

# Reproducible Analysis of RNA-Seq Data – SRP033351

Mengyuan Kan

July 31, 2018

## Overview

Use RNA-Seq data from SRA study SRP033351 as an example.

The goal of taffeta is to perform reproducible analysis and validation of RNA-Seq data, as a part of [RAVED pipeline](#)<sup>1</sup>:

- Download SRA .fastq data
- Perform preliminary QC
- Align reads to a reference genome
- Perform QC on aligned files
- Create a report that can be used to verify that sequencing was successful and/or identify sample outliers
- Perform differential expression of reads aligned to transcripts according to a given reference genome
- Create a report that summarizes the differential expression results

Generate LSF scripts in each step for HPC use.

## Informatics Tools

RNA-Seq data analysis is performed on HPC. Directly use softwares that are already installed.

Check and load pre-installed softwares. For example:

---

```
module avail
module load Trimmomatic-0.32
```

---

BASH

Uninstalled softwares or those that need re-configuration will be installed locally.

## Raw reads process

### trimmomatic

- usage: trim raw reads
- version: trimmomatic-0.32
- location: /opt/software/Trimmomatic/0.32/trimmomatic-0.32.jar

### FastQC

Needs re-configuration. Use a customized contaminant\_list.txt with updated adapter and primer sequences.

- usage: report reads quality
- version: FastQC v0.11.7
- location: /home/mengykan/.local/bin/
- local installation:

---

BASH

---

```
cd ~/softwares
wget https://www.bioinformatics.babraham.ac.uk/projects/fastqc/fastqc_v0.11.7.zip
unzip fastqc_v0.11.7.zip
cd FastQC/
chmod 775 fastqc
ln -s /home/mengykan/softwares/FastQC/fastqc /home/mengykan/.local/bin/
```

---



### Important

Prepare for the **contaminant\_list.txt** in FastQC configuration with most updated adapter and primer sequences. We provide a list of sequences in **template\_files/rnaseq\_adapter\_primer\_sequences.txt**. Add the last two columns (i.e. Description and Sequence without header line) to the FastQC configuration file `~/softwares/FastQC/Configuration/contaminant_list.txt`. Users can also add the sequences from their own protocol.

## Align and mapping

### STAR

- usage: align and mapping RNA-Seq reads
- version: STAR\_2.5.2b
- location: /project/bhimeslab/STAR-2.5.2b/bin/Linux\_x86\_64/STAR. Create symbolic link to local bin:

---

BASH

---

```
ln -s /project/bhimeslab/STAR-2.5.2b/bin/Linux_x86_64/STAR /home/mengykan/.local/bin/
```

---

## samtools

- usage: sort, index, statistics
- version: 1.8 (the current version has multi-thread option for .bam file index and sort)
- location: /home/mengykan/.local/bin
- local installation:

---

BASH

---

```
cd ~/softwares
wget https://github.com/samtools/samtools/releases/download/1.8/samtools-1.8.tar.bz2
tar xvjf samtools-1.8.tar.bz2
cd samtools-1.8
./configure --prefix=/home/mengykan/.local # local installation
make
make install
```

---

## Bam file QC metrics

### bamtools

- usage: manipulate bam files
- version: 2.3.0
- location: /opt/software/bamtools/2.3.0/bin/bamtools. Add bamtools library path /opt/software/bamtools/2.3.0/lib to LD\_LIBRARY\_PATH in .bashrc if it is not there, otherwise will get the error *error while loading shared libraries: libbamtools.so.2.3.0: cannot open shared object file: No such file or directory*

### picard

- usage: stats of RNA metrics and insertsize metrics in RNA-Seq data
- version: picard-tools-1.96
- location: /opt/software/picard/picard-tools-1.96

## Gene and transcript-level quantification

### HTseq

- usage: count mapping reads for DESeq2 DE identification
- version: HTSeq-0.6.1
- location: /home/mengykan/.local/bin/htseq-count
- local installation:

---

BASH

---

```
cd ~/softwares
wget https://pypi.python.org/packages/72/0f/566afae6c149762af301a19686cd5fd1876deb2b48d09546dbd5caeabb78/HTSeq-0.6.1.tar.gz#md5=b7f4\
f38a9f4278b9b7f948d1efbc1f05
tar -zxvf HTSeq-0.6.1.tar.gz
cd HTSeq-0.6.1
python setup.py build
python setup.py install --install-platlib=$HOME/.local/python_lib --install-scripts=$HOME/.local/bin
```

