Introduction to RNA-Seq & Transcriptome Analysis

Jessica Holmes

PowerPoint by Shayan Tabe Bordbar
Edited by Gio Madrigal & Roberto Cucalón Tamayo

Introduction

In this lab, we will do the following:

1. On the IGB Biocluster:
   a) Use **STAR** to align RNA-Seq reads to mouse genome.
   b) Use **featureCounts** to count the reads.
   c) Use **multiqc** to assess the quality of alignment.
   d) Use **edgeR** to find differentially expressed genes.
2. On the Virtual Machine:
   a) View and inspect the results of differential expression analysis.
   b) Visualize our results on the desktop using the Integrative Genomics Viewer **(IGV)** tool.

Start the VM

- Follow instructions for starting VM (This is the Remote Desktop software).

- The instructions are different for UIUC and Mayo participants.

- Find the instructions for this on the course website under Lab Set-up:

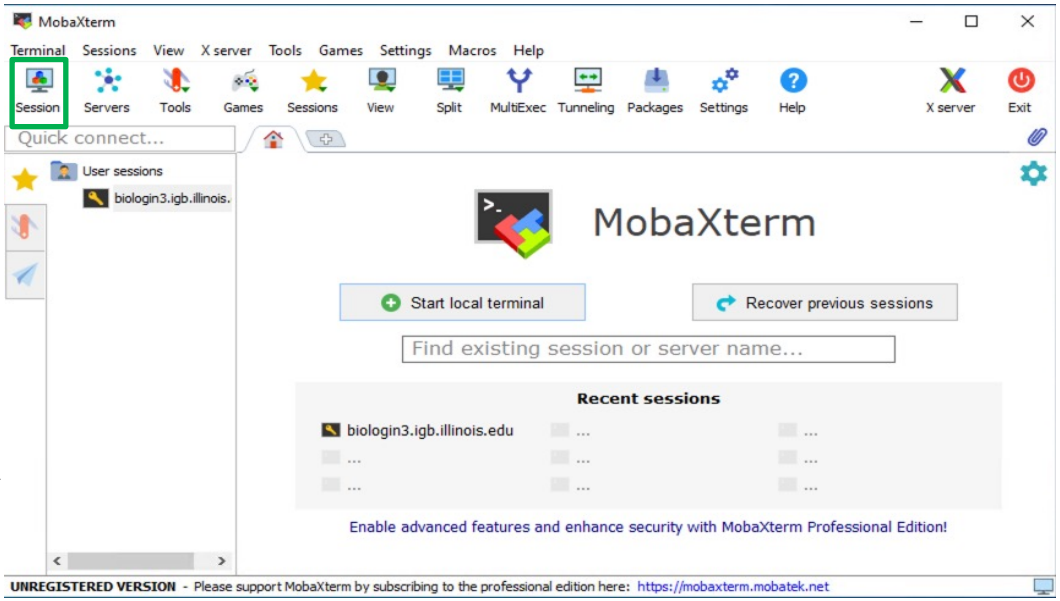
https://publish.illinois.edu/compgenomicscourse/2023-schedule/

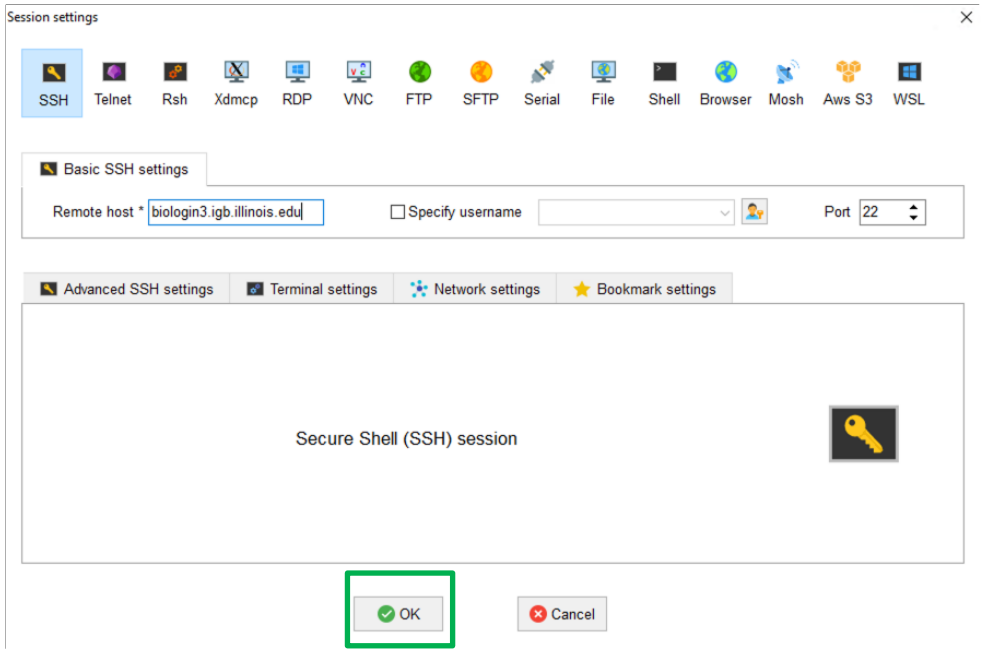
Step 0A: Accessing the IGB Biocluster for First Time

- Open **MobaXterm** from the VM
- In a new session, select **SSH** and type the following host name:

