PM4NGS Documentation

Release 1.0.0

Roberto Vera Alvarez

Jun 28, 2021

CONTENTS

1 RNA-seq of the lobe-less (lol) mutants in Drosophila melanogaster **1**
- 1.1 BioProject . . . 2
- 1.2 SRA . . . 3

2 GCP Instance Setup **7**
- 2.1 Create a GCP instance . . . 7
- 2.2 Accessing the instance with SSH . . . 9
- 2.3 Installing dependencies on the GCP instance with Ubuntu . . . 10
- 2.4 Testing the Docker daemon . . . 10

3 Installing PM4NGS **11**
- 3.1 What is a Python virtual environment? . . . 11
- 3.2 Installing PM4NGS . . . 11
- 3.3 Testing PM4NGS . . . 12

4 The Jupyter Server **13**
- 4.1 Accessing the instance with SSH . . . 13
- 4.2 Activating pm4ngs_venv virtual environment . . . 14
- 4.3 Running the Jupyter Server . . . 14
- 4.4 Open the jupyter web interface in your local computer . . . 15

5 PM4NGS for RNA-Seq data analysis **17**
- 5.1 Sample sheet . . . 19
- 5.2 Creating the PM4NGS RNA-Seq project . . . 20
- 5.3 Pre-processing QC . . . 24
- 5.4 Samples trimming . . . 31
- 5.5 Alignment and Quantification . . . 36
- 5.6 Differential Gene Expression Analysis . . . 44
  - 5.6.1 Volcano Plots . . . 46
  - 5.6.2 Sample correlation . . . 47
  - 5.6.3 Expression correlation . . . 47
  - 5.6.4 PCA plots . . . 47
  - 5.6.5 List of differentially expressed genes . . . 48
- 5.7 Gene Ontology Enrichment . . . 48

6 Introduction **53**

7 Background reading **55**

8 Help and Support **57**

CHAPTER

ONE

RNA-SEQ OF THE LOBE-LESS (LOL) MUTANTS IN DROSOPHILA MELANOGASTER

Lobe-less (lol) encodes a long noncoding RNA (lncRNA) required for the development of Drosophila mushroom body, a center of insect memories and learning. RNA-seq was carried out in the control (w[1118]) and lol mutant flies to identify genes regulated by lol lncRNA.

From the manuscript: Genetic interactions between Protein Kinase D and Lobe mutants during eye development of Drosophila melanogaster

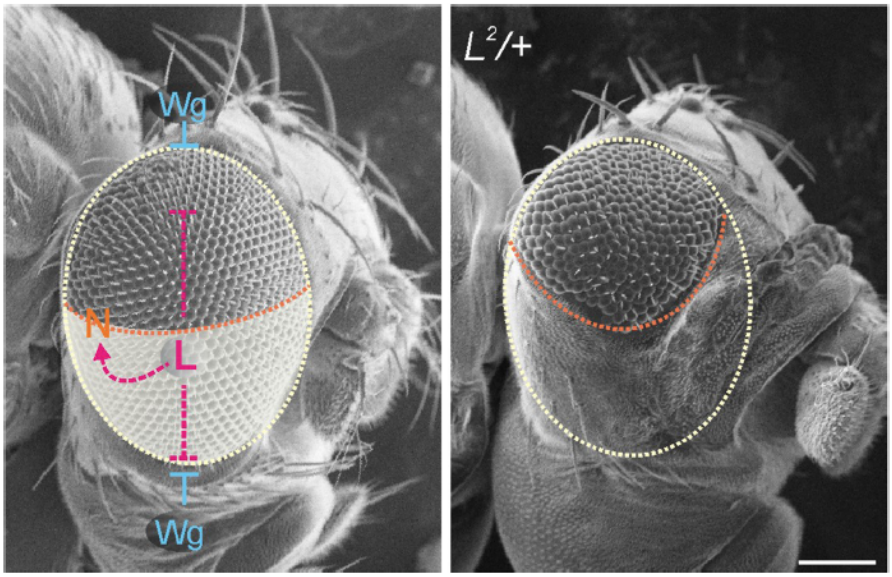
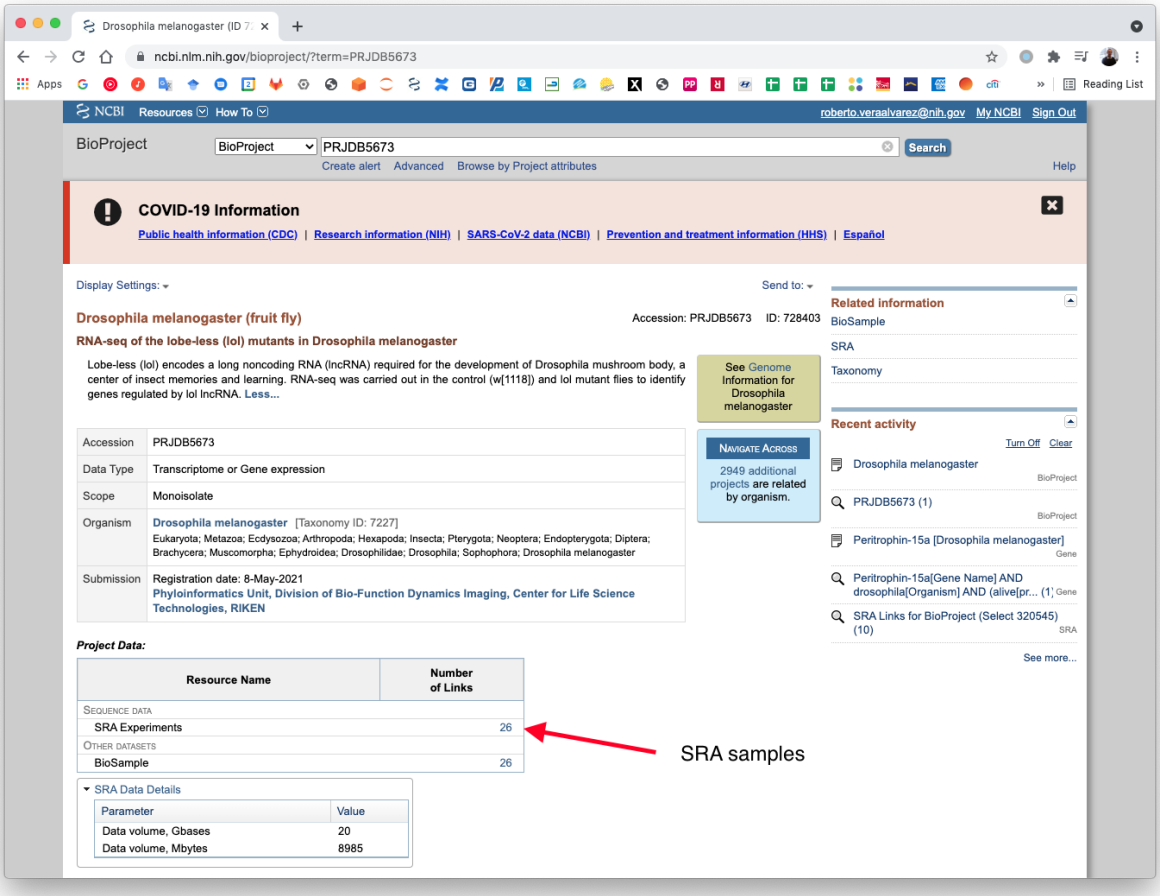