---

## kallisto

- usage: pseudoalign and quantify transcript
- version: 0.44.0
- location: /project/bhimeslab/kallisto\_linux-v0.42.3/kallisto (symbolic to ~/.local/bin/kallisto)

---

BASH

---

```
cd /project/bhimeslab/softwares
wget https://github.com/pachterlab/kallisto/releases/download/v0.44.0/kallisto_linux-v0.44.0.tar.gz
tar zxvf kallisto_linux-v0.44.0.tar.gz
cd kallisto_linux-v0.44.0
ln -s /project/bhimeslab/softwares/kallisto_linux-v0.44.0/kallisto /home/mengykan/.local/bin/kallisto
```

---

## DE analysis

### sleuth

- sleuth R package usage: identify DE and visualize results
- version: 0.30.0, dependency rhdf5 2.22.0
- location: /home/mengykan/.local/R-3.4/libs

### local installation

#### 1. set R local enviroment

add R enviromental path in .bashrc

```
export R_LIBS=/home/mengykan/.local/R-3.4/libs
```

#### 2. install rhdf5

---

R

---

```
source("http://bioconductor.org/biocLite.R")
biocLite("rhdf5")
```

---

#### 3. install devtools

---

R

---

```
install.packages("devtools")
```

---

## 4. install sleuth

---

R

---

```
library(devtools)
withr::with_libpaths(new="/.local/R/libs", install_github('pachterlab/sleuth'))
```

---

## DESeq2

- usage: Gene-based DE analysis and results visualization
- version: 1.18.1
- location: \$HOME/.local/R/libs

### 1. pre-install r package RcppArmadillo [optional]



#### Note

Install DESeq2 first to check if this package is installed already. The latest version 0.7.500 requires g++ version 4.6 or greater. Check *module avail* and load a higher g++ version *module load gcc/6.2.1*

### 2. local installation

---

R

---

```
source("http://bioconductor.org/biocLite.R")
biocLite("DESeq2")
```

---

## Visualization

### R packages

- genefilter
- gplots: heatmap2 plots

## Reference Genome

### Human genome reference files

- indexed reference genome with ERCC spike-in: hg38/genome.ERCC.fa and hg38/genome.ERCC.fa.fai
- known gene/transcript annotations:
  - ▶ hg38/genes.gtf (human genes)
  - ▶ ERCC92.gtf (ERCC spike-in)
  - ▶ hg38/genes.ERCC.gtf (human genes with ERCC spike-in by concatenating the above two)
- rRNA annotations: hg38/rRNA\_hg38.gtf
- [refFlat format](#)<sup>2</sup> position file used through Picard command to generate RNA-Seq metrics: hg38/refFlat.txt

## STAR index creation

Create reference genome index for STAR if it does not exist. [STAR tutorial](#) <sup>3</sup> recommended to remove all files from the genome directory before running the genome generation step. Create a new directory STAR\_index under previous reference folder.

---

BASH

---

```
mkdir /project/bhimeslab/Reference/hg38/STAR_index
cd /project/bhimeslab/Reference/hg38/STAR_index
STAR --runThreadN 12 --runMode genomeGenerate --genomeDir /project/bhimeslab/Reference/hg38/STAR_index --genomeFastaFiles /project/bhimeslab/Reference/hg38/genome.ERCC.fa --sjdbGTFfile /project/bhimeslab/Reference/hg38/genes.ERCC.gtf
```

---

- 15 files generated: chrLength.txt, chrNameLength.txt, chrName.txt, chrStart.txt, exonGeTrInfo.tab, exon-Info.tab, geneInfo.tab, Genome, genomeParameters.txt, SA, SAindex, sjdbInfo.txt, sjdbList.fromGTF.out.tab, sjdbList.out.tab, transcriptInfo.tab
- --sjdbOverhang default is 100

## Kallisto index

Create reference genome index for kallisto if it does not exist. Kallisto indexing is very fast.

