RNA-seq alignment and quantification

1. Prepare the working directory.

1.1 Create a working directory "/workdir/$USER".

Copy all data files for this exercise from "/shared_data/RNAseq/exercise1/" into the working directory.

```
mkdir -p  /workdir/$USER/exercise1

cd /workdir/$USER/exercise1

cp /shared_data/RNAseq/exercise1/* ./

ls -l
```

1.2 Install Filezilla client on your laptop

Filezilla is a sftp client software. If you do not have any sftp client software on you laptop, download Filezilla client from this page. Double click to install.

https://filezilla-project.org/download.php?show_all=1

*** The installer might prompt you to install other additional software, e.g. virus protector, always click "no" to decline.**

1.3 Install IGV on your laptop

Go to the IGV web site (https://software.broadinstitute.org/software/igv/), click “Download”. Double click the IGV installation tool to install IGV. On Windows computer, the software is installed in the directory C:\Program Files.

1.4 Getting familiar with "screen"

If you do not know about Linux "screen" or "tmux" commands, now it is time to get familiar with it.

Most of the tools you will be using in this exercise take long time to finish. You will need to use the "screen" persistent sessions to run the software, so that you can safely detach from the session, and the job will keep running in the background on the server.

```
#Start screen. You would notice that "[screen 0 ...]" shows up at the header of
the terminal window.
screen

#Now you are in the "screen" persistent session, and you can run any software in
the session now.
ls -l

#To detach from the "screen", press "ctrl-a" "d" ("d" for "detach").
```

```
#After this, you will notice that "[screen 0 ...]" dispears from the header, you are back to regular session.
#The screen session is still alive in the background.

#To re-attach back to the screen session
screen -r

#You can create many independent sessions within one "screen" by pressing "ctrl-a" "c" ("c" for "create").
#After the new session is created, "[screen 1 ...]" shows up at the header of the terminal window.
#You can create as many independent sessions as you want, they will be named "screen 0","screen 1", "screen 2", et al.

#You can switch between these sessions by pressing "ctrl-a" "n" ("n" for next).

#To kill a screen session
#press "ctrl-d" when you are inside screen session.

#To detach from the screen
#press "ctrl-a" "d" to detach from "screen".

#Please be aware of the difference bwtween "detach" and "kill". With "detach", you can re-attach back to the session. With "kill", the session is permanently removed, together with any jobs running in the session.
```

2. Examine qualities of the RNA-seq data files

2.1 Run fastqc on the fastq file

```
fastqc ERR458493.fastq.gz
```

2.2 Examine the fastqc results.

The fastqc software creates a new file called “ERR458493_fastqc.html”. You can download this file to your laptop.

Open Filezilla, and enter the following information:

Host: cbsuxxxx.biohpc.cornell.edu   (Replace cbsuxxxxis with the name of your assigned computer)

UserName: your userID

Password: your password

Port: 22

Then click “Quickconnect”. If connected, you should see files on the remote server showing up in the right panel. On top of the right panel next to "Remote site:", type in “/workdir/” and press “Enter”, navigate to the “ERR458493_fastqc.html” file and drag it into the left panel.

After downloading the file, double clicking the file “ERR458493_fastqc.html” to open the page.

3. Map RNA-seq reads to the reference genome using STAR

We are going to use STAR to map the reads in the fastq files to the reference genome. STAR is a fast alignment software, but it requires a computer with large memory (30GB for the 3GB human genome).

3.1 Examine the input data files

List of data files in the directory

File	Description
R64.fa	Reference genome sequence file, in fasta format
R64.gtf	Genome annotation file, in gtf format.
ERR458493.fastq.gz	RNA-seq data file, wt_sample1
ERR458494.fastq.gz	RNA-seq data file, wt_sample2
ERR458495.fastq.gz	RNA-seq data file, wt_sample3
ERR458500.fastq.gz	RNA-seq data file, mu_sample1
ERR458501.fastq.gz	RNA-seq data file, mu_sample2
ERR458502.fastq.gz	RNA-seq data file, mu_sample3

Examine the content of input data files:

```
ls -la

zcat ERR458493.fastq.gz | head

zcat ERR458493.fastq.gz | wc -l

less R64.fa

less R64.gtf
```

- When using the “less” command, press “space” key to move on to the next page, or press “q” key to exit.
- The “wc -l” command counts the number of lines in a file. As every sequence read takes up 4 lines in the fastq file, the line number divided by 4 gives you the number of sequencing reads in the file.

