

NGS workshop

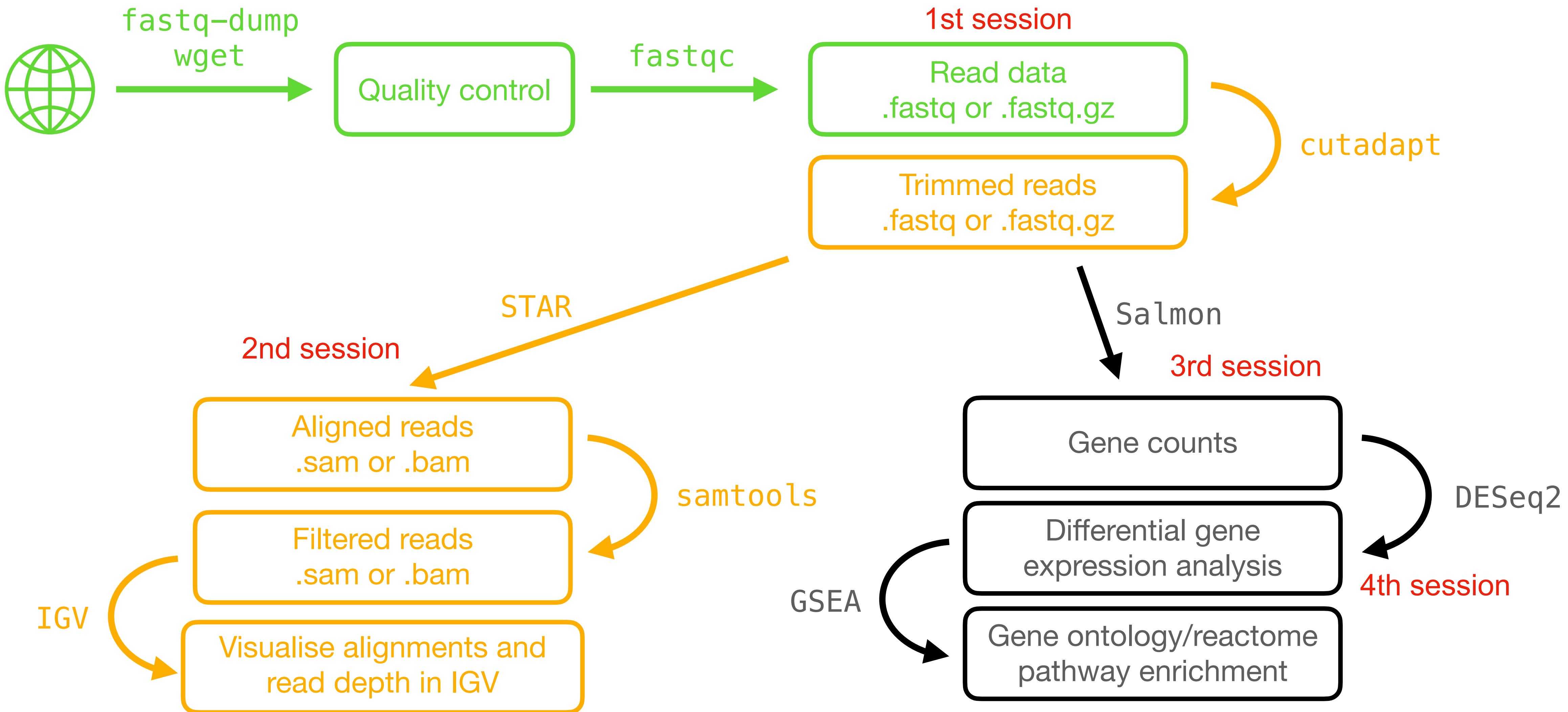
Gene count using Salmon

3rd session

10th May 2023

Mirvat Surakhy and James Carrington

Overview of the course



Quantification of Gene expression

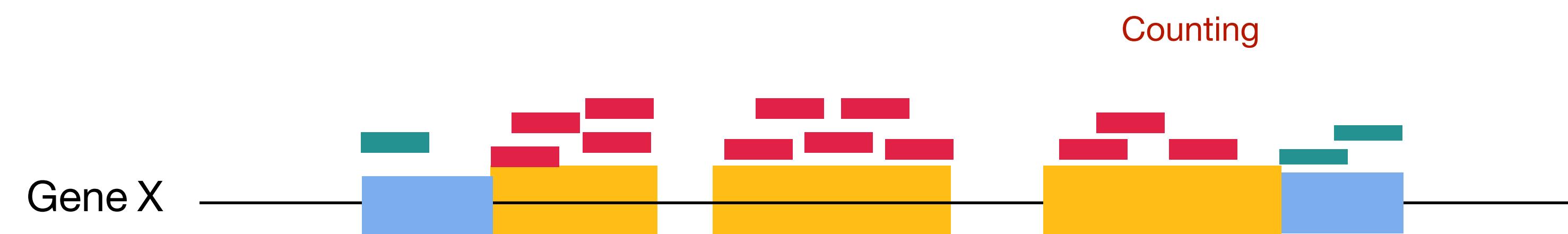
1. Gene level counting

- ◆ Reads aligned to the reference genome using transcriptome annotation (GTF) as guidance
The goal is to identify the genomic location where these reads originated from.
- ◆ Tools:
 - ◆ FeatureCounts from Subread
 - ◆ Htseq-count
- ◆ output: raw count of reads that map to a single location
i.e (the sum of reads associated with each of the exons (feature) that belong to that gene)

Mapping to reference genome

Ref: AATCCTGGGAATT CGCGTTAATTACGTTCCAA
Read: TT CGCGTT ATTACGTTGCAA

Ref: AATCCTGGGAATT CGCGTTAATTACGTTCCAA
Read: TT CGCGTT — TTACGTTGCAA



- ◆ Different transcripts of the same gene and alternative splicing are not accounted for.