Fig. 1 Development of the ventral eye is affected by the *L* mutation. During development, the *Drosophila* eye field is confined by the activity of Wg. Establishment of the dorso-ventral boundary requires the activation of N. Concluded from genetic interactions, *Lobe* acts as a positive regulator of N and a negative regulator of Wg activities. Accordingly, L^2/+ mutant flies display small eyes, where the ventral eye compartment is reduced at the expense of head epidermis. Arrows indicate positive, and bars negative interactions. Size bar, 100 µm

1.1 BioProject

A BioProject is a collection of biological data related to a single initiative, originating from a single organization or from a consortium. A BioProject record provides users a single place to find links to the diverse data types generated for that project.

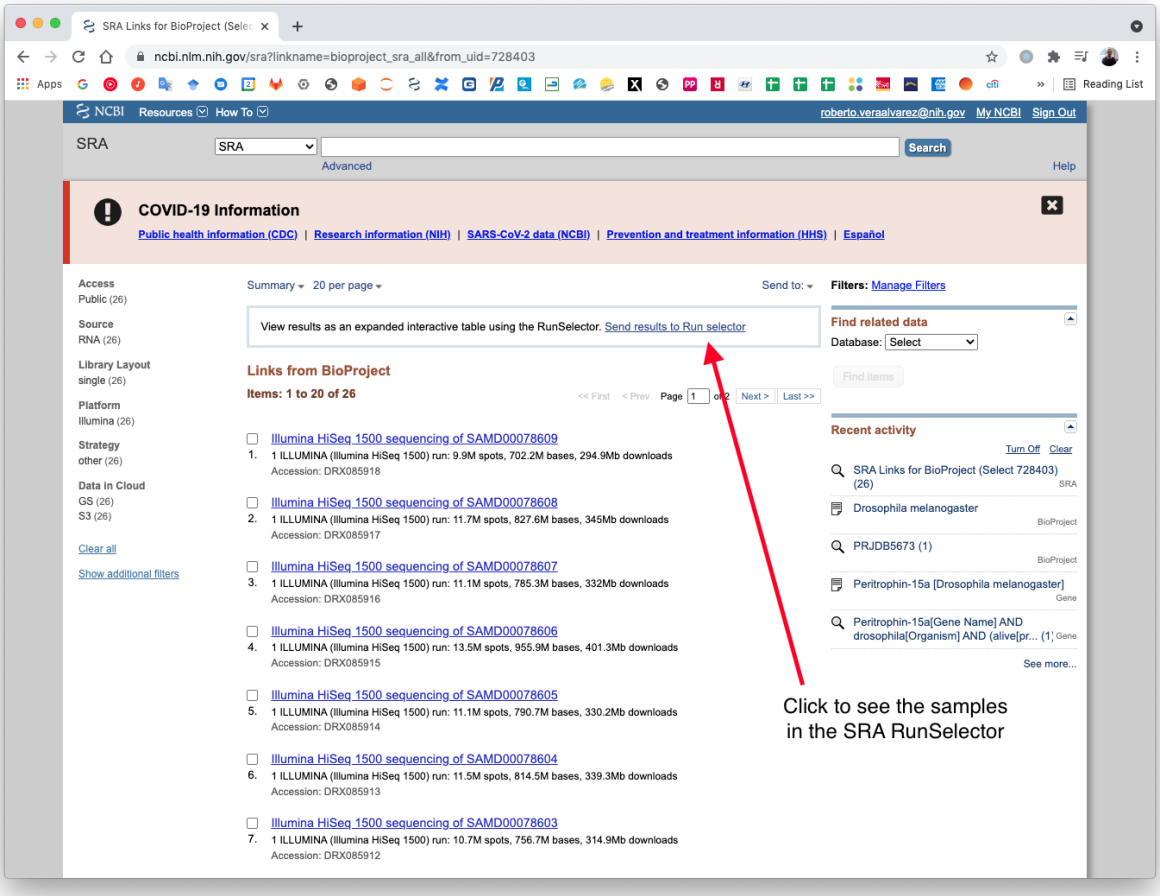
BioProject: PRJDB5673



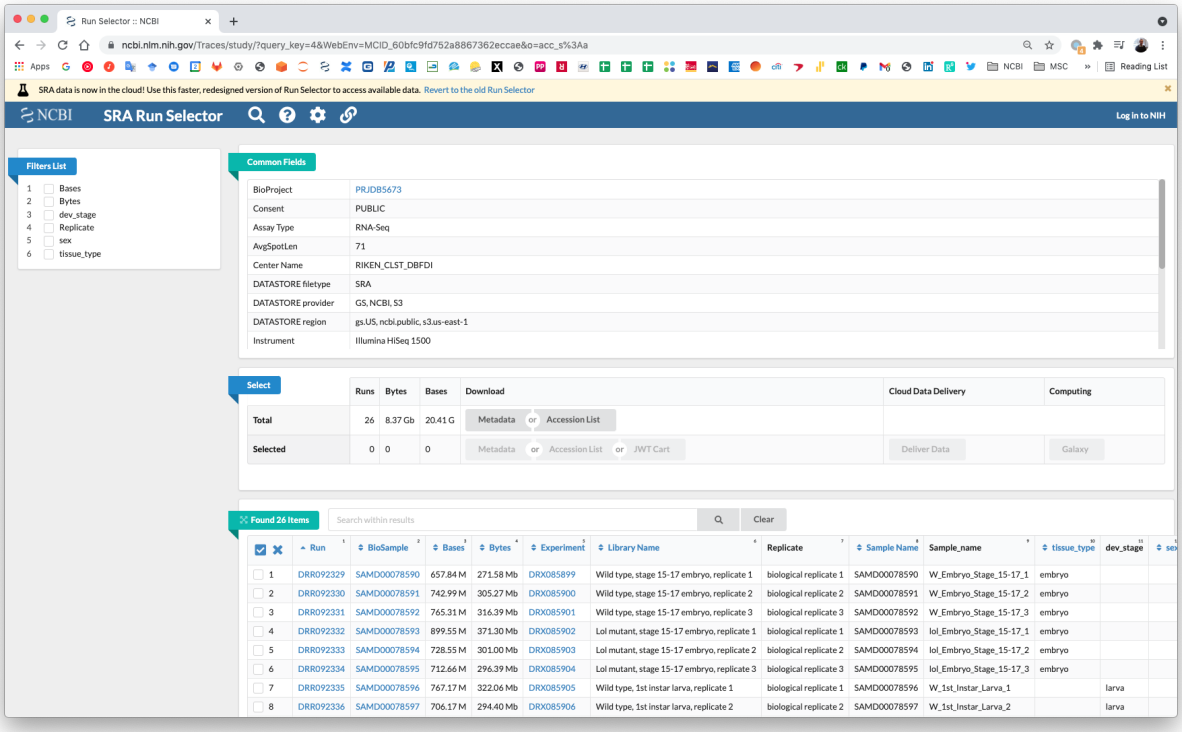
1.2 SRA

Sequence Read Archive (SRA) data, available through multiple cloud providers and NCBI servers, is the largest publicly available repository of high throughput sequencing data.

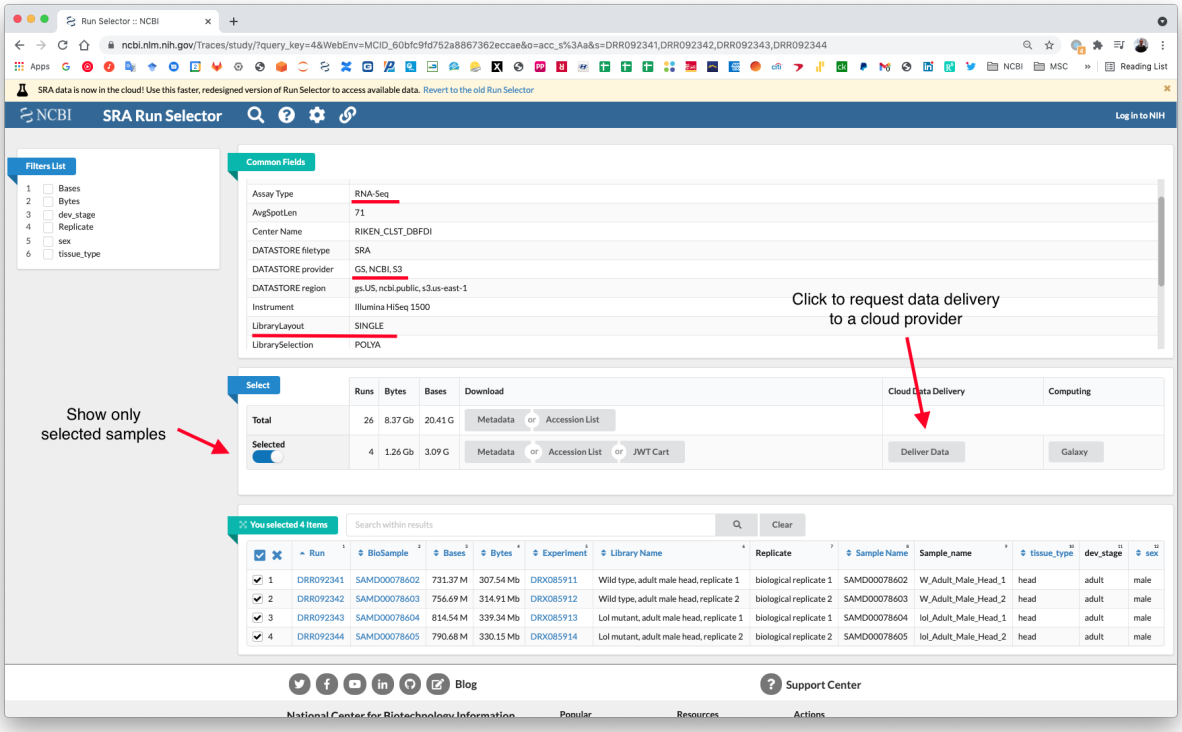
List of SRA samples in the BioProject



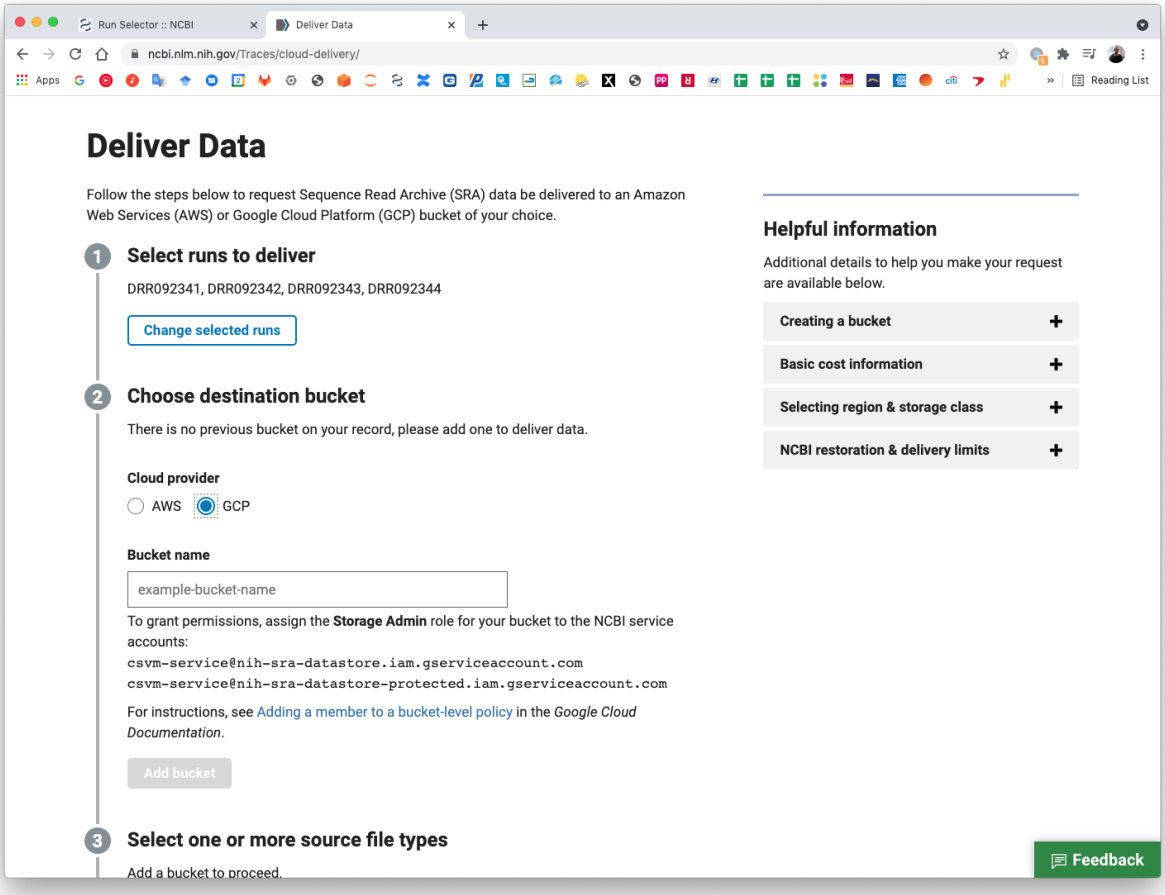
Click on the **Send results to Run selector** to see the SRA data in the Run selector app.