  `biologin3.igb.illinois.edu`

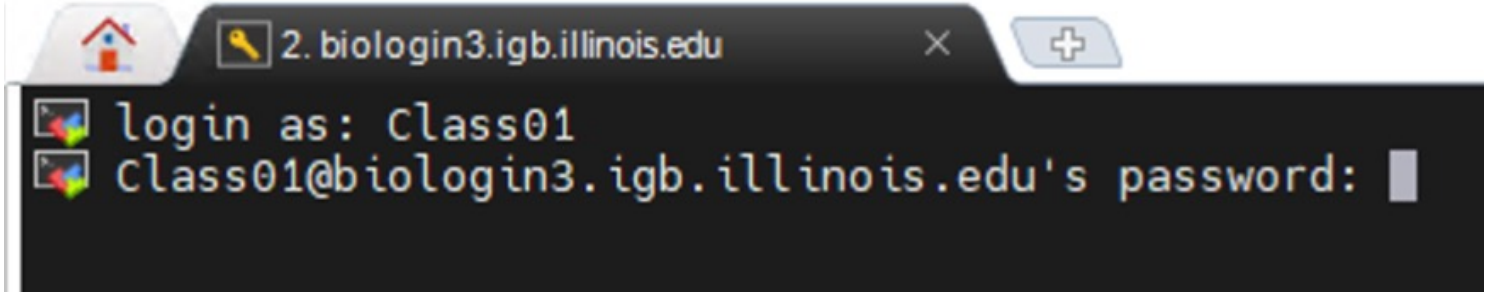
- Click **OK**





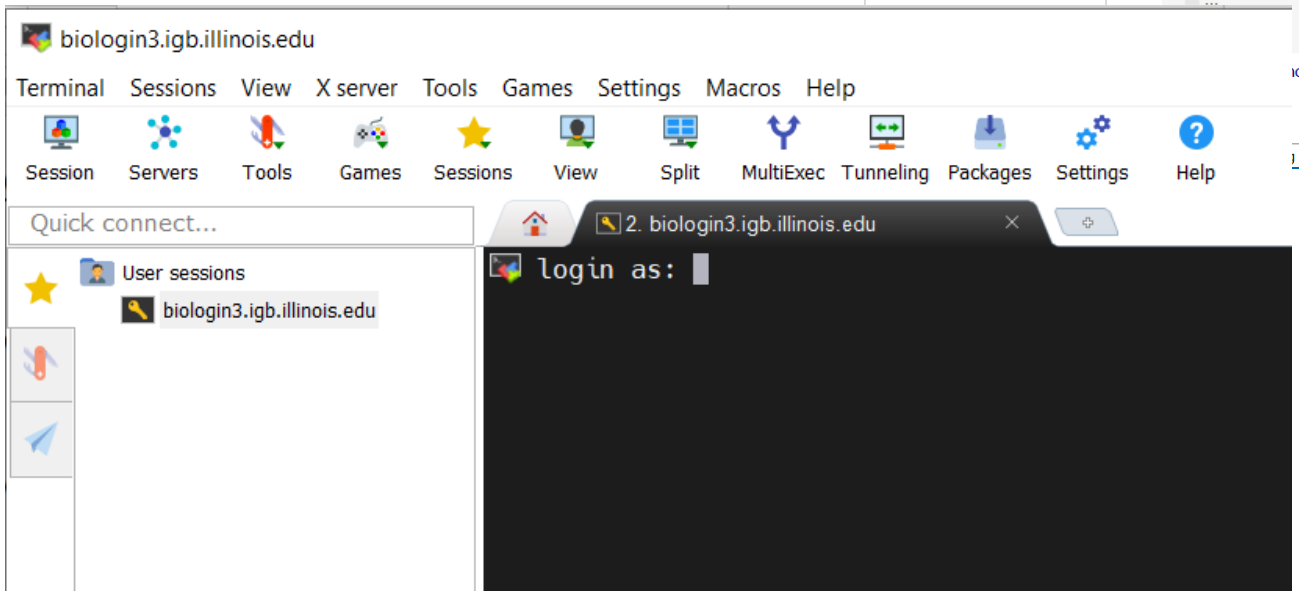
Step 0A: Accessing the IGB Biocluster

- Enter login credentials assigned to you.
- Example username: **Class01**
- You will not see any characters on screen when typing in password. Just type it.



Step 0A: Accessing the IGB Biocluster

If you have done this before, just double-click on the session you created once and type username and password.



Step 0B: Lab Setup

In MobaXterm

The lab is located in the following directory:

`/home/classroom/mayo/2020/mouse-rnaseq-2020/`

Following commands will copy a shell script, designed to prepare the working directory, to your home directory. Follow these steps to first copy, and then submit the script as a job to biocluster:

```
$ cd ~/

# Note ~ is a symbol in Unix paths referring to your home directory

$ cp /home/classroom/hpcbio/mayo-rnaseq/mouse-rnaseq-2020/src/Mayo-RNASeq/prep-directory.sh ./

# Copies prep-directory.sh script to your working directory.

$ sbatch prep-directory.sh

# submits a job to biocluster to populate your home directory with necessary files

$ squeue -u $USER # to check the status of the submitted job
```

Note: In this lab, we will **NOT** login to a node on the biocluster. Instead, we will submit jobs to the biocluster.

Step 0C: Working directory: data

Navigate to the created directory for this exercise and look what data folder contains.

```
$ cd mouse-rnaseq-2020
$ ls
# output should be:
# data results src
$ ls data/
# genome  rawseq
$ ls data/rawseq
$ ls data/genome
```

`data/rawseq`

File name	Time points	Replicate #	# Reads
a_0.fastq b_0.fastq	TP0	1 2	~ 1 million
a_8.fastq b_8.fastq	TP8	1 2	~ 1.1 million

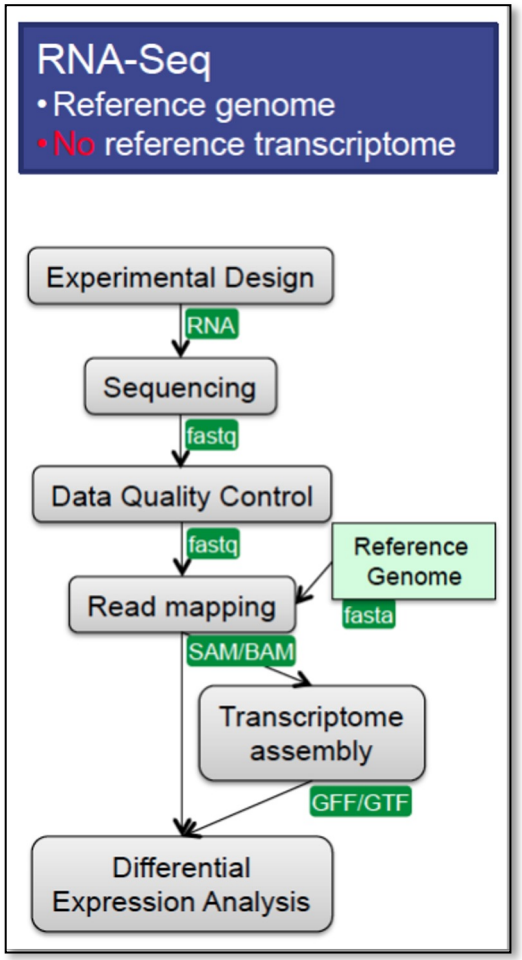
`data/genome`

Name	Description
mouse_chr12.fna	Fasta file with the sequence of chromosome 12 from the mouse genome
mouse_chr12.gtf	GTF file with gene annotation, known genes

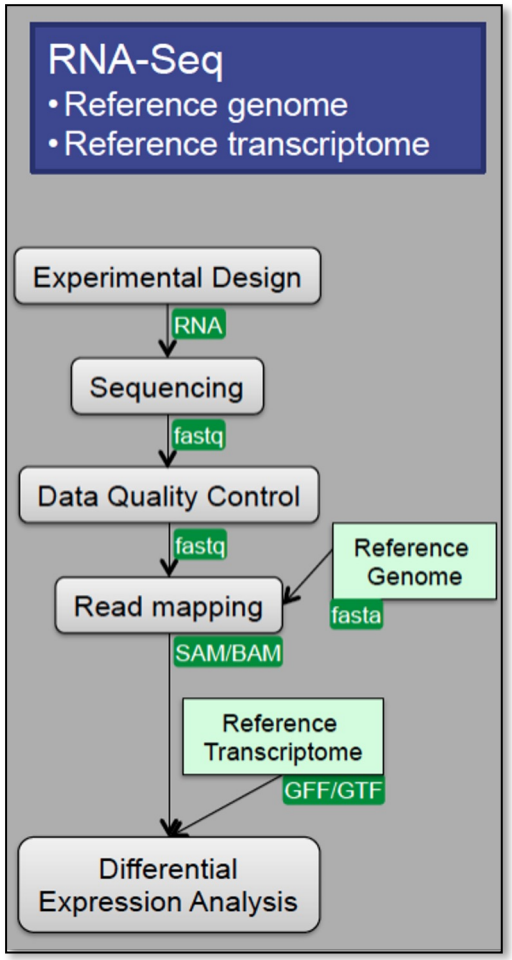
Step 0D: Working directory: scripts

Navigate to the directory containing the scripts and look what's inside.

```
$ cd ~/mouse-rnaseq-2020/src

$ ls *.sh *.R

# lists the scripts to be used in this lab:

# edgeR.sh         multiqc_summary.sh STAR-index-mouse-genome.sh  featureCounts.sh

# prep-directory.sh         stats_edgeR.R      makeTargetsFinal.R STAR-alignment.sh
```



v



Pipeline Overview

Step 1: Alignment using STAR

In this exercise, we will be aligning RNA-Seq reads to a reference genome.

Step 1A: Create a STAR index of the mouse genome (chromosome 12 only)

In this step, we will start a genome index generation job using the **sbatch** command. Additionally, we will gather statistics about our job using the **squeue** command.

Run the following command (colored black):

```
$ sbatch STAR-index-mouse-genome.sh

# This will execute STAR-index-mouse-genome.sh on the biocluster.

# OUTPUT in ~/mouse-rnaseq-2020/data/genome/

# STAR-2.7.3a_mouse-chr12_Index/


$ squeue –u $USER

# Get statistics on your submitted job

# This job takes 3-5 mins to complete.
```

Please do not try to Run the commands in this slide. This is just to explain what the script that we just ran (STAR-index-mouse-genome.sh) is supposed to do in more detail.