Quantification of Gene expression

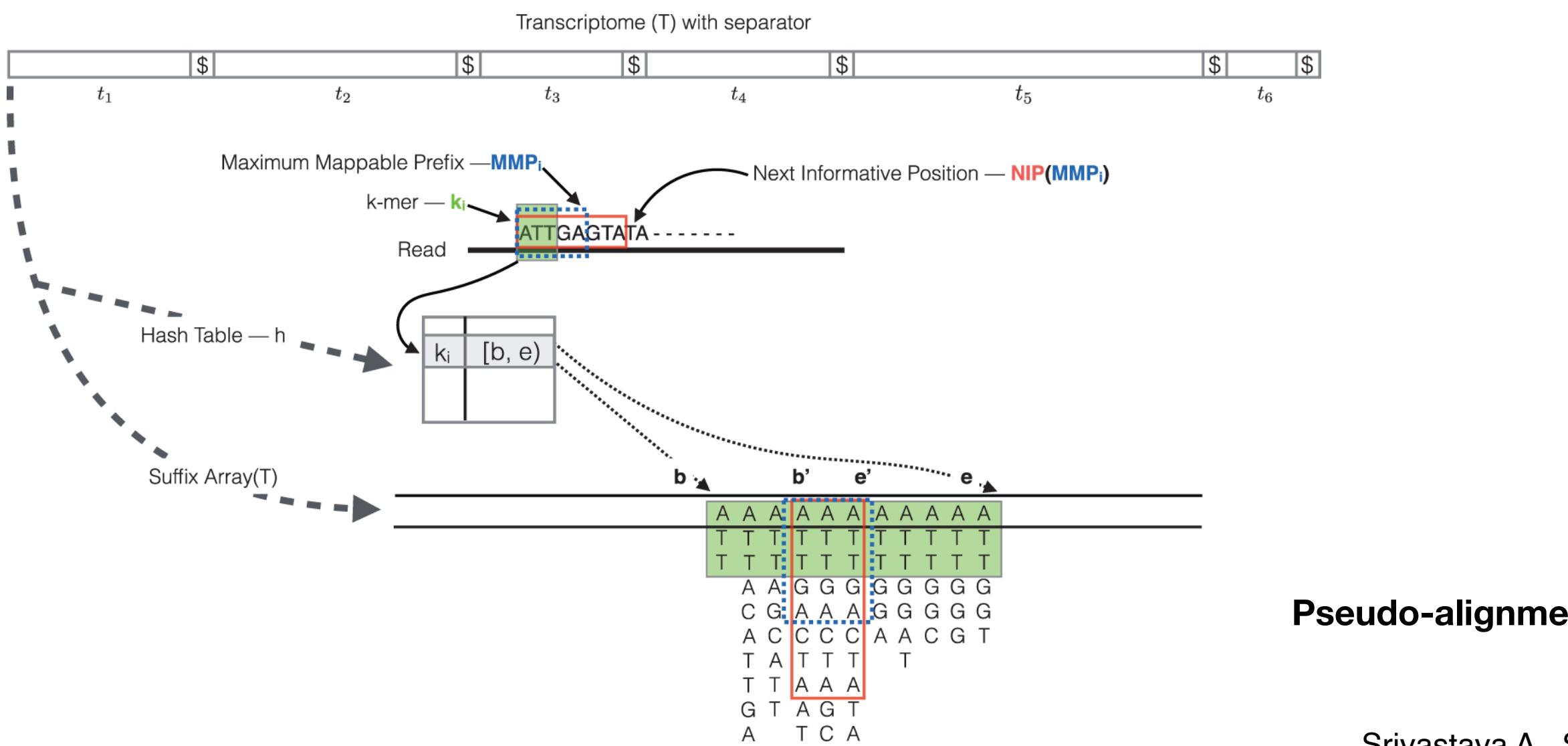
2. Transcriptome mapping and counting

- ◆ Lightweight alignment tools (**Salmon**, Kallisto)

- ◆ Quasi-mapping or pseudo-alignment

Index

- ◆ Suffix array
- ◆ A hash table (mapping each transcript to its location in the SA)



Quantification of Gene expression

2. Transcriptome mapping and counting

- ◆ Salmon utilizes sample-specific bias models for abundance estimation
- ◆ Improve the accuracy of gene expression
- ◆ Faster
- ◆ Output is a pseudo-count showing transcript expression estimates (abundance)
- ◆ Can be converted to raw gene counts for differential expression
- ◆ Cannot be used for variant calling and fusion detection

Salmon



Salmon 1.10.1 documentation

Search

Requirements

Installation

Salmon

Alevin

Salmon Output File Formats

Fragment Library Types



Build MongoDB Atlas databases
with Python, Java, C# & more.

Try it for free today.

Ad by EthicalAds

v: latest

Salmon

Salmon is a tool for **wicked-fast** transcript quantification from RNA-seq data. It requires a set of target transcripts (either from a reference or *de-novo* assembly) to quantify. All you need to run Salmon is a FASTA file containing your reference transcripts and a (set of) FASTA/FASTQ file(s) containing your reads. Optionally, Salmon can make use of pre-computed alignments (in the form of a SAM/BAM file) to the transcripts rather than the raw reads.

The **mapping-based** mode of Salmon runs in two phases; indexing and quantification. The indexing step is independent of the reads, and only needs to be run once for a particular set of reference transcripts. The quantification step, obviously, is specific to the set of RNA-seq reads and is thus run more frequently. For a more complete description of all available options in Salmon, see below.

Note

Selective alignment

Selective alignment, first introduced by the `--validateMappings` flag in salmon, and now the default mapping strategy (in version 1.0.0 forward), is a major feature enhancement introduced in recent versions of salmon. When salmon is run with selective alignment, it adopts a considerably more sensitive scheme that we have developed for finding the potential mapping loci of a read, and score potential mapping loci using the chaining algorithm introduced in minimap2⁵. It scores and validates these mappings using the score-only, SIMD, dynamic programming algorithm of ksw2⁶. Finally, we recommend using selective alignment with a *decoy-aware* transcriptome, to mitigate potential spurious mapping of reads that actually arise from some unannotated genomic locus that is sequence-similar to an annotated transcriptome. The selective-alignment algorithm, the use of a *decoy-aware* transcriptome, and the influence of running salmon with different mapping and alignment strategies is covered in detail in the paper [Alignment and mapping methodology influence transcript abundance estimation](#).