---

BASH

---

```
cd /project/bhimeslab/Reference/hg38
kallisto index -i hg38_new.idx /project/bhimeslab/Reference/hg38/Homo_sapiens.GRCh38.rel179.cdna.all.fa
```

---

## RNA-Seq Pipeline

### GitHub structure

- **pipeline.scripts:** Python scripts should be added in an executable search path.
- **template\_files:** Rmd template and other text files used in the pipeline. Put into a template directory specified as `template_dir`.
- **example\_files:** output phenotype file, RMD scripts and output HTML report for this example.
- **miscellaneous:** random useful scripts and files not specify in this pipeline



### Note

Edit `pipeline.scripts/rnaseq_userdefine_variables.py` with user-defined variables before add it to an executable search path.

## SRA download and fastqc

### run command line

**pipeline\_scripts/rnaseq\_sra\_download.py**: download .fastq files from SRA.

Read in **template\_files/rnaseq\_sra\_download\_Rmd\_template.txt** from specified directory `template_dir` to create a RMD script.

Ftp addresses for corresponding samples are obtained from SRA SQLite database using R package SRADB.

If .fastq files with the same names exist in the directory, skip downloading.

---

BASH

```
rnaseq_sra_download.py --geo_id GSE52778 --path_start /home/mengykan/Projects/SRP033351 --project_name SRP033351 --template_dir /home/mengykan/Projects/shared_files/RNASeq --fastqc
```

---



### Note

Check the error (.screen) files to see if the ftp address is available. For example, SRR1039513.3.fastq.gz is in sqlite database but not in the ftp `ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR103/003/SRR1039513`

### script options

The option `--pheno_info` refers to using user provided SRA ID for download which is included in the SRA\_ID column in the provided phenotype file. If the phenotype file is not provided, use phenotype information from GEO. SRA\_ID is retrieved from the field relation.1.

The option `--fastqc` refers to running FastQC for downloaded .fastq files.

### submit LSF script

Generate LSF scripts for each download .fastq file. Submit LSF jobs on HPC that enables to run in parallel.

---

BASH

```
for i in *_download.lsf; do bsub < $i; done
```

---

```
cat SRR1039508_1_download.lsf
```

---

OUTPUT

```
#!/bin/bash
#BSUB -L /bin/bash
#BSUB -J SRR1039508_1_download
#BSUB -q normal
#BSUB -o SRR1039508_1_download_%J.out
#BSUB -e SRR1039508_1_download_%J.screen
#BSUB -M 36000
#BSUB -n 1
wget ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR103/008/SRR1039508/SRR1039508_1.fastq.gz
fastqc /home/mengykan/Projects/SRP033351/SRP033351_SRAdownload/SRR1039508_1.fastq.gz -o /home/mengykan/Projects/SRP033351/SRR1039508
```

---

## output files

Output files are saved in ~/Projects/SRP033351/SRP033351\_SRAdownload:

- GEO phenotype GSE52778\_withoutQC.txt
- SRA download information SRP033351\_sraFile.info
- The RMD and corresponding HTML report files:
  - ▶ SRP033351\_SRAdownload\_RnaSeqReport.Rmd
  - ▶ SRP033351\_SRAdownload\_RnaSeqReport.html
- Raw .fastq files downloaded from SRA
- FastQC results are saved under each sample folder

## User-tailored phenotype file preparation

The sample info file used in the following steps should be provided by users.

### required columns

- 'Sample' column containing sample ID
- 'Status' column containing variables of comparison state
- 'R1' and/or 'R2' columns containing full paths of .fastq files

### other columns

'Treatment', 'Disease', 'Donor' (i.e. cell line ID if *in vitro* treatment is used), 'Tissue', 'ERCC\_Mix' (i.e. ERCC mix ID if ERCC spike-in sample is used), 'protocol' designating sample preparation kit information.

### Index column

'Index' column contains index sequence for each sample. If provided, trim raw .fastq files based on corresponding adapter sequences.

If use data from GEO, most GEO phenotype data do not have index information. However, FastQC is able to detect them as *Overrepresented sequences*. Users can tailor the 'Index' column based on FastQC results. We provide a file with most updated adapter and primer sequences for FastQC detection.

### Sample file

An example phenotype file can be found here [example\\_files/sample\\_info\\_file.txt](#). Use this file in the following steps.

