# RnaSeq Data Pre-Processing

**Dr. Rudramani Pokhrel,**

**Last modified: 14 Apr 2022**

**Immunobiology Department, The university of Arizona, Tucson Arizona**

email: rpokhrel@email.arizona.edu

The aim of this tutorial is to generate gene vs sample count matrix from raw fastq Bulk RNAseq read files. It starts with preparing the virtual environment and step by step process to generate count data. At the end I have provided the necessary script files to process the method in batches.

## Download the miniconda package from conda repo for linux platform.

Type following commands in the terminal:

- Download: ***wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh***
- Install: ***bash Miniconda3-latest-Linux-x86_64.sh***
- Initialize the conda environment: ***conda init bash***

Note: You can download your version according to the platforms (linux, macOS, Windows) from this link: https://docs.conda.io/en/latest/miniconda.html. For Apple M1 please follow this link: https://naolin.medium.com/conda-on-m1-mac-with-miniforge-bbc4e3924f2b

## Create the virtual environment of named *rna-seq* and installing all the required packages.

Run following commands in the terminal:

- Create virtual environment named *rna_seq*: ***conda create -y -n rna_seq python=3***
- Activate the environment: ***conda activate rna_seq***
- Install required packages: ***conda install -y -c bioconda -c conda-forge -c anaconda fastp bwa htseq glob2 pandas samtools==1.11***

## Mappping read files to the reference genome and get transcriptomic counts

- Use Ensembl (https://uswest.ensembl.org/info/data/ftp/index.html) or from USCS genome browser (https://genome.ucsc.edu) or from NCBI ( https://www.ncbi.nlm.nih.gov) or from GENECODE (https://www.gencodegenes.org) to download reference genome and annotation files
- Make a folder in which you will put all of you genome data and fastq read files
  - ***mkdir rnaseq***
  - ***cd rnaseq***
- copy all the files into this directory using cp command or simply copy/paste
- Next process would be doing quality control of your fastq reads. I would prefer **fastp** package which can be used to inspect the read data and at the same time perform QC (adapter trimming and filtering) on read fastq file:
  - First create a folder for filtered reads
    - ***mkdir filtered_read***
  - Run fastp for single end read
    - ***fastp -i read1 -o filtered_read/read1 -h filtered_read/filtered.html***
  - For paired end reads:
    - ***fastp -i read1 -I read2 –detect_adapter_for_pe -o filtered_read/read1 -O filtered_read/read2 -h filtered_read/filtered.html***
      The filtered.html file gives the summary of read before and after filtering.

## Mapping to genome

I am using bwa aligner package to map the filtered read to the reference genome.

- First prepare the index file:
  - ***bwa index reference.fasta -p bwa_index***
    Note: -p is for prefix for all index files
- Now run following command for mapping to genome:
  - For single end read:
    - ***bwa mem -t 8 bwa_index filtered_read/read1 > mapped_read.sam***
  - For paired end read:
    - ***bwa mem -t 8 bwa_index filtered_read/read1 filtered_read/read2 > mapped_read.sam***

    Note: -t option is for number of cores/cpus you want to use. I recommend using minimum of 8GB of memory
- Use samtools to convert sam to bam file, sort the bam file and index it
  - ***samtools view -Sb mapped_read.sam > mapped_read.unsorted.bam***
  - ***samtools sort -o mapped_read.sorted.bam mapped_read.unsorted.bam***
  - ***samtools index mapped_read.sorted.bam***
- Use htseq-count to obtain gene/transcriptomic count matrix
  - ***htseq-count –format=bam –stranded=no –type=gene –order=pos –idattr=ID mapped_read.sorted.bam reference_annnotation.gff3 > count.txt***
    Note: you can change **–type** and **–idattr** options according to your study need. Look into the **Annotation.gff3** file for available type and ID.

## Employ above methods in multiple samples

Generally, we have multiple samples (eg. control vs disease) read files. Here I present a bash scripting to loop over all the samples for paired end reads.

# Create directory for filtered read and count matrix
```
mkdir -p filtered_read
mkdir -p htseq_count
```

# Looping over read1
```
for R1 in \*\_R1\_\*.gz do
```

# replacing *R1* with *R2* for reverse read. please look at your pattern
```
R2=(${R1//\_R1\_/\_R2\_})
```

# you can print with echo command to see whether it is working or not
```
#echo \$R1 \$R2
```

# create a variable B for base name for reading convenience which only captures the sample information
```
B=${R1::-16}
```

# Here B trims all the last 16 character from read file. You can choose your own number

```
fastp -i $R1 -I  $R2 --detect_adapter_for_pe -o filtered_read/$R1  -O filtered_read/$R2 -h filtered_read/$B.html
```

```
bwa mem -t 8 bwa_index filtered_read/$R1 filtered_read/$R2 > ${B}.sam
```

```
samtools view -Sb ${B}.sam > ${B}.unsorted.bam
samtools sort -o ${B}.bam ${B}.unsorted.bam
samtools index ${B}.bam
```

```
htseq-count --format=bam --stranded=no --type=gene --order=pos --idattr=ID ${B}.bam $gff3 > htseq_count/${B}.txt
```

***done***

## Alternatively

You can download the bash script from my github ripo and use it directly. Download script in current rnaseq folder from:

***wget https://raw.githubusercontent.com/githubrudramani/Bioinformatics/master/rna_seq_preprocess.sh***

Note: change the path/file name and location of reference in rna_seq_preprocess.sh script.
***bwaIndex=/path/to/bwa_index***
***gff3=/path/to /annotation.gff3***

Run command:
***chmod +x rna_seq_preprocess.sh***
***./rna_seq_preprocess.sh***

Now you have count matix for all the samples in ***htseq_count*** directory. You can use python or R to merge the count matrix and make sample vs transcriptomic data matrix.I have written a python script to merge the data. To download the script run following command in the inside htseq_count folder:

***cd htseq_count***
***wget https://raw.githubusercontent.com/githubrudramani/Bioinformatics/master/merge_htseq_counts.py***

Then run: ***python merge_htseq_counts.py***

The name of output file is merged_counts.csv in csv format.

## ———————Code Summary———————-

**Following lines of code is all needed for pre-procession bulk RNAseq raw fastq read files. Make sure you have changed the respective file names and paths in the scripts.**

***wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh bash Miniconda3-latest-Linux-x86_64.sh***

***conda init bash***

***conda create -y -n rna_seq python=3***

***conda activate rna_seq***

***conda install -y -c bioconda -c conda-forge -c anaconda fastp bwa htseq glob2 pandas samtools==1.11***

***bwa index reference.fasta -p bwa_index***

***wget https://raw.githubusercontent.com/githubrudramani/Bioinformatics/master/rna_seq_preprocess.sh***

***chmod +x rna_seq_preprocess.sh***

***./rna_seq_preprocess.sh***

***cd htseq_count***

***wget https://raw.githubusercontent.com/githubrudramani/Bioinformatics/master/merge_htseq_counts.py***

***python merge_htseq_counts.py***

## Output:

***merged_counts.csv***
After that you can use differential analysis statistical packages like **DEseq2**, **edgeR**, **Limma** from Bioconductor.