The use of selective alignment implies the use of range factorization, as mapping scores become very meaningful with this option. Selective alignment can improve the accuracy, sometimes considerably, over the faster, but less-precise mapping algorithm that was previously used. Also, there are a number of options and flags that allow the user to control details about how the scoring is carried out, including setting match, mismatch, and gap scores, and choosing the minimum score below which an alignment will be considered invalid, and therefore not used for the purposes of quantification.

The **alignment-based** mode of Salmon does not require indexing. Rather, you can simply provide Salmon with a FASTA file of the transcripts and a SAM/BAM file containing the alignments you wish to use for quantification.

Salmon is, and will continue to be, [freely and actively supported on a best-effort basis](#). If you are in

ON THIS PAGE

Using Salmon

Preparing transcriptome indices
(mapping-based mode)

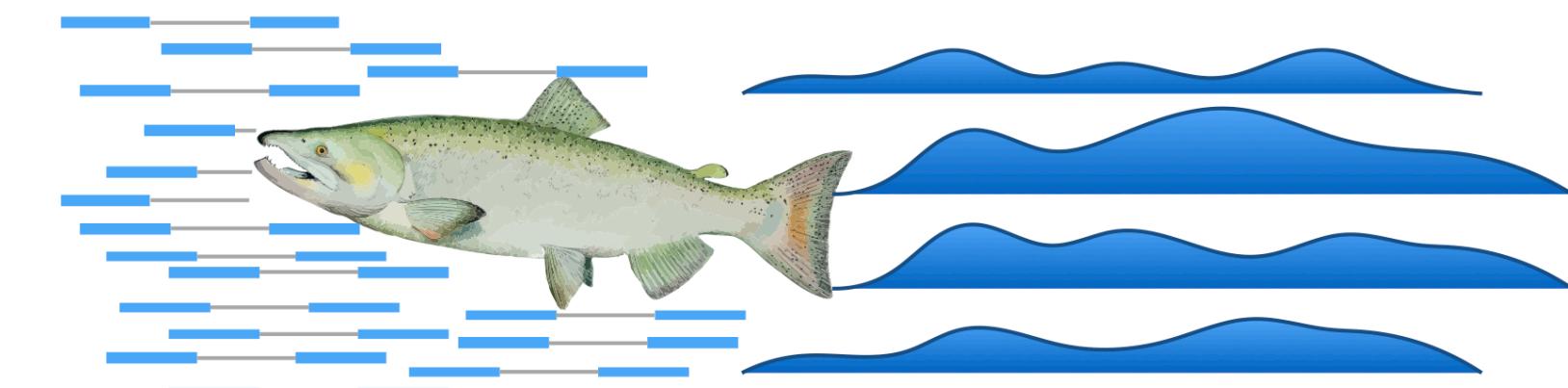
Quantifying in mapping-based
mode

Providing multiple read files to
Salmon

Quantifying in alignment-based
mode

Description of some important
options

`--mimicBT2`
`--mimicStrictBT2`
`--meta`
`--recoverOrphans`
`--hardFilter`
`--skipQuant`
`--allowDovetail`
`-p / --threads`
`--dumpEq`
`--incompatPrior`
`--fldMean`
`--fldSD`
`--minScoreFraction`
`--bandwidth`
`--maxMMPExtension`
`--ma`
`--mp`
`--go`
`--ge`
`--rangeFactorizationBins`
`--useEM`
`--numBootstraps`
`--numGibbsSamples`
`--seqBias`
`--gcBias`
`--posBias`
`--biasSpeedSamp`
`--writeUnmappedNames`