What's inside the `STAR-index-mouse-genome.sh` script?

```
#!/bin/bash
#SBATCH -N 1
#SBATCH -n 2
#SBATCH --mem 32G
#SBATCH -J make.index
#SBATCH -p classroom

# load the tool environment
module load STAR/2.7.3a-IGB-gcc-8.2.0

cd ~/mouse-rnaseq-2020/
mkdir -p data/genome/STAR-2.7.3a_mouse-chr12_Index/


STAR --runThreadN $SLURM_NTASKS \
     --runMode genomeGenerate \
     --genomeDir data/genome/STAR-2.7.3a_mouse-chr12_Index \
     --genomeFastaFiles data/genome/mouse_chr12.fna \
     --limitGenomeGenerateRAM 32000000000 \
     --genomeSAindexNbases 12 \
     --outTmpDir /scratch/$SLURM_JOB_ID
```

Tells the cluster 'job manager' what resources you want (1 Core, 32GB memory, run on the 'classroom' nodes, and name the job 'make.index'

Load the software. We are using a tool called 'STAR' to create an index for chr12 of mm9 mouse genome.

Change and make directory to store the index.

Run STAR tool in 'genomeGenerate' mode.

Step 1B: Align sequences using the created index

In this step, we will align sequences from fastq files to the mouse genome using STAR.

Run the following command (colored black):

```
$ sbatch STAR-alignment.sh

# This will execute STAR-alignment.sh on the biocluster.

# OUTPUT in ~/mouse-rnaseq-2020/results/star/

$ squeue –u $USER # to check the status of the submitted job

# This job takes 2-4 mins.

$ more STAR-alignment.sh

# Take a look at the script

# press “space” to go to the next page when using more
```

Please do not try to Run the commands in this slide. This is just to explain what the script that we just ran (STAR-alignment.sh) is supposed to do in more detail.

What's inside the `STAR-alignment.sh` script?

```
#!/bin/bash
#SBATCH -N 1
#SBATCH -n 2
#SBATCH --mem 16G
#SBATCH --job-name=align_star
#SBATCH -p classroom
#SBATCH --array=1-4%2

# load the tool environment
module load STAR/2.7.3a-IGB-gcc-8.2.0

cd ~/mouse-rnaseq-2020/
mkdir -p results/star

STAR --runThreadN $SLURM_NTASKS \
     --genomeDir data/genome/STAR-2.7.3a_mouse-chr12_Index \
     --readFilesIn data/rawseq/${line}.fastq  \
     --sjdbGTFfile data/genome/mouse_chr12.gtf \
     --outFileNamePrefix results/star/${line}_ \
     --limitGenomeGenerateRAM 32000000000 \
     --outSAMtype BAM SortedByCoordinate \
     --outTmpDir /scratch/${SLURM_JOB_ID}_${SLURM_ARRAY_TASK_ID}

module load SAMtools/1.10-IGB-gcc-8.2.0

samtools index results/star/${line}_Aligned.sortedByCoord.out.bam
```

Tells the cluster 'job manager' what resources you want (1 Core, 16GB memory, run on the 'classroom' nodes, and name the job 'align_star'. Runs two samples at a time.

Load the software. We are using a tool called 'STAR' to align fastq reads to mouse chr12 genome.

Change and make directory to store the alignment results.

Run STAR tool in 'alignReads' (default) mode. Options are described in the next slide.

Load SAMtools software to generate index bam files for visualization with IGV

Run 'samtools index' for all created alignment files.

Step 1B: Align sequences using the created index

Please do not try to Run the commands in this slide. This is just to explain what are the arguments for running STAR.

Here we go over the essential arguments to use with STAR for aligning sequences in fastq files.

```
STAR    --runThreadN $SLURM_NTASKS \# number of threads
        --genomeDir data/genome/STAR-2.7.3a_mouse-chr12_Index \
         # path to the indexed genome folder
        --readFilesIn data/rawseq/${line}.fastq \
         # path to the input fastq file
        --sjdbGTFfile data/genome/mouse_chr12.gtf \ # path to the gtf file
        --outFileNamePrefix results/star/${line}_ \
         # prefix to be used in the names of outputs
        --outSAMtype BAM SortedByCoordinate # TYPE OF OUTPUT
```

Step 1C: Output of STAR alignment Job

You should have 4 outputs per input fastq file when the job is completed.

Discussion:

- What did we just do?

Using STAR, we created an index for chr12 of mouse genome and aligned input fastq files.

Files

*.Aligned.sortedByCoord.out.bam
*_Log.final.out
*_Log.out
*_Log.progress.out
*_SJ.out.tab
*_STARgenome/

Where are these files located? Type the following command to see them:
ls ~/mouse-rnaseq-2020/results/star

Step 2: Read aligned counts

Use **featureCounts** to generate the aligned counts for each of the bam files generated in step 1.

Step 2A: Counting reads

featureCounts is part of the **Subread** module.

It takes alignment files (BAM, SAM), along with an annotation file (GTF file here) and counts the number of reads in the alignment that are associated to specified features in the annotation file.

```
$ sbatch featureCounts.sh

# OUTPUT in ~/mouse-rnaseq-2020/results/featureCounts/

$ squeue –u $USER # to check the status of the submitted job

# This job takes 1-4 mins.

$ more featureCounts.sh

# Take a look at the script
```

Please do not try to Run the commands in this slide. This is just to explain what the script that we just ran (featureCounts.sh) is supposed to do in more detail.

What's inside the `featureCounts.sh` script?

```
#!/bin/bash
#SBATCH -N 1
#SBATCH -n 1
#SBATCH --mem 8G
#SBATCH --job-name=counts
#SBATCH --array=1-4
#SBATCH -p classroom

# load the tool environment
module load Subread/2.0.0-IGB-gcc-8.2.0

cd ~/mouse-rnaseq-2020/
mkdir -p results/featureCounts

featureCounts -T 1 -s 2 -g gene_id -t exon \
-o results/featureCounts/${line}_featCounts.txt \
-a data/genome/mouse_chr12.gtf \
results/star/${line}_Aligned.sortedByCoord.out.bam
```

Tells the cluster 'job manager' what resources you want (1 Core, 8GB memory, run on the 'classroom' nodes, and name the job 'counts'. Runs 4 samples at a time.

Load the software. We are using 'featureCounts' tool from 'Subread' toolkit to count the reads assigned to genomic regions.

Change and make directory to store the count results.

Run featureCounts tool. Options are described in the next slide.

Step 2A: Counting reads

Please do not try to run the commands in this slide. This is just to explain the arguments for the featureCounts.

Here we go over the essential arguments for the featureCounts.

```
FeatureCounts -T 1 \ # number of threads
              -s 2 \ # use reverse strand (use -s 1 for forward strand)
              -t exon \ # describes the "feature" that this #software
                        # will look for in our GTF file
              -g gene_id_id \ # describes the "meta-feature" that should
                              # also be present in our GTF.
              -o results/ts/featureCounts/${line}_featCounts.txt \
              -a data/genome/mouse_chr12.gtf \# path to the gtf file
               results/star/${line}_Aligned.sortedByCoord.out.bam # path to
                     # the alignment file
```

Step 2B: Output of featureCounts

You should have **2** outputs per input fastq file when the job is completed.

Files

1. *.txt
2. *.txt.summary

Where are these files located? **Type the following command** to see them**:**

```
$ ls ~/mouse-rnaseq-2020/results/featureCounts
```

Run the following command to take a look at one of the output files:

```
$ more ~/mouse-rnaseq-2020/results/featureCounts/a_0_featCounts.txt.summary

# take a look at one of the summary output files
```

Step 3: Using MultiQC to generate quality report

Now we will use MultiQC to assess the quality of alignments and to collate STAR and featureCounts numbers.

We will also use a R script to generate plots on read mappings.

Step 3A: MultiQC

We will use multiqc tool to summarize the results from STAR and featureCounts.

```
$ sbatch multiqc_summary.sh
$ squeue –u $USER # to check the status of the submitted job
# This job takes ~ 1 minute.
# OUTPUT in ~/mouse-rnaseq-2020/results/
# multiqc_report.html      ReadFatePlot.jpeg         Targets_Final.txt
# we will analyze the results on VM
```

Note that the files generated by multiqc_summary.sh script have already been copied to **[course_directory]\04_Transcriptomics** on the VM for visualization.

Please do not try to Run the commands in this slide. This is just to explain what the script that we just ran (the multiqc_summary.sh) is supposed to do in more detail.