SRA_ID	Sample	Index	GEO_ID	Donor	Tissue	Treatment	ERCC_Mix	Protocol
SRX384345	SRR1039508	CGATGT	GSM1275862	N61311	ASM	untreated	-	TruSeq_RNA_Sample_Prep_Kit.v2
SRX384346	SRR1039509	TGACCA	GSM1275863	N61311	ASM	dex	-	TruSeq_RNA_Sample_Prep_Kit.v2
SRX384347	SRR1039510	ACAGTG	GSM1275864	N61311	ASM	alb	-	TruSeq_RNA_Sample_Prep_Kit.v2
SRX384348	SRR1039511	GCCAAT	GSM1275865	N61311	ASM	alb_dex	-	TruSeq_RNA_Sample_Prep_Kit.v2
SRX384349	SRR1039512	CAGATC	GSM1275866	N052611	ASM	untreated	-	TruSeq_RNA_Sample_Prep_Kit.v2



Table continued

Status	R1	R2
healthy_untreated	/path_to_file/SRR1039508.1.fastq.gz	/path_to_file/SRR1039508.2.fastq.gz
healthy_dex	/path_to_file/SRR1039509.1.fastq.gz	/path_to_file/SRR1039509.2.fastq.gz
healthy_alb	/path_to_file/SRR1039510.1.fastq.gz	/path_to_file/SRR1039510.2.fastq.gz
healthy_alb_dex	/path_to_file/SRR1039511.1.fastq.gz	/path_to_file/SRR1039511.2.fastq.gz
healthy_untreated	/path_to_file/SRR1039512.1.fastq.gz	/path_to_file/SRR1039512.2.fastq.gz



### Important

Column naming is rigid for the following columns: 'Sample', 'Status', 'Index', 'R1', 'R2', 'ERCC\_Mix', 'Treatment', 'Disease', 'Donor', because pipeline scripts will recognize these name strings, but the column order can be changed.

## Align and quantification

### run command line

Run **pipeline\_scripts/rnaseq\_align\_and\_qc.py** to: 1) trim adapter and primer sequences if index information is available, 2) run FastQC for (un)trimmed .fastq files, 3) align reads and quantify reads mapped to genes/transcripts, and 5) obtain various QC metrics from .bam files.

Edit **pipeline\_scripts/rnaseq\_userdefine\_variables.py** with a list user-defined variables (e.g. paths of genome reference file, paths of bioinformatics tools, versions of bioinformatics tools), and save the file under an executable search path.

If perform adapter trimming, read in **template\_files/rnaseq\_adapter\_primer\_sequences.txt** from specified directory `template_dir` used as a reference list of index and primer sequences for various library preparation kits.

BASH

```
rnaseq_align_and_qc.py --project_name SRP033351 --samples_in /home/mengykan/Projects/SRP033351/files/sample_info_file.txt --aligner \
star --ref_genome hg38 --library_type PE --index_type truseq_single_index --strand nonstrand --path_start /home/mengykan/Projects/SR\
P033351 --template_dir /home/mengykan/Projects/shared_files/RNaseq
```

### script options

The `--library_type` option refers to PE (paired-end) or SE (single-end) library.

The `--index_type` option refers to index used in sample library preparation. The index types provided in **template\_files/rnaseq\_adapter\_primer\_sequences.txt** are: `truseq_single_index` (TruSeq Single Indexes), `illumina_ud_sys1` (Illumina UD indexes for NovaSeq, MiSeq, HiSeq 2000/2500), `illumina_ud_sys2` (Illumina UD indexed for MiniSeq, NextSeq, HiSeq 3000/4000).

**template\_files/rnaseq\_adapter\_primer\_sequences.txt:**

- contains four columns (i.e. Type, Index, Description, Sequence). Sequences in the Index column is used to match those in Index column in sample info file. This column naming is rigid.
- based on the following resources:
  - [illumina adapter sequences](#) <sup>4</sup>
- If users provide new sequences, add the new index type in the 1st column 'Type' and specify it in index.type.

The --strand option refers to sequencing that captures both strands (nonstrand) or the 1st synthesized strand (reverse) or the 2nd synthesized strand (forward) of cDNA. If the 2nd strand is synthesized using dUTP, this strand will extinct during PCA amplification, thus only 1st (reverse) strand will be sequenced.