<https://combine-lab.github.io/salmon/>

<https://salmon.readthedocs.io/en/latest/salmon.html#using-salmon>

Practical

- ◆ Create an index to the transcriptome with Salmon (demo)
- ◆ Quantify transcript expression using Salmon
- ◆ Extract raw gene count in R using tximport

Salmon index... required files

- Files are from the Ensembl website <https://www.ensembl.org/info/data/ftp/index.html>

The screenshot shows the Ensembl FTP Download page. It features a search bar at the top and a sidebar with links for 'Using this website', 'Annotation and prediction', 'Data access' (which is selected), 'API & software', and 'About us'. Below the sidebar, there's a 'FTP Download' section with a 'Custom data sets' button. The main content area displays tables for 'Database' and 'Single species data'. The 'Database' table includes rows for Comparative genomics, BioMart, and Stable ids. The 'Single species data' table lists Human and Mouse entries with various file types like FASTA, GTF, TSV, and VCF. A note at the bottom states: 'This website requires cookies, and the limited processing of your personal data in order to function. By using the site you are agreeing to this as outlined in our Privacy Policy and Terms of Use'.

Index of /pub/release-109/fasta/homo_sapiens/cdna

Name	Last modified	Size	Description
Parent Directory		-	
CHECKSUMS	2023-01-27 13:44	118	
Homo_sapiens.GRCh38.cdna.ab initio.fa.gz	2022-12-13 11:44	20M	
Homo_sapiens.GRCh38.cdna.all.fa.gz	2022-12-13 11:30	75M	
README	2022-12-13 11:44	2.5K	

Salmon Index Don't run

Ready to use for the workshop: /mnt/beegfs/workshop/A_reference_files/salmon_index

```
GNU nano 4.8                                         New Buffer                                         Modified
#!/bin/bash
#SBATCH --job-name=Salmon_Index
#SBATCH --cpus-per-task=6
#SBATCH --mem=10000
#SBATCH --output=outfile.%j
#SBATCH --error=errfile.%j

#this script generates Salmon Index

salmon index \
-t Homo_sapiens.GRCh38.cdna_dna.fa \
-i salmon_index \
--decoys decoys.txt

##-t: the path to the transcriptome and dna combined files (in FASTA format)
#-i: the path to the folder to store the indices generated
#-k: the length of kmer to use to create the indices. -k 31 is the default value.
#It is optimized for 75bp or longer reads
#--decoys: text file that lists the names of the chromosomes
```

^G Get Help **^O** Write Out **^W** Where Is **^K** Cut Text **^J** Justify **^C** Cur Pos **M-U** Undo
^X Exit **^R** Read File **^** Replace **^U** Paste Text **^T** To Spell **^_** Go To Line **M-E** Redo

Fastq files for DGE

```
$cd /mnt/beegfs/workshop/A_demo_files/  
files_for_differentialExpression/fastq_files$ ls -lh  
total 723M  
-rw-rw-r-- 1 path1126 proudfoot 90M Apr 17 14:21 5aza_rep1_sample_1.fastq.gz  
-rw-rw-r-- 1 path1126 proudfoot 91M Apr 17 14:21 5aza_rep1_sample_2.fastq.gz  
-rw-rw-r-- 1 path1126 proudfoot 90M Apr 17 14:21 5aza_rep2_sample_1.fastq.gz  
-rw-rw-r-- 1 path1126 proudfoot 91M Apr 17 14:21 5aza_rep2_sample_2.fastq.gz  
-rw-rw-r-- 1 path1126 proudfoot 91M Apr 17 14:21 DMSO_rep1_sample_1.fastq.gz  
-rw-rw-r-- 1 path1126 proudfoot 91M Apr 17 14:21 DMSO_rep1_sample_2.fastq.gz  
-rw-rw-r-- 1 path1126 proudfoot 91M Apr 17 14:21 DMSO_rep2_sample_1.fastq.gz  
-rw-rw-r-- 1 path1126 proudfoot 91M Apr 17 14:21 DMSO_rep2_sample_2.fastq.gz
```