What's inside the `multiqc_summary.sh` script?

```
#!/bin/bash
#SBATCH -N 1
#SBATCH -n 1
#SBATCH --mem 8G
#SBATCH --job-name=multiqc
#SBATCH -p classroom

# load the tool environment
module load MultiQC/1.7-IGB-gcc-8.2.0-Python-3.7.2

cd ~/mouse-rnaseq-2020/results

multiqc ./

# load the tool environment
module purge
module load R/3.6.0-IGB-gcc-8.2.0

Rscript ../src/makeTargetsFinal.R
```

Tells the cluster 'job manager' what resources you want (1 Core, 8GB memory), run on the 'classroom' nodes, and name the job 'multiqc'.

Load the MultiQC module.

Change directory to results.

Run multiqc on the results directory.

Unload all modules. Then load R module.

Run makeTargetsFinal.R script. This R script creates visualizations on the fate of sequencing reads. can use Linux 'more' command to view the script.

Step 0: Local Files

For viewing and manipulating the files needed for this laboratory exercise, the path on the VM will be denoted as the following:

[course_directory]

We will use the files found in:

[course_directory]\04_Transcriptomics

[course_directory]= Desktop\VM

On Desktop

Step 3A: MultiQC

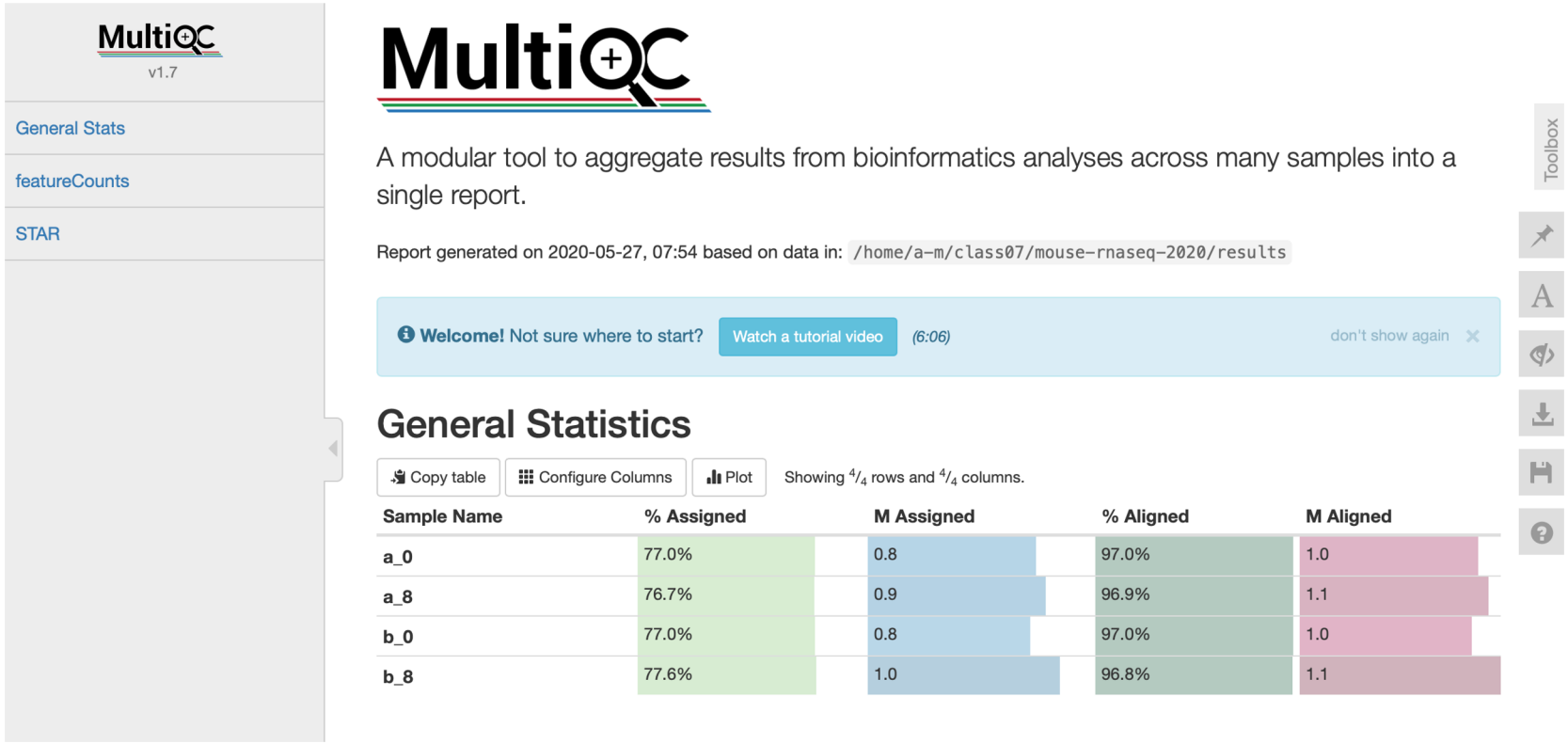
- Navigate to the following directory on your VM:

  `[course_directory]\04_Transcriptomics\`

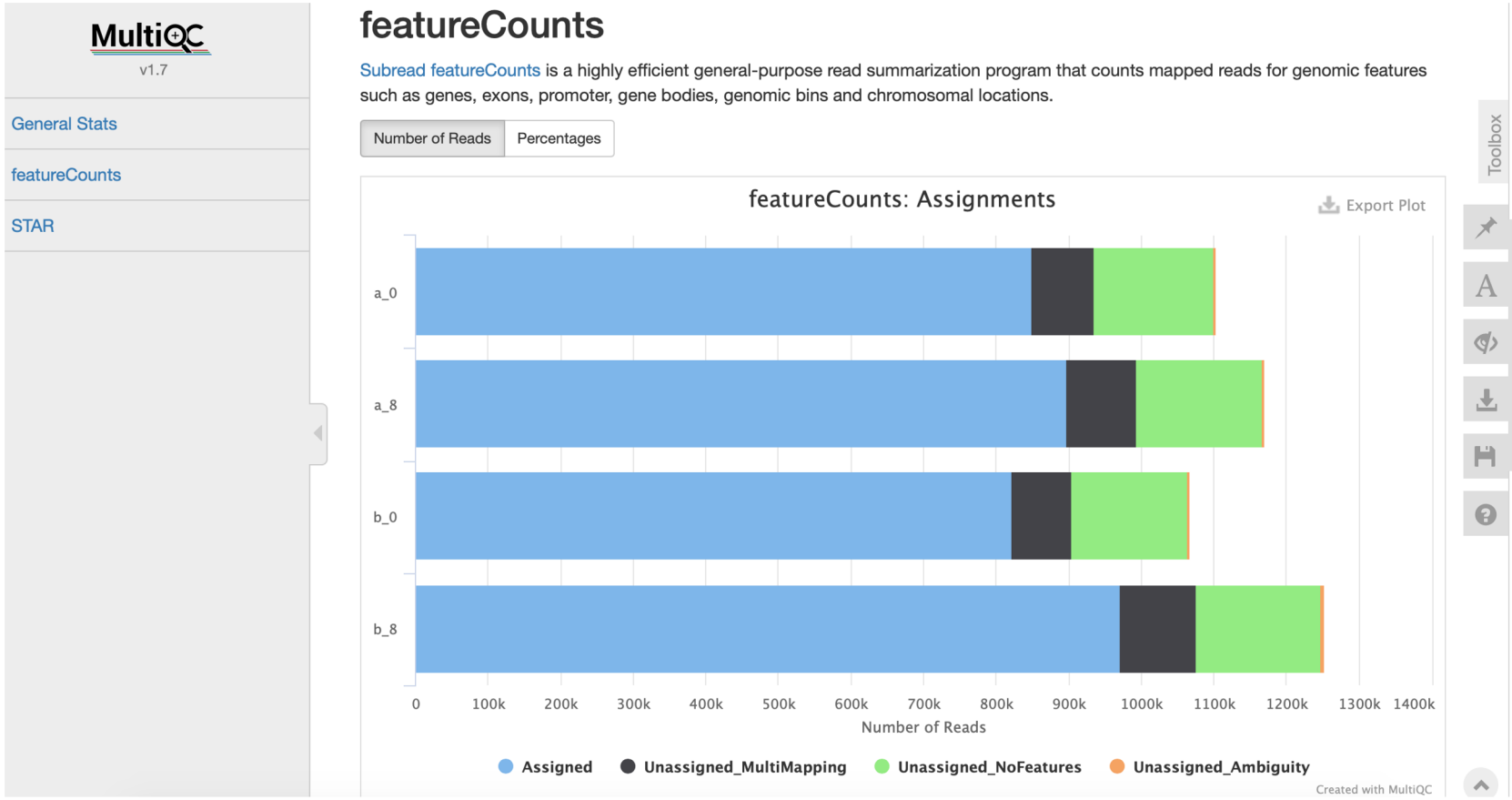
- Note that the files generated by multiqc_summary.sh script have already been copied to this directory for convenience.

