
THE UNIVERSITY *of* EDINBURGH
School of Biological Sciences

# BPSM Assignment 1
# Bioinformatics Programming and System Management

Student Exam Number: B200735

# BPSM Assignment 1

**B200735**

## ASSIGNMENT LINK

https://github.com/B200735-2021/CW1/
The password is: pass2021

## DESCRIPTION

The flow chart of this study is shown in Fig. 1.

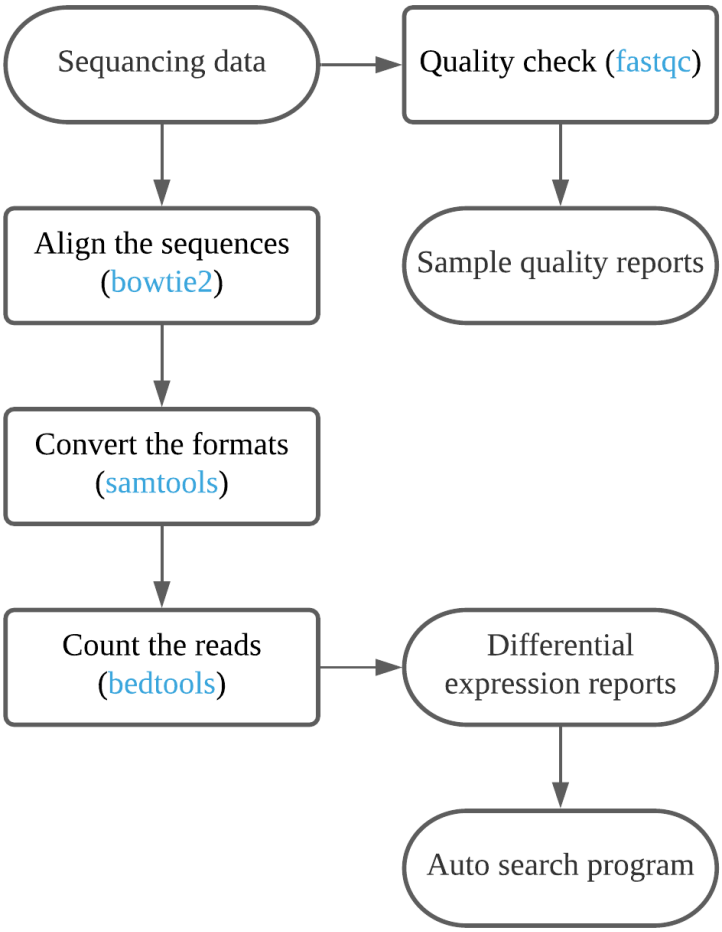


**Figure 1.** The flow chart explains the main procedures.

### Fastqc

After copying all the data we needed to the working directory, fastqc is used to generate a quality control report, in order to check the length concentration of all the samples.

```
cp /localdisk/data/BPSM/AY21/fastq/100k.fqfiles names.txt
sed -i '1d' names.txt
mkdir quality_control
cut -f 6,7 names.txt | while read file1 file2; do fastqc -t 12 -o quality_control --extract
    $file1 $file2 ;done
```

At first, we make a list containing all the information of samples, and remove the headline in the file. The "-t 12" means using 12 threads to complete this job, and "-o quality_control" means output all the result files into a directory called "quality_control", the parameters "- -extract" means output unzipped files which is more convenient for us to generate a quality overview later. By using the list of samples, no matter how much new data we are going to have, we can always finish the process.

### Bowtie2 and Samtools

Before aligning the sequences by bowtie2 with the genome given, we need to build an index for all the samples and their location within this genome. Here we use "bowtie2-build" to create the index with the genome given in fasta format, and we call this index "index".

```
bowtie2-build TriTrypDB-46_TcongolenseIL3000_2019_Genome.fasta index
cut -f 1,6,7 names.txt | while read name file1 file2;do bowtie2 -p 10 -x index -1 $file1 -2
    $file2 | samtools view -u | samtools sort -@ 10 > $name.bam; done
```

By using the previously created list, we use bowtie2 to match the genome fragments with our sample fragments, which outputs the locations of each read in the genome. The parameter "-p 10" indicates 10 threads are used for processing at the same time, and "-x index" means that the built index is used as the index. Since our samples are paired-end, we input both end data using the parameter "-1 $file1 -2 $file2". Using the environment variables help us make sure that the pipeline can still work perfectly even when we facing more different data.
After creating the sam files by aligning the data to the genome, we convert the sam files into the smaller bam files. Then, we sort the binary bam files in order to save more memory, and build indexes by samtools. This step is performed by "samtools view -u" to view the uncompressed bam output, then the bam files are "sorted" with 10 threads by the parameter "-@ 10".