Salmon quants for abundance estimation

```
salmon quant -i location of the salmon index folder \
-l A To allow Salmon to automatically infer the library type \
-1 R1.fastq.gz \
-2 R2.fastq.gz \
-o specify the output directory \
--seqBias \
--gcBias \
-p specifies the number of processors or cores we would like to use for
multithreading
```

```
#to optimize abundance estimates (more accurate)
#--seqBias --gcBias to correct for seq bias and gc bias
```

Salmon quant

```
$ cd /mnt/beegfs/workshop/<SSO>
$ mkdir -p results/salmon_quants
$ nano
```

```
GNU nano 4.8                                         New Buffer                                         Modified
#!/bin/bash
#SBATCH --job-name=Salmon_quant
#SBATCH --cpus-per-task=8
#SBATCH --mem=24000
#SBATCH --output=outfile_salmon.%j
#SBATCH --error=errfile_salmon.%j

cd /mnt/beegfs/workshop/A_demo_files/files_for_differentialExpression/fastq_files

for file in *_1.fastq.gz; do
    filename=$(basename "$file" _1.fastq.gz)   ###create a prefix called filename

    salmon quant -i /mnt/beegfs/workshop/A_reference_files/salmon_index \
    -l A \
    -1 ${filename}_1.fastq.gz \
    -2 ${filename}_2.fastq.gz \
    -o /mnt/beegfs/workshop/<SSO>/results/salmon_quants/${filename}_quant \
    --seqBias \
    --gcBias \
    -p 8

done
```

^G Get Help **^O** Write Out **^W** Where Is **^K** Cut Text **^J** Justify **^C** Cur Pos **M-U** Undo
^X Exit **^R** Read File **^** Replace **^U** Paste Text **^T** To Spell **^_** Go To Line **M-E** Redo

- ◆ Save your script as **salmon_quant.bash**
- ◆ Run the script and check the status of the job
 - What module to use?
 - What command do you use to run the job?
 - What command do you use to check the status of your job?

◆ What module to use?

- `module avail` ###optional for now
- `module load SALMON/1.9.0`

◆ Run the script

- `sbatch salmon_quant.bash`

◆ Check if the script is running

- `squeue -u <SSO>`

Output of salmon quant

- ◆ When your job has finished navigate to the output directory

```
cd /mnt/beegfs/workshop/<SSO>/results/salmon_quants
```

- ◆ View the content of the folder (what command do we use to list the files?)

quants file

```
$ ls -lh
```

```
drwxrwxr-x 5 jesu2166 hassan      6 May  5 12:10 5aza_rep1_sample_quant
drwxrwxr-x 5 jesu2166 hassan      6 May  5 12:15 5aza_rep2_sample_quant
drwxrwxr-x 5 jesu2166 hassan      6 May  5 12:19 DMSO_rep1_sample_quant
drwxrwxr-x 5 jesu2166 hassan      6 May  5 12:24 DMSO_rep2_sample_quant
```

Navigate to one of the folders

```
$ cd 5aza_rep1_sample_quant
```

```
$ ls -lh
```

```
total 12M
```

```
drwxrwxr-x 2 jesu2166 hassan  12 May  5 12:10 aux_info
-rw-rw-r-- 1 jesu2166 hassan 393 May  5 12:06 cmd_info.json
-rw-rw-r-- 1 jesu2166 hassan 575 May  5 12:10 lib_format_counts.json
drwxrwxr-x 2 jesu2166 hassan   1 May  5 12:10 libParams
drwxrwxr-x 2 jesu2166 hassan   1 May  5 10:59 logs
-rw-rw-r-- 1 jesu2166 hassan 12M May  5 12:10 quant.sf
```