Select samples to be analyzed: DRR092341, DRR092342, DRR092343, DRR092344. Wild type, adult male head compared with Lol mutant, adult male head.



Click on **Deliver Data** to request NCBI to transfer the data to a cloud bucket.


Run Selector :: NCBI
Deliver Data
ncbi.nlm.nih.gov/Traces/cloud-delivery/
Apps
Reading List
Deliver Data
Follow the steps below to request Sequence Read Archive (SRA) data be delivered to an Amazon Web Services (AWS) or Google Cloud Platform (GCP) bucket of your choice.
1
Select runs to deliver
DRR092341, DRR092342, DRR092343, DRR092344
Change selected runs
2
Choose destination bucket
There is no previous bucket on your record, please add one to deliver data.
Cloud provider
AWS
GCP
Bucket name
example-bucket-name
To grant permissions, assign the Storage Admin role for your bucket to the NCBI service accounts:
csvm-service@nih-sra-datastore.iam.gserviceaccount.com
csvm-service@nih-sra-datastore-protected.iam.gserviceaccount.com
For instructions, see Adding a member to a bucket-level policy in the Google Cloud Documentation.
Add bucket
3
Select one or more source file types
Add a bucket to proceed.
Helpful information
Additional details to help you make your request are available below.
Creating a bucket
Basic cost information
Selecting region & storage class
NCBI restoration & delivery limits
Feedback