## Important

Read sample preparation protocol carefully. Reads not in the specified strand will be discarded. Double check proportion of reads mapped to no feature category in QC report. If a lot of reads are mapped to 'no feature', the strand option setting is likely incorrect.

## submit LSF script

LSF scripts are generated for each sample. Submit LSF jobs on HPC that enables to run in parallel.

BASH

```
for i in *_align.lsf; do bsub < $i; done
```

## Check one sample LSF

```
cat SRR1039508_align.lsf
```

OUTPUT

```
#!/bin/bash
#BSUB -L /bin/bash
#BSUB -J SRR1039508_align
#BSUB -q normal
#BSUB -o SRR1039508_align_%J.out
#BSUB -e SRR1039508_align_%J.screen
#BSUB -M 36000
#BSUB -n 12
cd /home/mengykan/Projects/SRP033351/SRR1039508/
java -Xmx1024m -classpath /opt/software/Trimmomatic/0.32/trimmomatic-0.32.jar org.usadellab.trimmomatic.TrimmomaticPE -phred33 /home/mengykan/Projects/SRP033351/SRP033351_SRAdownload/SRR1039508_1.fastq.gz /home/mengykan/Projects/SRP033351/SRP033351_SRAdownload/SRR1039508_2.fastq.gz /home/mengykan/Projects/SRP033351/SRR1039508/SRR1039508_R1_Trimmed.fastq R1_Trimmed_Unpaired.fastq /home/mengykan/Projects/SRP033351/SRR1039508/SRR1039508_R2_Trimmed.fastq R2_Trimmed_Unpaired.fastq ILLUMINACLIP:/home/mengykan/Projects/SRP033351/SRR1039508/SRR1039508_adapter.fa:2:30:10 MINLEN:40 fastqc -o /home/mengykan/Projects/SRP033351/SRR1039508/ /home/mengykan/Projects/SRP033351/SRR1039508/SRR1039508_R1_Trimmed.fastq /home/mengykan/Projects/SRP033351/SRR1039508/SRR1039508_R2_Trimmed.fastq
cat /home/mengykan/Projects/SRP033351/SRR1039508/SRR1039508_R1_Trimmed.fastq | awk '((NR-2)%4==0){read=$1;total++;count[read]++;}END{for(read in count){if(count[read]==1){unique++;};print total,unique,unique*100/total}' > /home/mengykan/Projects/SRP033351/SRR1039508/SRR1039508_ReadCount
cat /home/mengykan/Projects/SRP033351/SRR1039508/SRR1039508_R2_Trimmed.fastq | awk '((NR-2)%4==0){read=$1;total++;count[read]++;}END{for(read in count){if(count[read]==1){unique++;};print total,unique,unique*100/total}' >> /home/mengykan/Projects/SRP033351/SRR1039508/SRR1039508_ReadCount
mkdir /home/mengykan/Projects/SRP033351/SRR1039508/star_out
cd /home/mengykan/Projects/SRP033351/SRR1039508/star_out
```

```
STAR --genomeDir /project/bhimeslab/Reference/hg38/STAR_index --runThreadN 12 --outReadsUnmapped Fastx --outMultimapperOrder Random --outSAMmultNmax 1 --outFilterIntronMotifs RemoveNoncanonical --outSAMstrandField intronMotif --outSAMtype BAM SortedByCoordinate --readFilesIn /home/mengykan/Projects/SRP033351/SRR1039508/SRR1039508_R1_Trimmed.fastq /home/mengykan/Projects/SRP033351/SRR1039508/SRR1039508_R2_Trimmed.fastq
mv Aligned.sortedByCoord.out.bam accepted_hits.bam
mkdir /home/mengykan/Projects/SRP033351/SRR1039508/htseq_out/
samtools view accepted_hits.bam | htseq-count -r pos --stranded=no - /project/bhimeslab/Reference/hg38/genes.ERCC.gtf > /home/mengykan/Projects/SRP033351/SRR1039508/htseq_out/SRR1039508_counts.txt
samtools sort accepted_hits.bam -@12 -T SRR1039508.tmp -o SRR1039508_accepted_hits.sorted.bam
samtools index -@12 SRR1039508_accepted_hits.sorted.bam
samtools idxstats SRR1039508_accepted_hits.sorted.bam > SRR1039508_accepted_hits.sorted.stats
bamtools stats -in SRR1039508_accepted_hits.sorted.bam > SRR1039508_accepted_hits.sorted.bamstats
java -Xmx2g -jar /opt/software/picard/picard-tools-1.96/CollectRnaSeqMetrics.jar REF_FLAT=/project/bhimeslab/Reference/hg38/refFlat.txt STRAND_SPECIFICITY=NONE VALIDATION_STRINGENCY=LENIENT INPUT=SRR1039508_accepted_hits.sorted.bam OUTPUT=SRR1039508_RNASeqMetrics
echo "Junction Spanning Reads: " $(bamtools filter -in SRR1039508_accepted_hits.sorted.bam -script /home/mengykan/Projects/SRP033351/SRR1039508/cigarN.script | bamtools count ) >> SRR1039508_accepted_hits.sorted.bamstats
java -Xmx2g -jar /opt/software/picard/picard-tools-1.96/CollectInsertSizeMetrics.jar VALIDATION_STRINGENCY=LENIENT HISTOGRAM_FILE=SRR1039508_InsertSizeHist.pdf INPUT=SRR1039508_accepted_hits.sorted.bam OUTPUT=SRR1039508_InsertSizeMetrics
rm accepted_hits.bam
```