- Open **multiqc_report.html**

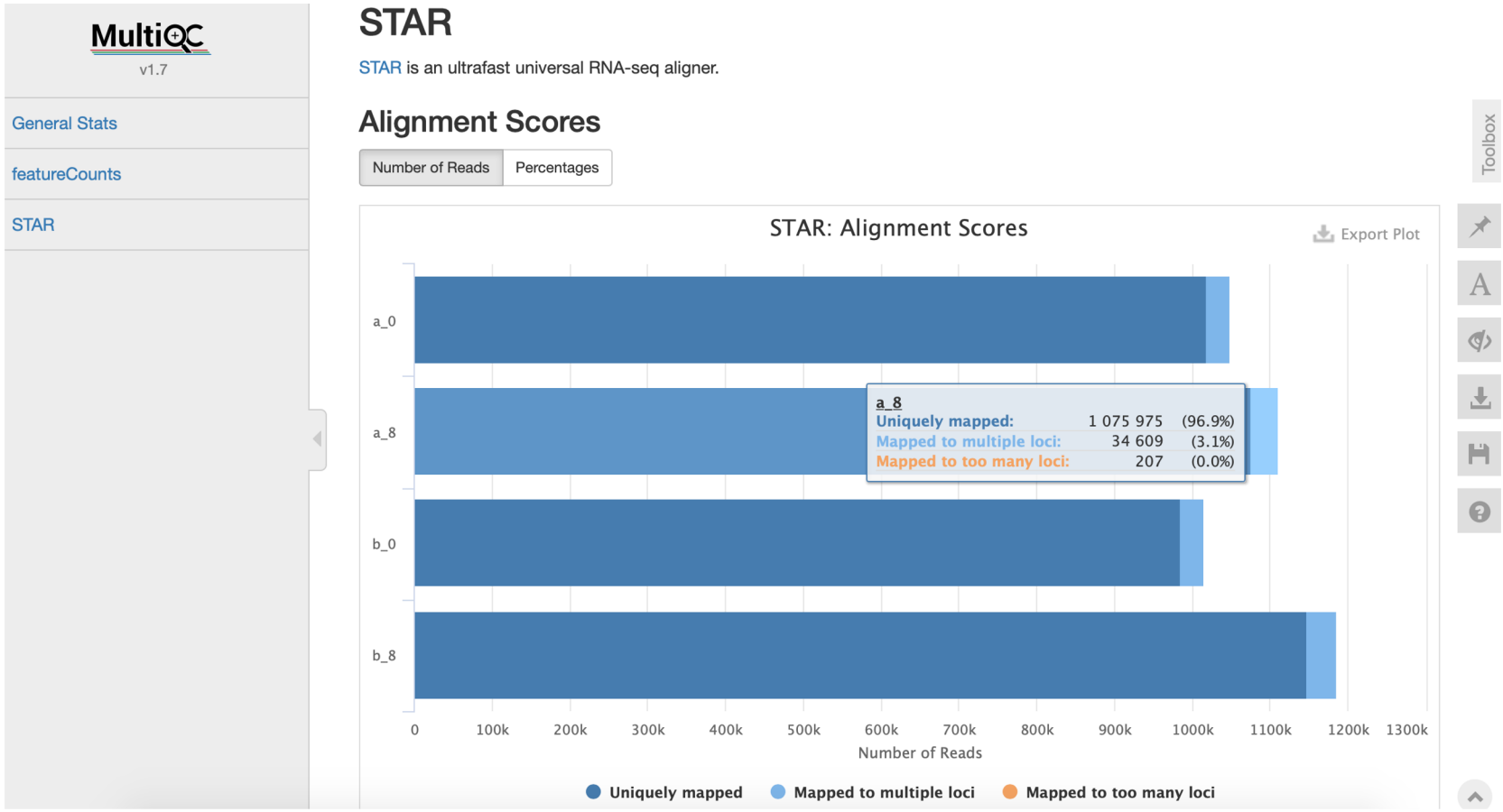
Step 3A: MultiQC



Step 3A: MultiQC

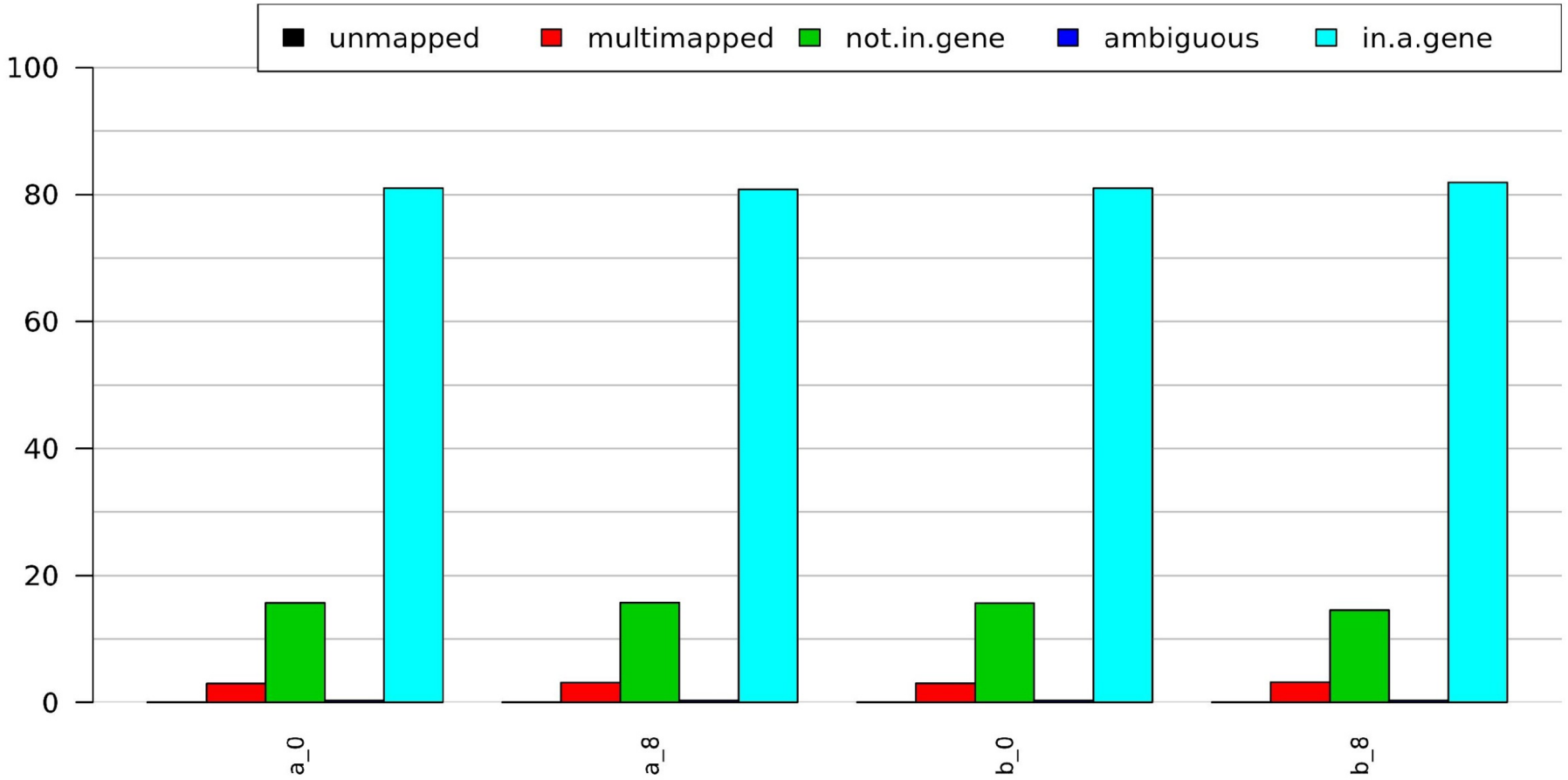


Step 3A: MultiQC



- Open ReadFatePlot.jpeg

This file is in the same directory as the previous one: [course_directory]\04_Transcriptomics



Step 4: Finding differentially expressed genes

Now we will use edgeR to analyze the count files generated in step 2 to find differentially expressed genes between two time points.