### Bedtools

Having built the indexes by bam files, we count the number of reads in each sample by bedtools. After aligning with the given genome, we have the files containing the location of the reads, and bedtools will check if the reads are within a gene.
"Bedtools multicov" can count the overlapping areas between reads in samples and the genes in the given bed file. Putting the bam filename after "-bam", and the name of provided bed file after "-bed".

```
cut -f 1 names.txt | while read name; do samtools index $name.bam;
bedtools multicov -bams $name.bam -bed TriTrypDB-46_TcongolenseIL3000_2019.bed > $name.
    reads_count; done
```

### Fold change

First we add 1 to every mean of reads to avoid 0 being the denominator, then the fold change is compared by doing $|log_2(mean\ of\ reads(A)+1) - log_2(mean\ of\ reads(B)+1)|$. We set the threshold of absolute value of $log_2$ fold change to 1 in differential expression analysis, and then output the number of genes with absolute value of $log_2$ fold change $> 1$, and consider these to be extremely differentially expressed gene.

### "Group-wise" comparisons

The program compares the mean expression levels between different treatments(Induced or Uninduced), different treating time and different groups. Also, we sort the gene expression fold change in descending order.

## INSTRUCTIONS

1. Clone the repository
   At first, the user needs to clone the repository from GitHub, and then unzip the zip file with the password: pass2021. Then you will see there are three files, called "runme.sh", "ask" and "all_the_things_I_did.txt".

2. Run the script "runme.sh"
   The user needs to type the command below to run the main script.

```
chmod 700 runme.txt
./runme.txt
```

3. Wait for the program to finish processing data.
4. Automatically enter the searching menu.
   After the program finished analyzing data, the user will automatically enter an interactive searching menu. The user can easily choose the reports that he is interested in and go through the report file. After finishing reading the report, the user can press "q" to exit watching report, and then can choose other reports he is interested in.
   If the user exit the search program, he can simply use the command "**sh ask**" to reopen the search menu.
5. Go through all the reports manually
   All the reports are in the folder called "B200735" and start with "Report".

## DIFFICULTIES

At first, I am not familiar with the command lines and coding, which requires lots of logical thinking, especially when writing loops. Sometimes I struggle with the syntax in shell, where I try to solve the problems by googling for solutions from others who have encountered similar problems.
Also, I do not have much Bioinformatics background, so I read some relevant literature from PubMed to understand the meaning of the whole experiment. I have learned a lot from what is RNAi including siRNA and microRNA [2] to how does the Ago protein [1] collaborate in the interfering process. Also, I have learned the function of adaptors and how they can lead to inaccuracy, e.g., sometimes the adaptor can combine with each other without a read between both adaptors. I tried to understand the details in RNA-seq including why we need to break our data into fragments and how can this step help us align the reads with the genome.

## ALTERNATIVE FEATURES

### Generated quality reports

A summary of all the samples is generated, outputting the number of "WARN" and "FAIL", where samples without "WARN" or "FAIL" are skipped.

### Counted the number of extremely differently expressed genes

A report of the number of genes between different groups which fold change is larger than 1 is generated.

### Compared fold change between different groups and conditions

The fold change between different treatments is compared in order to analyze whether the RNAi can interfere the gene expression, where the specific genes that have been interfered are found out.
The fold change between different treating times is compared to discover whether the time is a vital variance of interfering gene expression.
The fold change between groups is compared to get the feedback of different responses by RNAi from different types of groups.

### Program for auto searching reports

There is a program that can help the user to search the reports he wants to see by simply selecting the way he wants to compare the fold change.

### Great applicability

The whole program has great general applicability, and can still work perfectly even there are more data that need to be processed. For example, data with more kinds of treatment or different treating times and more types of samples can be perfectly processed.

## REFERENCES

[1] Huafang Shi et al. "RNA Interference in Trypanosoma brucei". In: *Journal of Biological Chemistry* 284.52 (2009), pp. 36511–36520.

[2] Kai Zhang et al. “A novel class of microRNA-recognition elements that function only within open reading frames”. In: *Nature structural & molecular biology* 25.11 (2018), pp. 1019–1027.