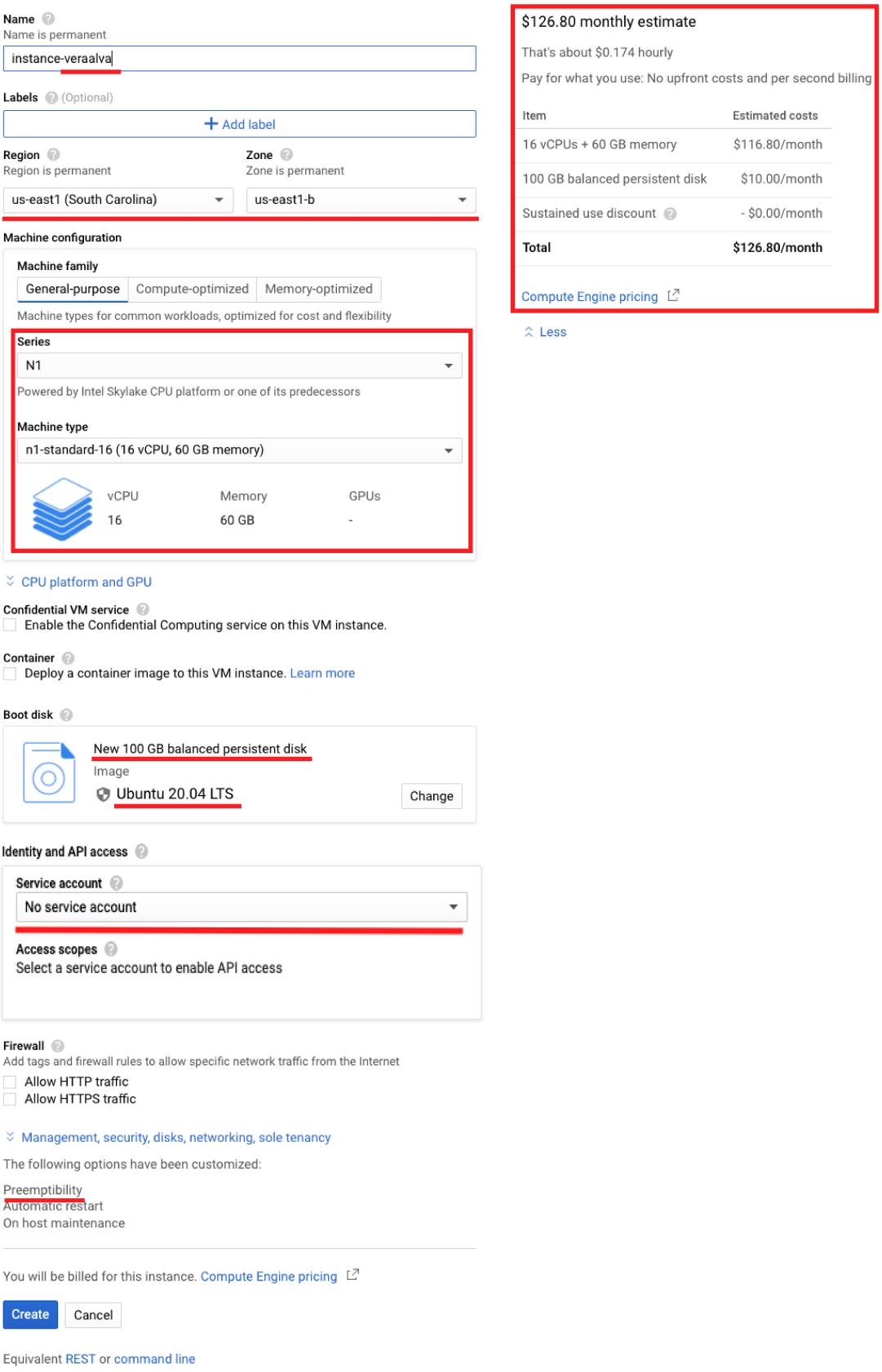
CHAPTER

TWO

GCP INSTANCE SETUP

2.1 Create a GCP instance

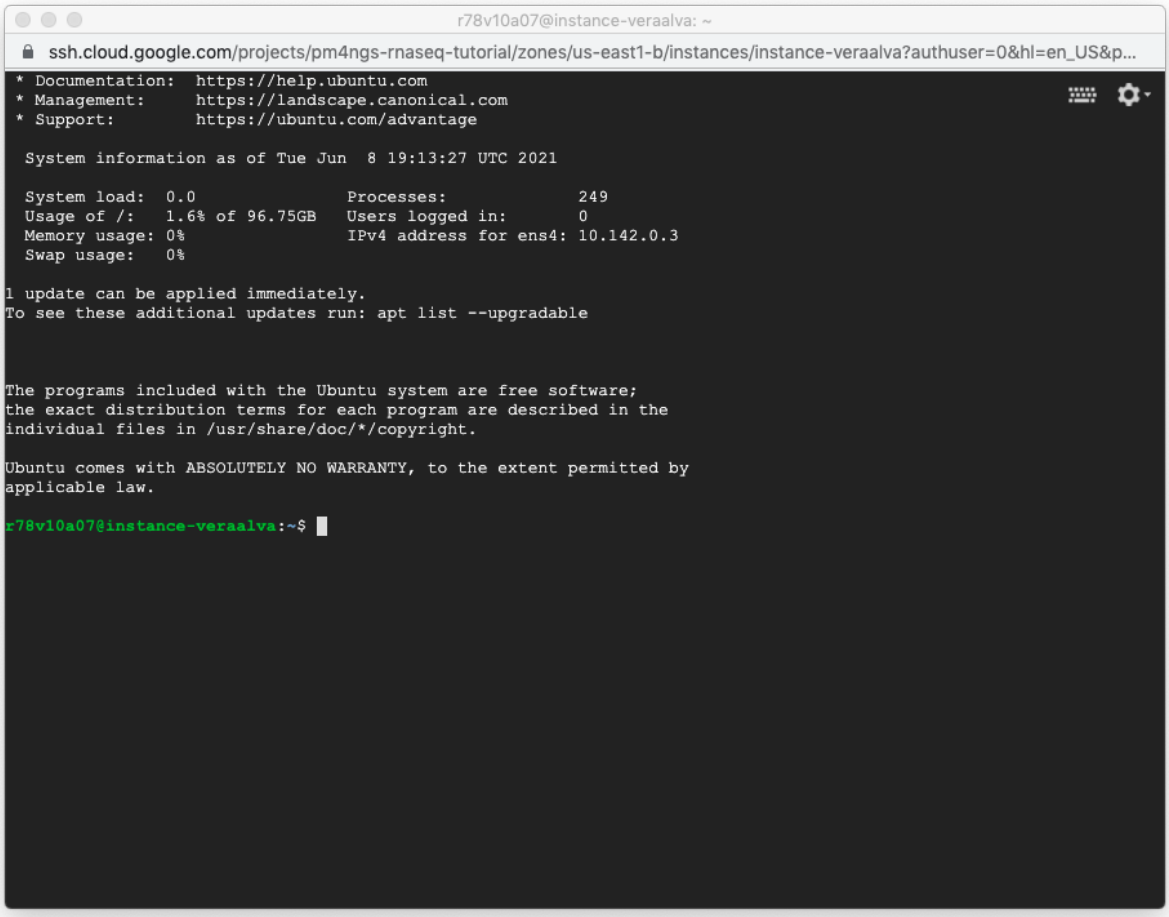
Create a GCP instance with 16 CPUs and 60GB of RAM (machine type: **n1-standard-16**). Click on **Change** button in the **Boot disk** box to select Operating system: Ubuntu, Version: Ubuntu 20.04 LTS. Click on **Management** to select **Preemptibility: ON**


Name
Name is permanent
instance-veraalva
Labels (Optional)
Add label
Region
Region is permanent
us-east1 (South Carolina)
Zone
Zone is permanent
us-east1-b
Machine configuration
Machine family
General-purpose
Compute-optimized
Memory-optimized
Machine types for common workloads, optimized for cost and flexibility
Series
N1
Powered by Intel Skylake CPU platform or one of its predecessors
Machine type
n1-standard-16 (16 vCPU, 60 GB memory)
vCPU
16
Memory
60 GB
GPUs
-
CPU platform and GPU
Confidential VM service
Enable the Confidential Computing service on this VM instance.
Container
Deploy a container image to this VM instance. Learn more
Boot disk
New 100 GB balanced persistent disk
Image
Ubuntu 20.04 LTS
Change
Identity and API access
Service account
No service account
Access scopes
Select a service account to enable API access
Firewall
Add tags and firewall rules to allow specific network traffic from the Internet
Allow HTTP traffic
Allow HTTPS traffic
Management, security, disks, networking, sole tenancy
The following options have been customized:
Preemptibility
Automatic restart
On host maintenance
You will be billed for this instance. Compute Engine pricing
Create
Cancel
Equivalent REST or command line
$126.80 monthly estimate
That's about $0.174 hourly
Pay for what you use: No upfront costs and per second billing
Item
Estimated costs
16 vCPUs + 60 GB memory
$116.80/month
100 GB balanced persistent disk
$10.00/month
Sustained use discount
- $0.00/month
Total
$126.80/month
Compute Engine pricing
Less

2.2 Accessing the instance with SSH

The instance will be available with two IP address, one internal for GCP and another external accessible from Internet. Click on the **SSH** button to login.

After clicking the **SSH** button, a new windows will be open. This is a Linux terminal running directly in the instance.