---

## output files

Various output files will be written for each sample in directories structured such as:

- Sample-level directory /home/mengykan/Projects/SRP033351/SRR1039508/SRR1039508
- Trimemd and FastQC files are:
  - ▶ SRR1039508\_R1\_Trimmed.fastq
  - ▶ SRR1039508\_R2\_Trimmed.fastq
  - ▶ SRR1039508\_R1\_Trimmed\_fastqc.zip
  - ▶ SRR1039508\_R2\_Trimmed\_fastqc.zip
  - ▶ SRR1039508\_ReadCount
- Aligned .bam and QC metrics files are saved in
  - ▶ star\_out/
- Quantification results are saved in
  - ▶ htseq\_out/

## Summary report of QC metrics

### run command lines

Run **pipeline\_scripts/rnaseq\_align\_and\_qc\_report.py** to create an HTML report of QC and alignment summary statistics for RNA-seq samples.

Read in **template\_files/rnaseq\_align\_and\_qc\_report.Rmd.template.txt** from specified directory `template_dir` to create a RMD script.

This script uses the many output files created in align and quantification step, converts these sample-specific files into matrices that include data for all samples, and then creates an Rmd document.

---

BASH

---

```
rnaseq_align_and_qc_report.py --project_name SRP033351 --samples_in /home/mengykan/Projects/SRP033351/files/sample_info_file.txt --a\
ligner star --ref_genome hg38 --library_type PE --path_start /home/mengykan/Projects/SRP033351 --template_dir /home/mengykan/Project\
s/shared_files/RNASeq
```

---

## submit LSF script

Generate a single LSF script SRP033351\_qc.lsf. This is a single-node analysis, but we recommend running it on HPC as the step of count normalization for PCA plots takes a lot of memory.

---

BASH

---

```
bsub < SRP033351_qc.lsf
```

---

```
cat SRP033351_qc.lsf
```

---

OUTPUT

---

```
#!/bin/bash
#BSUB -L /bin/bash
#BSUB -J SRP033351_qc
#BSUB -q normal
#BSUB -o SRP033351_qc_%J.out
#BSUB -e SRP033351_qc_%J.screen
#BSUB -M 36000
#BSUB -n 1
cd /home/mengykan/Projects/SRP033351/SRP033351_Alignment_QC_Report_star/; echo "library(rmarkdown); rmarkdown::render
('SRP033351_QC_RnaSeqReport.Rmd')" | R --no-save --no-restore
```

---

## output files

Output files are saved in ~/Projects/SRP033351/SRP033351\_Alignment\_QC\_Report\_star

The RMD and corresponding HTML report files are:

- SRP033351\_QC\_RnaSeqReport.Rmd
- SRP033351\_QC\_RnaSeqReport.html

## Gene-based DE analysis

### run command lines

Run **pipeline\_scripts/rnaseq\_de\_report.py** to perform DE analysis and create an HTML report of differential expression summary statistics.