View the content of the quant.sf

\$less quant.sf

Name	Length	EffectiveLength	TPM	NumReads
ENST00000631435.1	12	12.000	0.000000	0.000
ENST00000415118.1	8	7.000	0.000000	0.000
ENST00000448914.1	13	13.000	0.000000	0.000
[ENST00000434970.2	9	9.000	0.000000	0.000
ENST00000632524.1	11	11.000	0.000000	0.000
ENST00000633009.1	20	19.000	0.000000	0.000
ENST00000634070.1	18	17.000	0.000000	0.000
ENST00000632963.1	20	19.000	0.000000	0.000
ENST00000633030.1	19	18.000	0.000000	0.000
.				
.				
.				
ENST00000620516.4	1179	1145.496	12.961913	5.427

TPM: Transcripts Per Million

Press q to exit the less command

Extract the raw counts from the quants file

We will be using R and R studio

- ◆ R: programming language and environment for data manipulation, statistical computing, and graphical display
- ◆ R: Free and open source <https://www.r-project.org/>
- ◆ R studio: Free and open source IDE (Integrated Development Environment) for R. Available for Mac, Windows and Linux



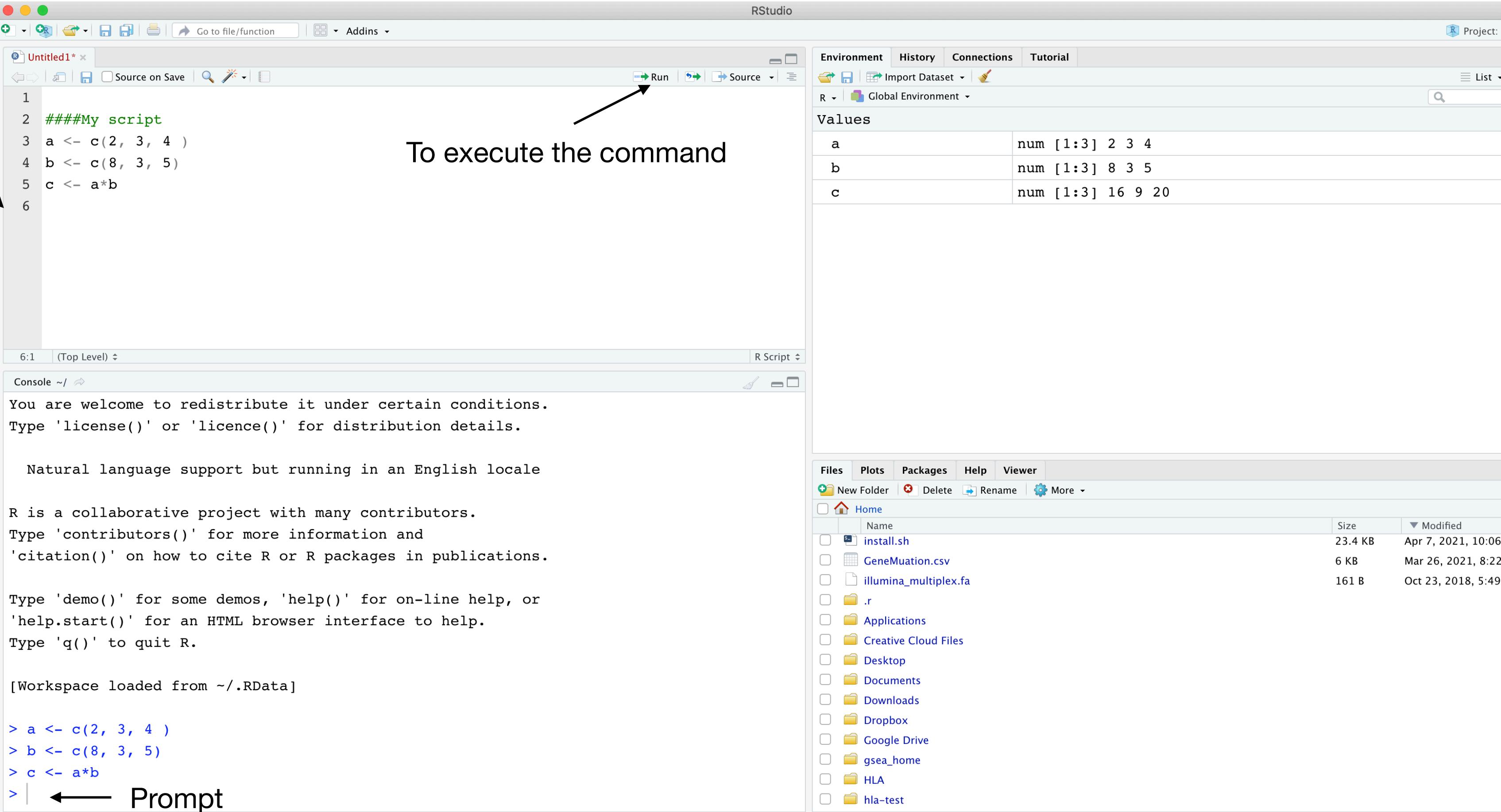
Rstudio interface

Script editor

To execute the command

Object, history and Environment

R Console



Tree of folders, graph window,
Packages, help and Viewrt

Best practices when using R

- ◆ Document everything you do so your code is reproducible.
- ◆ Use the # sign to comment. In R anything after the # is ignored.
- ◆ Write the code in the script editor rather than R console.
- ◆ Keep track of all the libraries used in the analysis together with the R version.

R script to extract the raw count and save it

- ◆ Before running R we need to create an output directory to save the results
- ◆ `cd /mnt/beegfs/workshop/<SSO>/`
- ◆ `mkdir /mnt/beegfs/workshop/<SSO>/results/counts`
- ◆ Use the NoMachine
- ◆ `$ srun -c 8 --mem 8000 --X11 --pty bash`
- ◆ Are you sure you want to continue connecting (yes/no/[fingerprint])? `yes`
- ◆ SSO@linux020's password: Type your SSO password
- ◆ `$ module load RSTUDIO TXIMETA/3.15 R_NGS_ANALYSIS/1.0`
- ◆ `$ rstudio`

R script.... tximport.r

Open an R script