Step 4: Statistical analysis with edgeR

In MobaXterm

We run edgeR.sh, that uses an R script “stats_edgeR.R” to perform the statistical analysis and find differentially expressed genes.

We use FDR 0.05 to call differential expression.

```
$ sbatch edgeR.sh

$ squeue –u $USER # to check the status of the submitted job

# This job takes ~ 30 seconds.

# OUTPUT in ~/mouse-rnaseq2020/results/edgeR/

# MDSclustering.jpeg     NumSigGenes_FDR0.05.csv  RawCounts.txt

# t8_vs_t0_AllResults.txt        t8_vs_t0_MeanDifferencePlot.jpeg
```

Note that the files generated by **edgeR.sh** script have already been copied to **[course_directory]\04_Transcriptomics** on the VM for convenience.

Please do not try to Run the commands form the first grey box. This is just to explain what the script that we just ran (the edgeR.sh) is supposed to do in more detail.

What's inside the `edgeR.sh` script?

```
#!/bin/bash
#SBATCH -N 1
#SBATCH -n 1
#SBATCH --mem 8G
#SBATCH --job-name=edgeR
#SBATCH -p classroom

# load the tool environment
module load R/3.6.0-IGB-gcc-8.2.0

cd ~/mouse-rnaseq-2020/
mkdir -p results/edgeR

Rscript src/stats_edgeR.R
```

Tells the cluster 'job manager' what resources you want (1 Core, 8GB memory), run on the 'classroom' nodes, and name the job 'edgeR'.

Load the R module.

Change and create directory to store results.

Run stats_edgeR.R script. This script performs the differential expression analysis.

```
$ more stats_edgeR.R

# to view stats_edgeR.R
```

Exit MobaXterm by either closing the window or typing ‘exit’ in the command prompt.

Examining the results

On Desktop

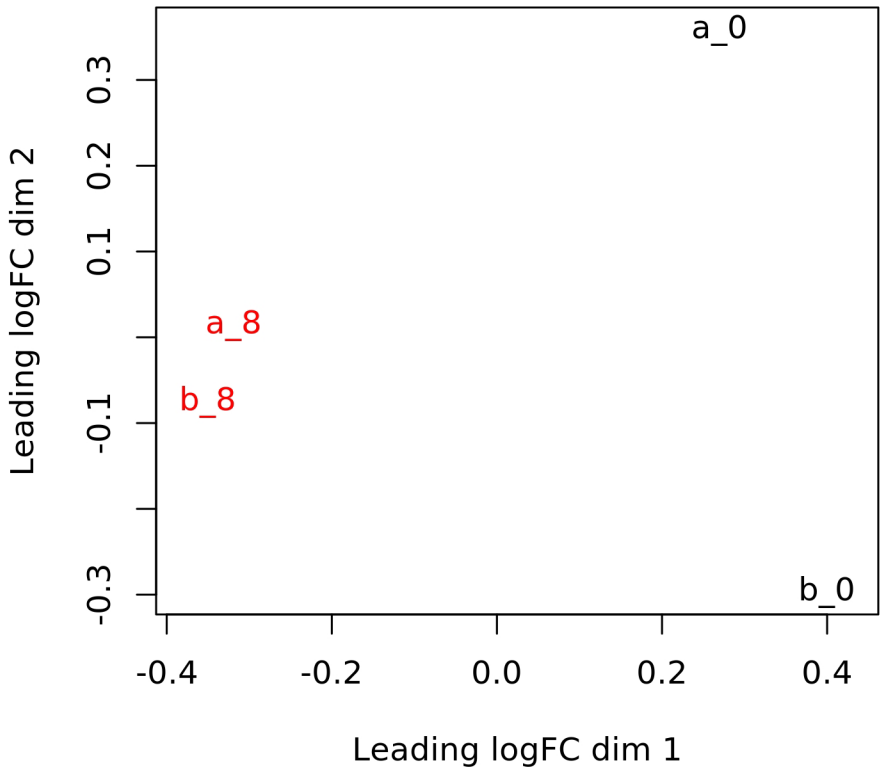
- Navigate to the following directory on your VM:

`[course_directory]\04_Transcriptomics\`

- Open MDSclustering.jpeg

Multidimensional Scaling is used to identify outliers and batch effects on large number of samples.

We used the top 500 most highly variable genes to construct this plot

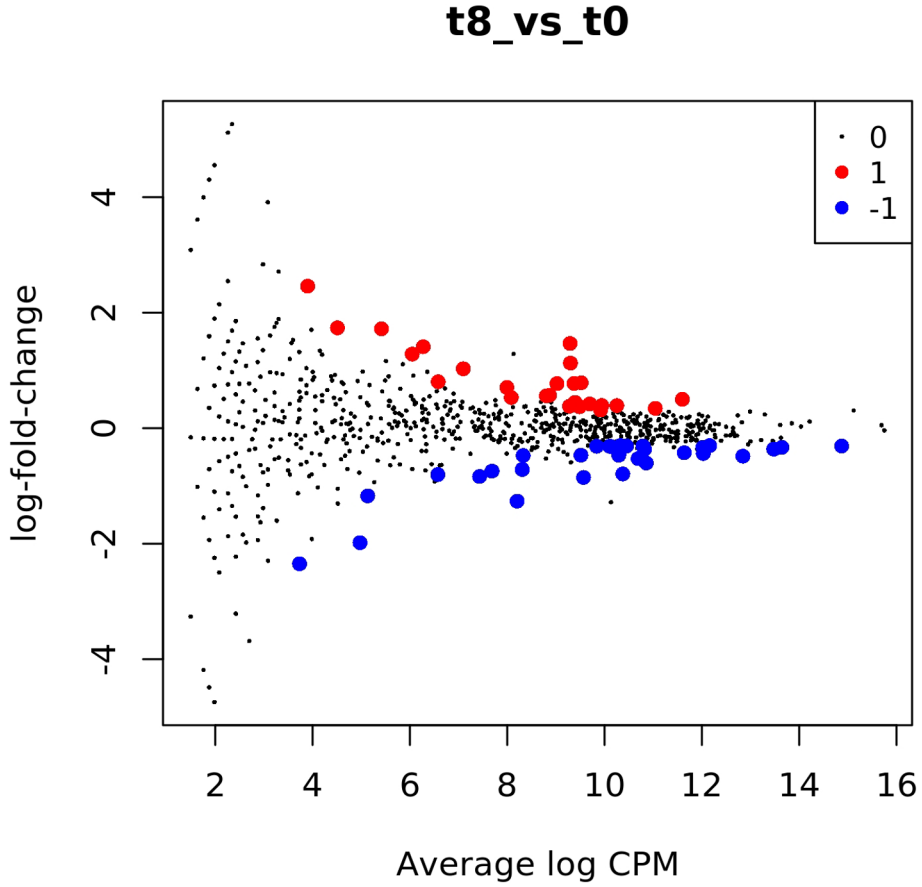


Examining the results

- Open t8_vs_t0_MeanDifferencePlot.jpeg

Each point in the plot represents a gene.

Upregulated genes are marked with red and down-regulated genes are marked with blue.



Visualization Using IGV

The **Integrative Genomics Viewer (IGV)** is a tool that supports the visualization of mapped reads to a reference genome, among other functionalities. We will use it to observe where hits were called for the *alignment* for the two samples (TP0 and TP8), and the differentially expressed genes.

Start IGV on Desktop

On Desktop

In this step, we will start **IGV** to visualize the differential expression for a selected gene.