Read in **template\_files/rnaseq\_de\_report.Rmd.template.txt** from specified directory **template\_dir** to create a RMD script.

---

BASH

---

```
rnaseq_de_report.py --project_name SRP033351 --samples_in /home/mengykan/Projects/SRP033351/files/sample_info_file.txt_withQC.txt --\
comp /home/mengykan/Projects/SRP033351/files/SRP033351_comp_file.txt --de_package deseq2 --ref_genome hg38 --path_start /home/mengyk\
an/Projects/SRP033351 --template_dir /home/mengykan/Projects/shared_files/RNASeq
```

---

## script options

The `--sample_in` option specifies user provided phenotype file for DE analysis. The columns are the same as **example\_files/sample\_info\_file.txt** but with an additional column *QC.Pass* designating samples to be included (*QC.Pass*=1) or excluded (*QC.Pass*=0) after QC. This column naming is rigid which will be recognized in pipeline scripts, but column order can be changed. In the current example, all samples pass QC.

The `--comp` option specifies comparisons of interest in a tab-delimited text file with one comparison per line with three columns (i.e. Condition1, Condition0, Design), designating Condition1 vs. Condition2. The DE analysis accommodates a *paired* or *unpaired* option specified in Design column. For paired design, specify the condition to correct for that should match the column name in the sample info file - e.g. *paired:Donor*. Note that if there are any samples without a pair in any given comparison, the script will automatically drop these samples from that comparison, which will be noted in the report.

Find the example comp file here **example\_files/SRP033351\_comp\_file.txt**.

Condition1	Condition0	Design
healthy_alb	healthy_untreated	paired:Donor
healthy_dex	healthy_untreated	paired:Donor
healthy_alb_dex	healthy_untreated	paired:Donor

## submit LSF script

Generate a single LSF script `SRP033351_deseq2.lsf`. This is a single-node analysis, but we recommend running it on HPC as the steps of count normalization for pairwise comparisons take a lot of memory.

---

BASH

---

```
bsub < SRP033351_deseq2.lsf
```

---

```
cat SRP033351_deseq2.lsf
```

---

OUTPUT

---

```
#!/bin/bash
#BSUB -L /bin/bash
#BSUB -J SRP033351_deseq2
#BSUB -q normal
#BSUB -o SRP033351_deseq2_%J.out
#BSUB -e SRP033351_deseq2_%J.screen
#BSUB -M 36000
#BSUB -n 1
cd /home/mengykan/Projects/SRP033351/SRP033351_deseq2_out/; echo "library(rmarkdown); rmarkdown::render('SRP033351_DE
Seq2_Report.Rmd')" | R --no-save --no-restore
```

---

## output files

Output DE results are saved in `~/Projects/SRP033351/SRP033351_deseq2_out`:

- Pairwise DE comparisons

- ▶ e.g. `SRP033351_healthy_dex_vs_healthy_untreated_full_DESeq2_results.txt`
- Normalized counts in samples for each pairwise comparison
  - ▶ e.g. `SRP033351_healthy_dex_vs_healthy_untreated_counts_normalized_by_DESeq2.txt`
- Normalized counts for all samples
  - ▶ `SRP033351_counts_normalized_by_DESeq2.txt`

The RMD and corresponding HTML report files are:

- `SRP033351_DESeq2_Report.Rmd`
- `SRP033351_DESeq2_Report.html`

## References

1. [RAVED pipeline]  
<https://github.com/HimesGroup/raved>
2. [refFlat format]  
<https://genome.ucsc.edu/FAQ/FAQformat.html>
3. [STAR tutorial]  
<https://github.com/alexdobin/STAR/blob/master/doc/STARmanual.pdf>
4. [illumina adapter sequences]  
[https://support.illumina.com/content/dam/illumina-support/documents/documentation/chemistry\\_documentation/experiment-design/illumina-adapter-sequences-1000000002694-07.pdf](https://support.illumina.com/content/dam/illumina-support/documents/documentation/chemistry_documentation/experiment-design/illumina-adapter-sequences-1000000002694-07.pdf)