3.2 Index the reference genome

The reference genome you are using for this exercise is very small. If you are working with a large genome, this step could take 30 min. We highly recommend you to run this step in a "screen" session. To start "screen", simply type the command: screen.

```
cd /workdir/$USER/exercise1

mkdir genome

export PATH=/programs/STAR:$PATH

STAR --runMode genomeGenerate --runThreadN 2 --genomeDir genome \
    --genomeFastaFiles R64.fa --sjdbGTFfile R64.gtf \
    --sjdbOverhang 50
```

Once the software starts to run, you can detach from "screen" by pressing "ctrl-a" followed by pressing "d".

- The “export PATH=/programs/STAR:$PATH” command puts STAR in your current PATH, so that you can run the software by simply typing the command “STAR”.
- The parameters:

--runMode genomeGenerate: Set runMode to “genomeGenerate” to index the genome;

--runThreadN: Number of CPU cores;

--genomeDir: Output directory for indexed genome database;

--genomeFastaFiles: Reference genome file

--sjdbGTFfile: Genome annotation file and it should be GTF format.

--sjdbOverhang: Use the value (reads_length -1). The RNA-seq reads used in this exercise are 51 bp long, so you put the number 50.

3.3. Map RNA-seq reads to the indexed genome.

Again, you should run it in "screen" session. To re-attach back to an existing "screen" session, use the command "screen -r".

```
STAR --quantMode GeneCounts --genomeDir genome --runThreadN 2 \
   --readFilesIn ERR458493.fastq.gz --readFilesCommand zcat \
   --outFileNamePrefix wt1_ --outFilterMultimapNmax 1 --outFilterMismatchNmax 2 
\
   --outSAMtype BAM SortedByCoordinate
```

After the mapping is finished, many new files are produced.

```
ls -lrt
```

The "ls -lrt" command lists the files based on the time they are modified, with the new ones at the bottom.

- The parameters:

--quantMode GeneCounts: Output a file with read counts per gene;

--genomeDir: Reference genome index directory;

--runThreadN: Number of CPU cores;

--readFilesIn: Sequence data file;

--readFilesCommand zcat: Input file is a decompressed .gz file;

--outFileNamePrefix: Prefix of the output file names;

--outFilterMismatchNmax 2 : Only report alignment with up to 2 mismatches per read;

--outSAMtype BAM SortedByCoordinate: Output sorted bam files.

- The files you need to keep are:

1) wt1_Aligned.sortedByCoord.out.bam: BAM file with the alignment results;

2) wt1_Log.final.out: A report file that shows percentage of read can be mapped;

3) wt1_ReadsPerGene.out.tab: a tab delimited text file with read count per gene.

<u>3.4 Inspect the mapping results</u>

```
less wt1_Log.final.out

less wt1_ReadsPerGene.out.tab
```

- The wt1_Log.final.out file gives you some basic statistics of the alignment.
- The wt1_ReadsPerGene.out.tab file gives you the read counts per gene. The four columns are,

· *column 1: gene ID*

· *column 2: counts for unstranded RNA-seq*

· *column 3: counts for reads aligned with plus strand of RNA*

· *column 4: counts for reads aligned with minus strand of RNA*

Use column 2 for un-stranded RNA-seq data. Use column 4 for stranded RNA-seq data. Use column 3 for 3’ RNA-seq.

For this exercise, you will need to use column 2, as the data is un-stranded.

4. Visualize the BAM file with IGV genome browser

<u>4.1 Index the bam files</u>

For IGV to read the BAM files, the “.bam” files need to be indexed. You will use samtools to index the bam file:

```
samtools index wt1_Aligned.sortedByCoord.out.bam
```

After this step, you will see a “.bai” file created for each “.bam” file.