If IGV is already open from a previous session, just close it and open again by double clicking on the IGV icon on your Desktop.

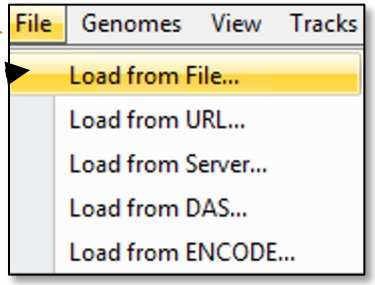
Graphical Instruction: Load Genome

1. Within IGV, click the '**Genomes**' tab on the menu bar.
2. Click the the '**Load Genome from File**' option.
3. In the browser window, Navigate to:
   `[course_directory]\04_Transcriptomics\`
4. Select **mouse_chr12.fna**
5. An index file called **mouse_chr12.fna.fai** will be automatically created in your directory that is necessary for IGV visualization.

Loading bam and GTF Files

On the menu bar, click **File**

Click **Load from File…**



Navigate to: [course_directory]\04_Transcriptomics\

Hold the **Ctrl** key down.

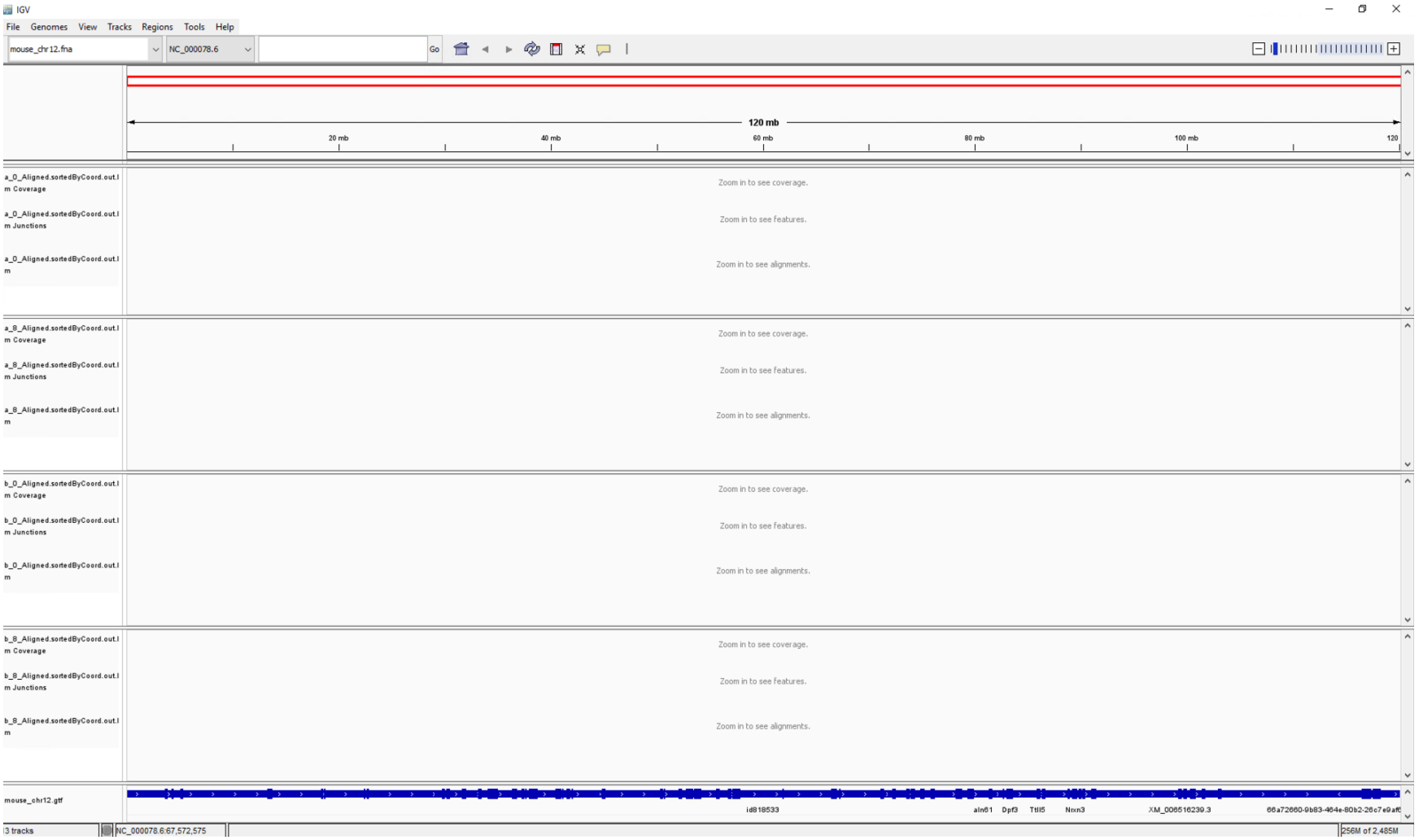
Click on these files

Click **Open.**

Files to Load

mouse_chr12.gtf
a_0_Aligned.sortedByCoord.out.bam
b_0_Aligned.sortedByCoord.out.bam
a_8_Aligned.sortedByCoord.out.bam
b_8_Aligned.sortedByCoord.out.bam

Resulting window should look like this

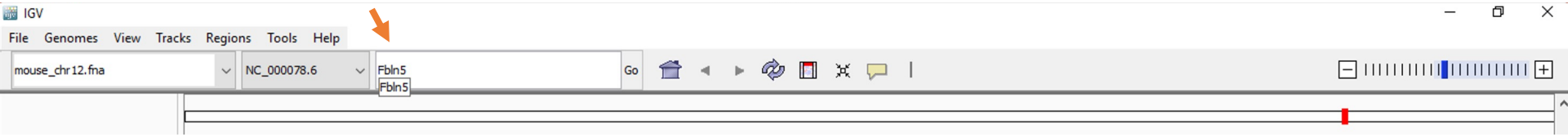


- **Fbln5** is the most significant differentially expressed gene.

- You can check this later in:

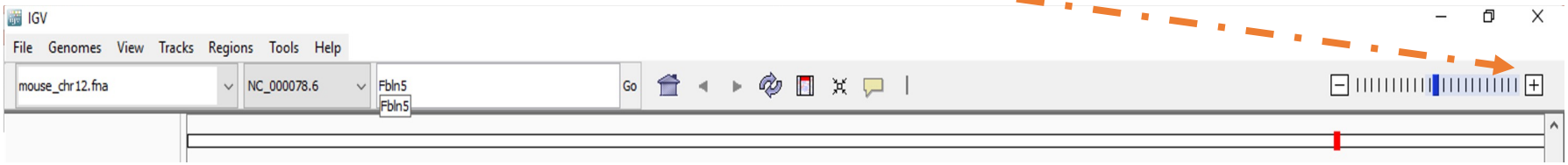
[course_directory]\04_Transcriptomics\t8_vs_t0_AllResults.txt