2.3 Installing dependencies on the GCP instance with Ubuntu

Runs these commands on a terminal to prepare the instance to run PM4NGS

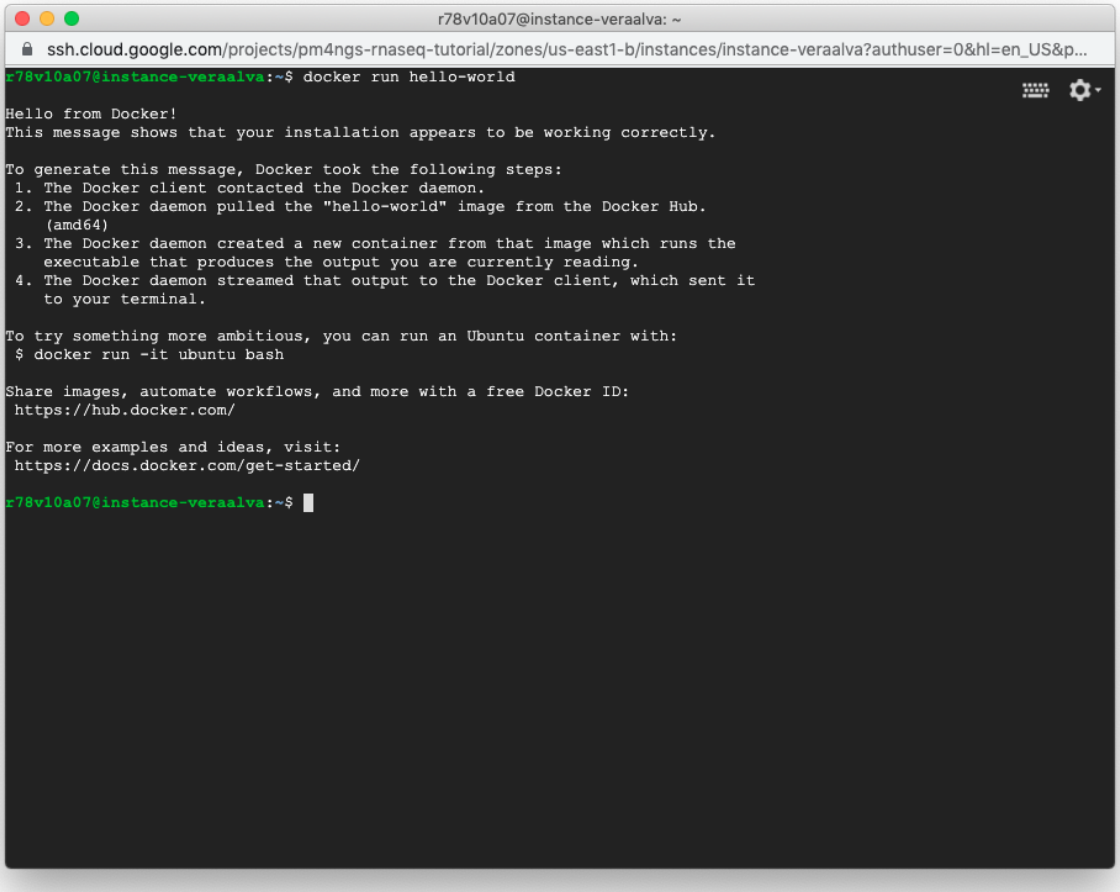
```
veraalva@instance-veraalva:~$ sudo apt-get update
veraalva@instance-veraalva:~$ sudo apt-get install docker.io python3 python3-pip python3-
↪venv python3-dev poppler-utils gcc nodejs tree
veraalva@instance-veraalva:~$ sudo usermod -aG docker $USER
veraalva@instance-veraalva:~$ logout
```

Close and reopen the terminal to set the docker group in the user. Then, click on the SSH button again to re-launch the terminal.

2.4 Testing the Docker daemon

```
veraalva@instance-veraalva:~$ docker run hello-world
```

Docker will pull the **hello-world** image and run it in a container. A **Hello from Docker!** message is displayed in the terminal.



CHAPTER

THREE

INSTALLING PM4NGS

Read PM4NGS published manuscript in GIGAScience

3.1 What is a Python virtual environment?

A python virtual environment is a "new Python installation" with their own site directories, optionally isolated from system site directories. Each virtual environment has its own Python binary (which matches the version of the binary that was used to create this environment) and can have its own independent set of installed Python packages in its site directories.

Creation of virtual environments is done by executing the command venv:

Note: python3 -m venv /path/to/new/virtual/environment

Running this command creates the target directory (creating any parent directories that don't exist already), a common name for the target directory uses as suffix *_venv*. It also creates a bin subdirectory containing a copy/symlink of the Python binary/binaries. It also creates an (initially empty) *lib/pythonX.Y/site-packages* subdirectory for the new packages. If an existing directory is specified, it will be re-used.

Once a virtual environment has been created, it can be **"activated"** using a script in the virtual environment's binary directory. The invocation of the script is platform-specific (<_venv> must be replaced by the path of the directory containing the virtual environment):

Note: $ source /path/to/new/virtual/environment/bin/activate

3.2 Installing PM4NGS

Creates a Python virtual environment named: **pm4ngs_venv** for installing PM4NGS.

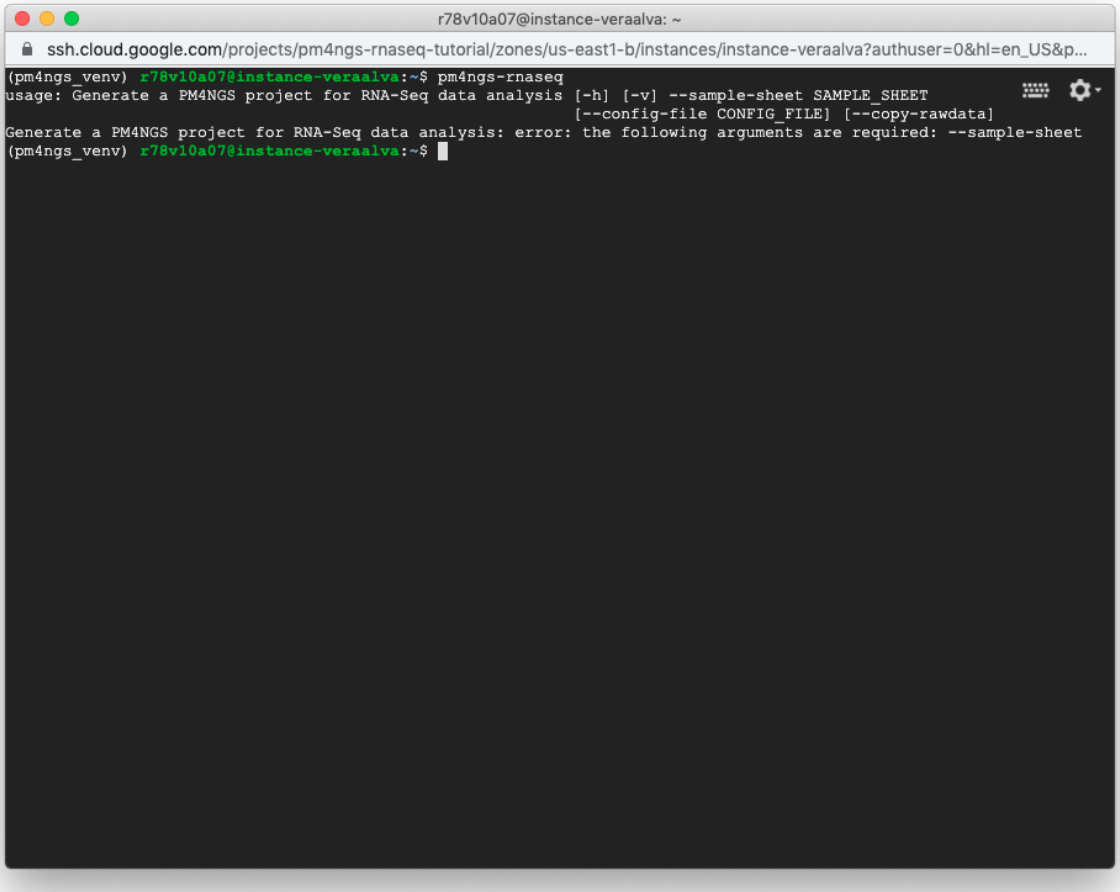
```
veraalva@instance-veraalva:~$ which python3
/usr/bin/python3
veraalva@instance-veraalva:~$ python3 -m venv pm4ngs_venv
veraalva@instance-veraalva:~$ source pm4ngs_venv/bin/activate
(pm4ngs_venv) veraalva@instance-veraalva:~$ which python3
/home/r78v10a07/pm4ngs_venv/bin/python3
(pm4ngs_venv) veraalva@instance-1:~$ pip install wheel
(pm4ngs_venv) veraalva@instance-1:~$ pip install pm4ngs
```