<u>4.2 Download files to your laptop.</u>

Using Filezilla to download the “*.bam*” ,“.bai”, “R64.fa” and “R64.gtf” files.

4.3 Launch IGV on your laptop.

Double click “igv.bat” in the directory "C:\Program Files\IGV_2.8.11" to start IGV. It might take a few seconds before you see the software starting.

Most commonly used genomes are already loaded in IGV. In this exercise, you will create your own genome database.

Click “Genomes”->”Create .genome” file. Fill out the following fields:

Unique identifier: R64

Descript name: R64

Fasta: use the “Browse” button to find the R64.fa file

Gene file: use the “Browse” button to find the R64.gtf file

Then save the genome database on your computer.

From menu “File” -> “Load file”, open the “wt1_Aligned.sortedByCoord.out.bam”.

Inspect the following regions by enter the text in the box next to “Go” and click “Go”.

II:265,593-282,726

5. Process multiple RNA-seq data files in parallel

In a typical RNA-seq experiment, you have many samples. It could take hours or even days to finish the alignments. There are two things you can do to make computing faster.

- Create a batch command ("a shell script") and process all files in parallel;
- Use the “Shared Memory” feature of STAR. (You will not use this feature in this exercise, but I will explain it at the end of this file.)

5.1 Prepare a shell script

We have prepared a shell script for you. It is located in the data directory with the file name “runSTAR.sh”.

If you would like to make it by yourself, you can also use a text editor to make a text file with the following lines.

```
STAR --quantMode GeneCounts --genomeDir genome --runThreadN 2 --readFilesIn ERR458493.fastq.gz --readFilesCommand zcat --outFileNamePrefix wt1_ --outFilterMultimapNmax 1 --outFilterMismatchNmax   2 --outSAMtype   BAM SortedByCoordinate

STAR --quantMode GeneCounts --genomeDir genome --runThreadN 2 --readFilesIn ERR458494.fastq.gz --readFilesCommand zcat --outFileNamePrefix wt2_ --outFilterMultimapNmax 1 --outFilterMismatchNmax   2 --outSAMtype   BAM SortedByCoordinate
STAR --quantMode GeneCounts --genomeDir genome --runThreadN 2 --readFilesIn ERR458495.fastq.gz --readFilesCommand zcat --outFileNamePrefix wt3_ --outFilterMultimapNmax 1 --outFilterMismatchNmax  2 --outSAMtype  BAM SortedByCoordinate

STAR --quantMode GeneCounts --genomeDir genome --runThreadN 2 --readFilesIn ERR458500.fastq.gz --readFilesCommand zcat --outFileNamePrefix mu1_ --outFilterMultimapNmax  1 --outFilterMismatchNmax  2 --outSAMtype  BAM SortedByCoordinate

STAR --quantMode GeneCounts --genomeDir genome --runThreadN 2 --readFilesIn ERR458501.fastq.gz --readFilesCommand zcat --outFileNamePrefix mu2_ --outFilterMultimapNmax  1 --outFilterMismatchNmax  2 --outSAMtype  BAM SortedByCoordinate

STAR --quantMode GeneCounts --genomeDir genome --runThreadN 2 --readFilesIn ERR458502.fastq.gz --readFilesCommand zcat --outFileNamePrefix mu3_ --outFilterMultimapNmax  1 --outFilterMismatchNmax  2 --outSAMtype  BAM SortedByCoordinate
```

If you are not familiar with Linux text editors like "vi" or "nano", Mac users can use “BBEdit” (The free version is fine). (https://www.barebones.com/products/bbedit/), Windows users can use “Notepad++” (http://notepad-plus-plus.org/). If the file is made on a Windows computer, you need to make sure to save the file as a LINUX style text file. From NotePad++, used the "Edit -> EOL Conversion -> UNIX" option. After the file is saved you can use FileZilla to upload the file to your home directory.

5.2 Run the script in parallel

You will use the GNU parallel to run the scripts in parallel.

```
parallel -j 3  < runSTAR.sh >& log &
```

- The "-j 3" parameter is to specify that "parallel" will run three "jobs" at a time. You need to make sure that the job number times CPU core per job does not exceed total number of CPU cores of the computer.