File > New_File> R Script

Save it in your working directory: File > save > tximport.r

```
###To install libraries in R, demo only, don't run all done for this course #####
```

```
#if (!require("BiocManager", quietly = TRUE))
```

```
# install.packages("BiocManager")
```

```
#BiocManager::install("tximport")
```

```
#install.packages("dplyr")
```

```
#Load the required libraries
```

```
library("tximport")
```

```
library("dplyr")
```

```
## assign your working directory
```

```
setwd ("~/mnt/beegfs/workshop/<SSO>/results")
```

```
## List all directories containing data
```

```
samples <- list.files(path = "./salmon_quants", full.names = T, pattern = "_sample_quant$")
```

```
## Obtain a vector of all filenames including the path
```

```
files <- file.path(samples, "quant.sf")
```

```
## list all the files to the console
```

```
files
```

```
##assign a shorter name for each element
```

```
names(files) <- list.files("salmon_quants")
```

R script.... tximport.r

```
## Read the annotation file

tx2gene <- read.csv("/mnt/beegfs/workshop/A_reference_files/tx2gene_ens109.csv")

# Run tximport

txi <- tximport(files, type="salmon", tx2gene=tx2gene[,c("TXNAME", "GENEID")], countsFromAbundance="lengthScaledTPM",
ignoreTxVersion = TRUE)

## Check the output

head(txi[["counts"]])

## Extract the counts, round the values and change the list to a data frame

counts <- txi$counts %>% round() %>% data.frame()

## Change the column names

colnames(counts) <- c("5aza_rep1", "5aza_rep2", "DMSO_rep1", "DMSO_rep2" )

View(counts)

### Save the results for the next session using the write.table command

write.table(counts, "counts/Count_fromtximport_Salmon.txt", sep ="\t", quote = F)

####keep a note of the packages and their versions

sessionInfo()
```

Before the end of this session

- Save your code
- Quite Rstudio
- cancel your job for the Rstudio session otherwise, your group will be charged as long as the job is running
- **scancel job ID**
- or use: **scancel -u <SSO>** #this will cancel all your jobs