These commands need to be executed only one time during the tutorial.

3.3 Testing PM4NGS

Test PM4NGS RNA-Seq module:

```
(pm4ngs_venv) veraalva@instance-veraalva:~$ pm4ngs-rnaseq
```



CHAPTER

FOUR

THE JUPYTER SERVER

Jupyter is a free, open-source, interactive web tool known as a computational notebook, which researchers can use to combine software code, computational output, explanatory text and multimedia resources in a single document. Computational notebooks are essentially laboratory notebooks for scientific computing that keep track of all steps in a research project. Jupyter notebooks can be shared with collaborators and are used to guarantee reproducibility.

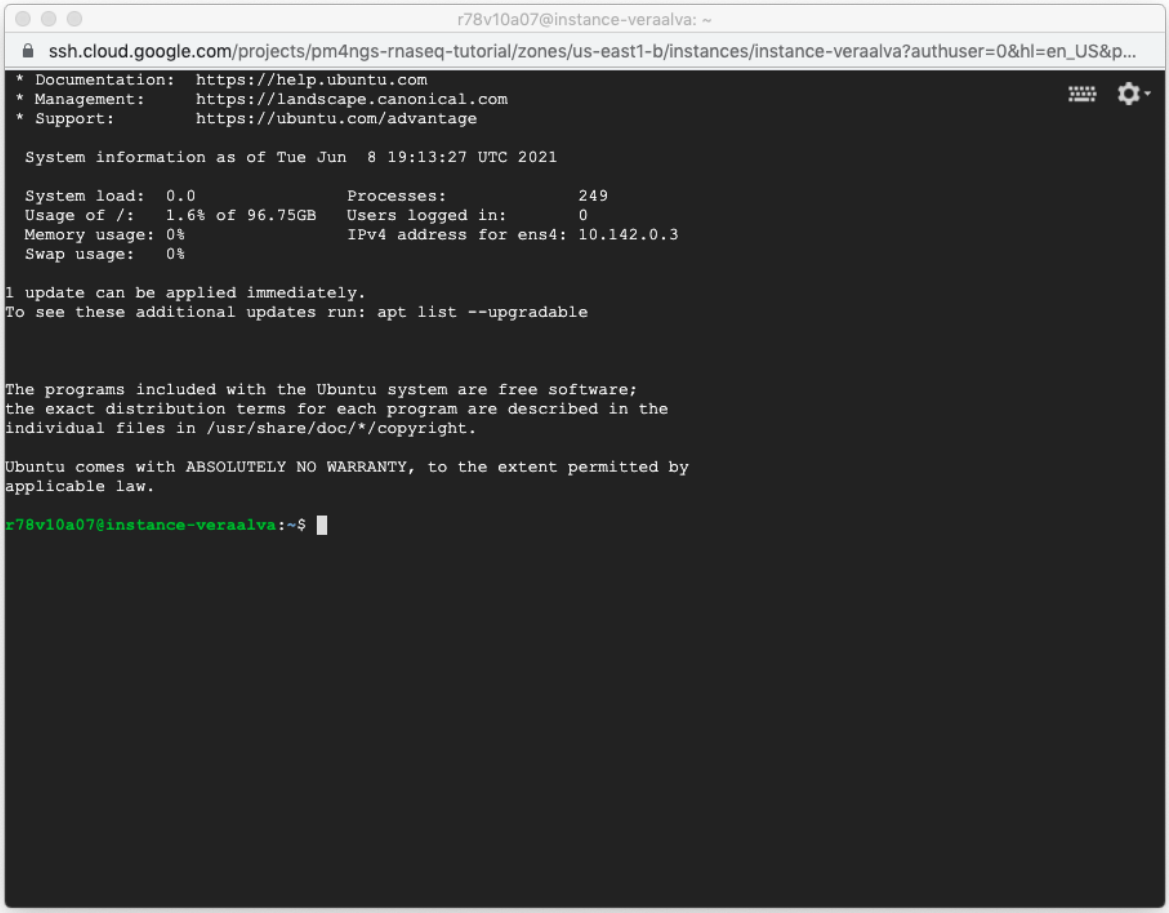
The Jupyter notebook has two components. Users input programming code or text in rectangular cells in a front-end web page. The browser then passes that code to a back-end 'kernel', which runs the code and returns the results. The kernel reside in the server, executing all code in that computer and the front-end in the user local browser.

Read this interesting article about Jupyter Notebooks for data scientists: https://www.nature.com/articles/d41586-018-07196-1

4.1 Accessing the instance with SSH

Click on the **SSH** button to login.

After clicking the **SSH** button, a new windows will be open. This is a Linux terminal running directly in the instance.