Once the software starts, check the running jobs by "top" command:

```
top -u $USER
```

To exit "top", press "q".

6. Generate a read count matrix.

After running the shell script, you will get 6 files read count files, with one file per sample (*_ReadsPerGene.out.tab). Now you will need to combine the 6 files into one single file for statistical analysis. You can use Excel to do this, and then save the merged file as a tab-delimited text file. Or you can use the following commands:

```
paste wt1_ReadsPerGene.out.tab    wt2_ReadsPerGene.out.tab
wt3_ReadsPerGene.out.tab mu1_ReadsPerGene.out.tab    mu2_ReadsPerGene.out.tab
mu3_ReadsPerGene.out.tab | \
   cut -f1,2,6,10,14,18,22    | \
   tail -n +5 > gene_count.txt

less gene_count.txt
```

- paste: Merge the 5 files by columns;
- cut -f1,2,6,10,14,18,22: Extract columns 1,2,6,10,14,18,22 from the merged data (column 1 is the gene name, columns 2-22 are second column from each individual file);
- tail -n +5: Discard the first 4 lines of statistics summary and start from line 5;
- >gene_count.txt: Write the result into a file gene_count.txt
-

7. Load the matrix into R and make PCA Plot with DESeq2

In the exercise data directory, there is a file named “samples.txt”. It is a tab-delimited text file, you can inspect this file with “less samples.txt”. This files specify the sample names and experimental conditions for each sample used in this exercise.

The default R on BioHPC computers does not work for DESeq2 due to its parallel BLAS library. You will need to start R with “/programs/R-3.5.0s/bin/R”.

When you work on your own data, make sure to remove the part "+ xlim(-2.5, 2.5) + ylim(-1, 1)" in the last command. You can add these two parameters as needed to set the range of x and y axis range.

```
cd /workdir/$USER/exercise1

/programs/R-3.5.0s/bin/R

library(DESeq2)

library(ggplot2)

cts <- as.matrix(read.csv("gene_count.txt", sep="\t", row.names=1,
header=FALSE))

coldata <- read.csv("samples.txt", sep="\t", row.names=1)

colnames(cts) <- rownames(coldata)

dds <- DESeqDataSetFromMatrix(countData = cts, colData = coldata, design = ~
Genotype)

vsd <- vst(dds, blind=FALSE)

pcaData <- plotPCA(vsd, intgroup=c("Genotype"), returnData=TRUE)

percentVar <- round(100 * attr(pcaData, "percentVar"))

ggplot(pcaData, aes(PC1, PC2, color=Genotype)) + geom_point(size=3) + xlim(-2.5,
2.5) + ylim(-1, 1) +      xlab(paste0("PC1: ",percentVar[1],"% variance")) +
ylab(paste0("PC2: ",percentVar[2],"% variance")) +
geom_text(aes(label=name),vjust=2)
```

```
ggsave("myplot.png")
```

After this, you can download the "myplot.png" file using Filezilla. Double click to open the file.

Appendix. use the “shared memory” feature of STAR

The first step of running STAR is to load the genome database into memory. There are two issues here:

1. Each job would take several minutes to load the same genome database into memory;
2. Each job would use significant amount of memory to keep its own copy of the genome database;

STAR provides a feature, which allows you to pre-load the genome database into the shared memory space, which can be used by all STAR alignment jobs on this computer.

Here are the steps:

Step 1. Load genome into database and keep it there.

```
STAR --genomeLoad LoadAndExit --genomeDir genome
```

Step 2. Run STAR using the shared memory

Make a shell script with STAR alignment commands as you did in section 5. Add these two parameters into each STAR command: “--genomeLoad LoadAndKeep --limitBAMsortRAM 4000000000” . The "genomeLoad" option instructs STAR to use shared memory, and the "limitBAMsortRAM" option is to instruct STAR to limit 4GB for bam sorting step. You can decrease or increase the sorting memory based on the computer you are using.

Now you can run multiple jobs of STAR with “parallel” script, and each job will use the same shared memory.

Step 3. Remove the genome database from the shared memory.

```
STAR --genomeLoad Remove --genomeDir genome
```