srp_meta <- crawl_gsms(gsm_names[1])
#Download fastqq to current dirctory
data_dir <- getwd()
SRR_name<-srp_meta[1,1]
res <- get_fastqs(srp_meta, data_dir)
SRR_name<-paste(SRR_name, "fastq.gz", sep = ".")
gunzip(SRR_name)
```

Perform **quality control** on sample

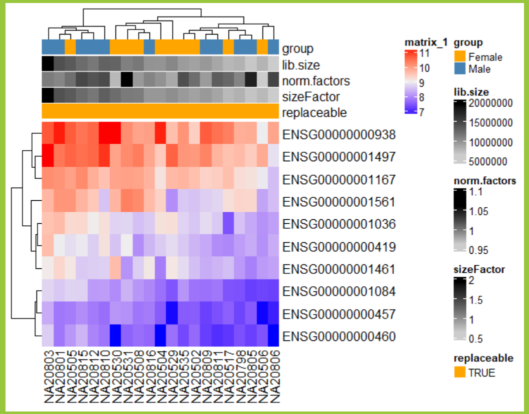
```
BiocManager::install("DEGreport")
BiocManager::install("DESeq2")
library(DEGreport)
data(humanGender)
Library(DESeq2)
index <- c(1:10, 75:85)
sample <-
DESeqDataSetFromMatrix(assays(humanGender)[[1]][1:1000, index],colData(humanGender)[index,],
design=~group)
sample <- DESeq(sample)
res <- results(sample)
counts <- counts(sample, normalized = TRUE)
design <- as.data.frame(colData(sample))
degQC(counts, design[["group"]], pvalue =
res[["pvalue"]])
```



QC plot

Filter Genes by the group, based on read counts in sample

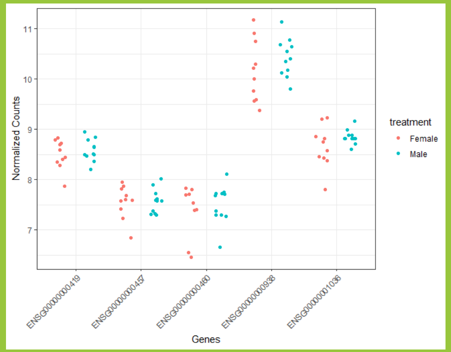
```
filter_count <- degFilter(counts(sample),
                          design, "group",
                          min=1, minreads = 50)
library(ComplexHeatmap)
th <- HeatmapAnnotation(df = colData(sample),
                        col = degColors(colData(sample), TRUE))
Heatmap(log2(counts(sample) + 0.5)[1:10,],
        top_annotation = th)
```



Gene Group results

Plot top genes counts in sample by group

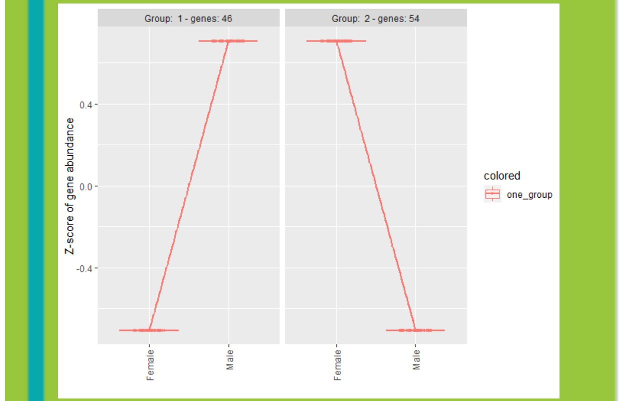
```
degPlotWide(dds, rownames(sample)[1:5],
group="group")
```



Normalized gene counts

Detect patterns of Expression on sample

```
ma = assay(rlog(sample))[row.names(res)[1:100],]
res <- degPatterns(ma, design, time = "group")
```



Expression Pattern Analysis