- Paste **Fbln5** here in the IGV window

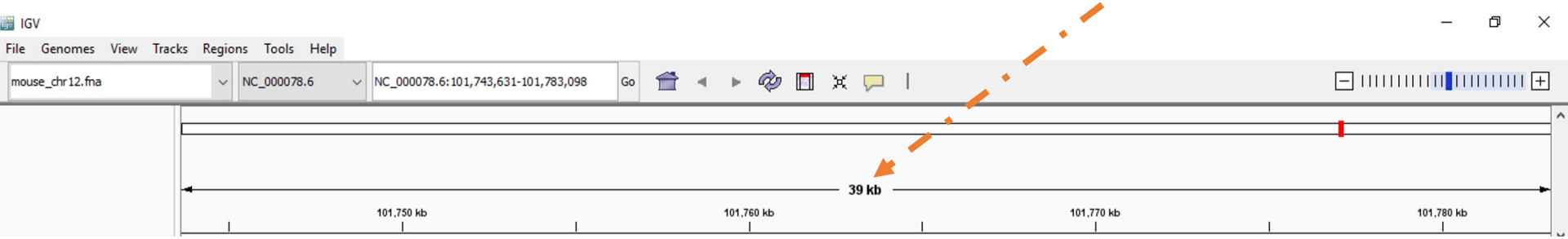


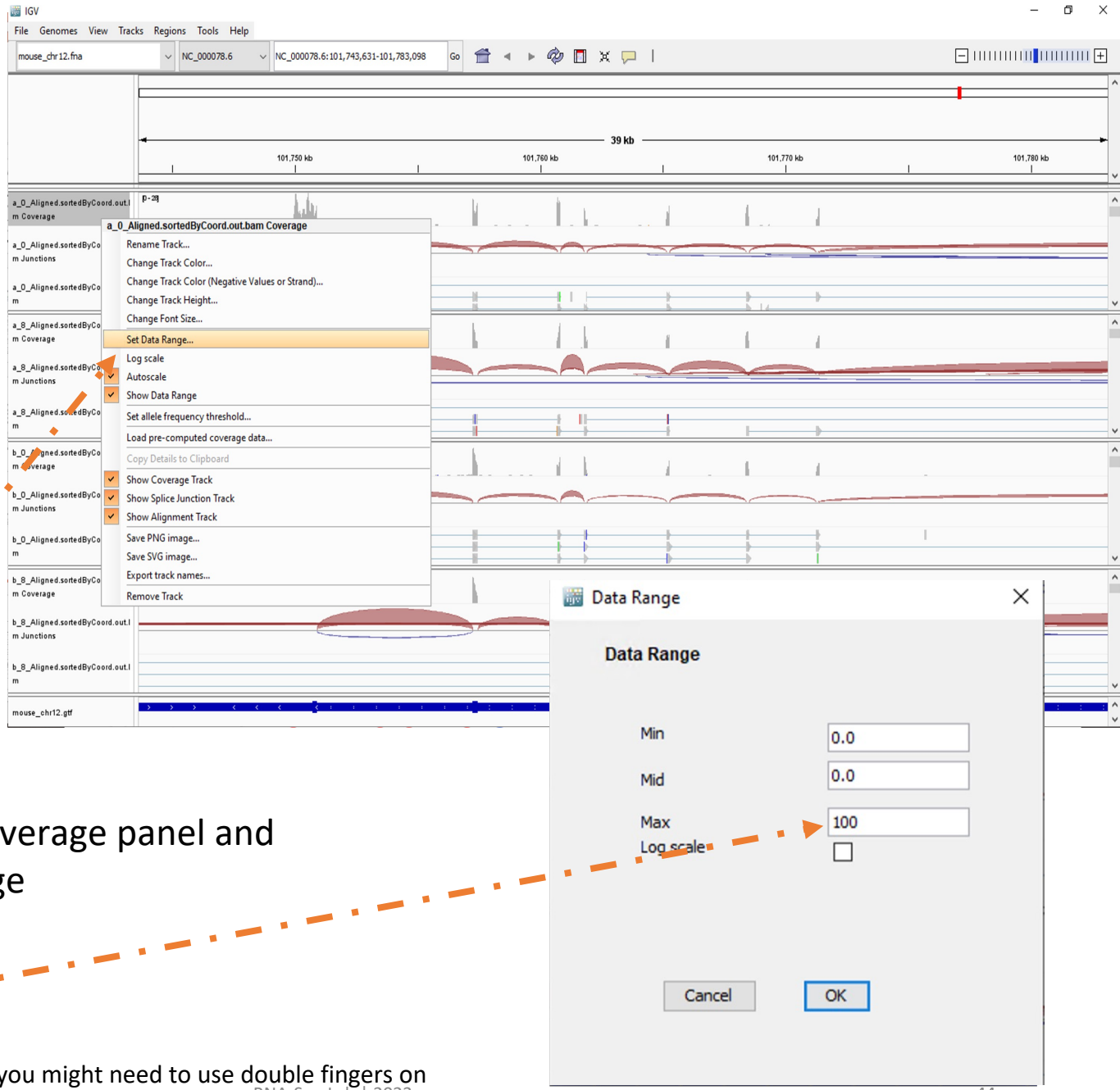
- Press Enter or click Go.

- Click on the + sign to zoom in.



- To view an image similar to next slide, Zoom in so that you see ~40 kb of the gene.

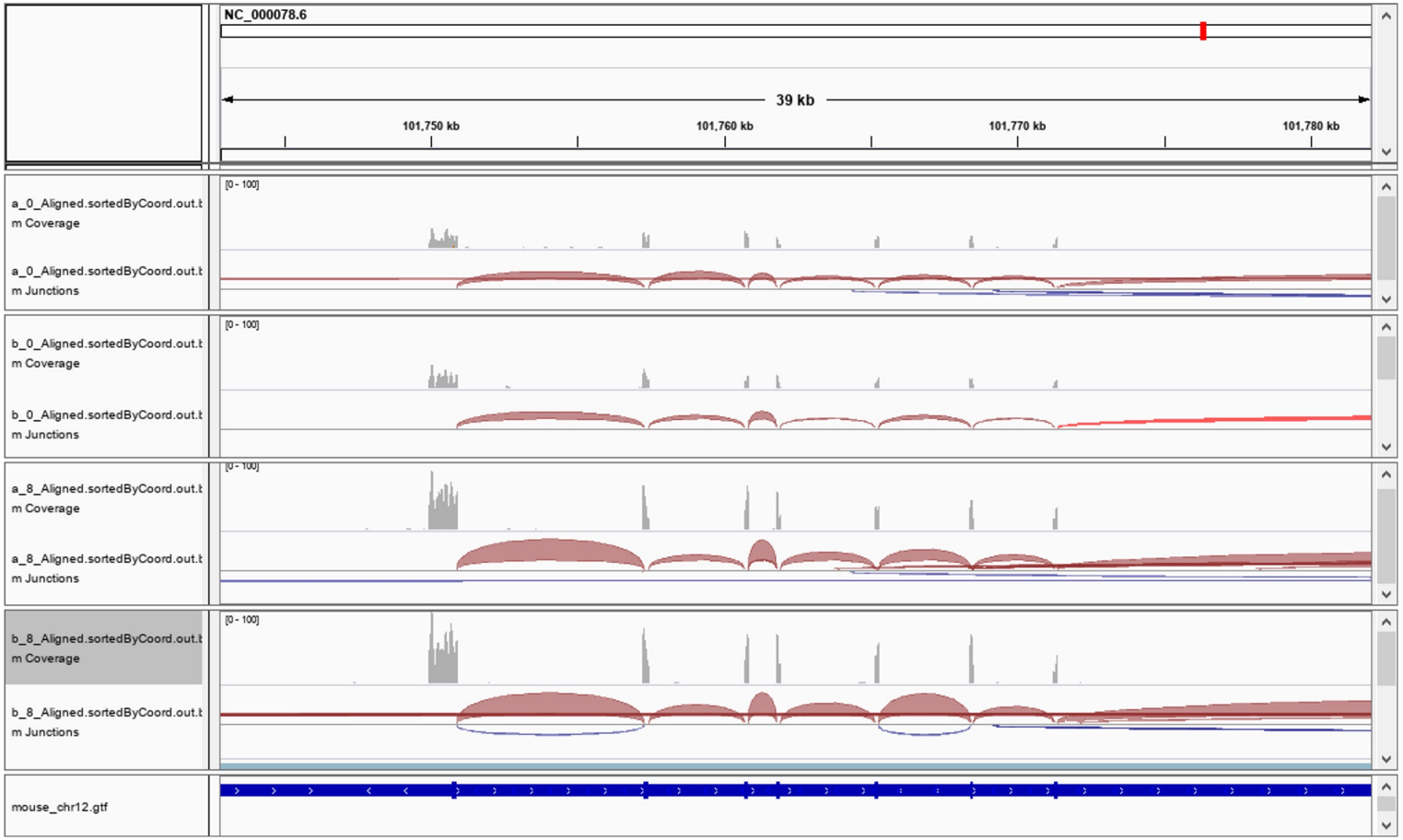




- Right click on each coverage panel and click on set Data Range

- Set the Max to 100

★For Mac users with no mouse, you might need to use double fingers on the mouse pad on the VM to be able to right-click.

Look at a differentially expressed gene

The gene appears to be more highly expressed in the TP8 time point in both replicates