4.2 Activating pm4ngs_venv virtual environment

Everytime we login into the instance, we need to activate the **pm4ngs_venv** virtual environment. This mean that we will be using the python packages installed in this virtual environment instead of the packages in the instance.

```
veraalva@instance-veraalva:~$ source pm4ngs_venv/bin/activate
```

4.3 Running the Jupyter Server

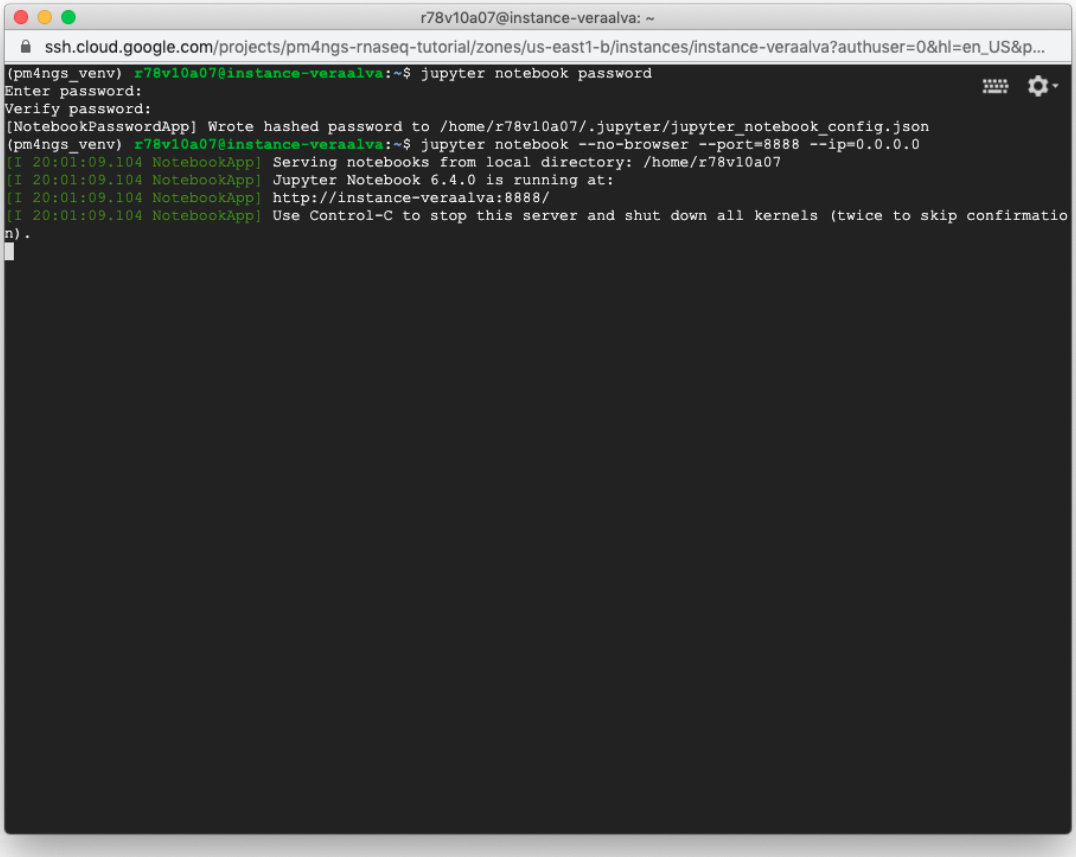
Open a terminal and activate the **pm4ngs_venv** virtual environment and run the jupyter server. As the GCP instance is a remote computer, we need to run the jupyter server with the **--port** and **--ip** options.

Creates a password for the Jupyter server:

```
(pm4ngs_venv) r78v10a07@instance-veraalva:~$ jupyter notebook password
```

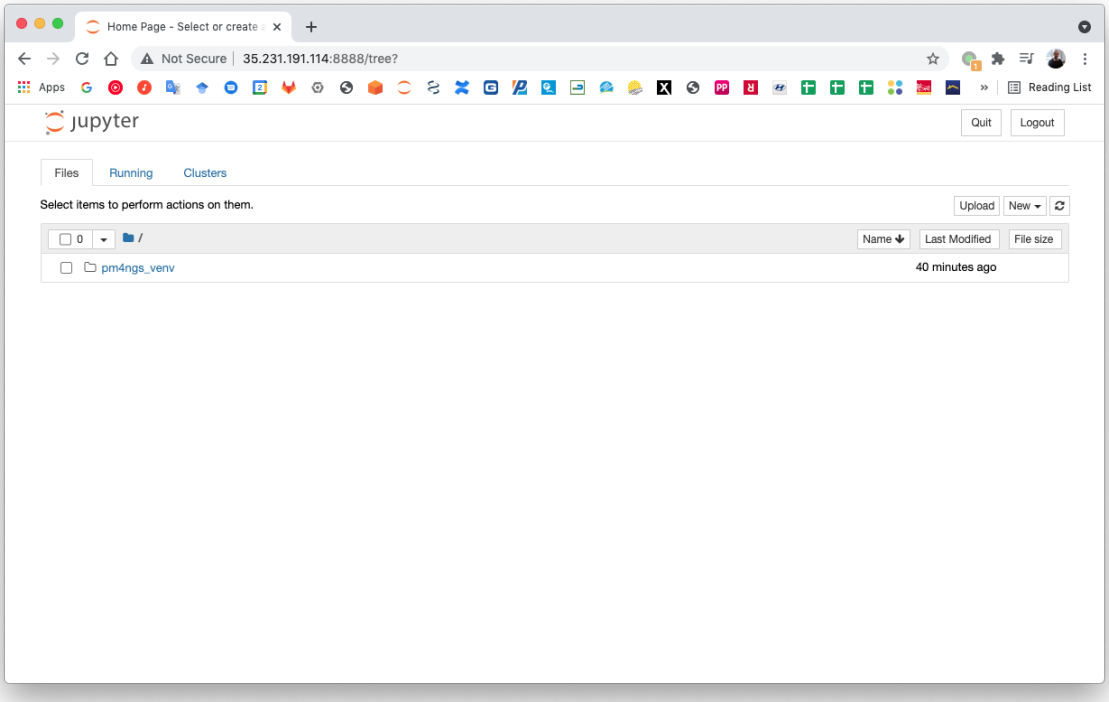
Start the Jupyter server

```
(pm4ngs_venv) r78v10a07@instance-veraalva:~$ jupyter notebook --no-browser --port=8888 --
↪ip=0.0.0.0
```



4.4 Open the jupyter web interface in your local computer

Get the **External IP** for your instance in the GCp Cloud console: **VM instances**. Then, type the address in your local browser plus the jupyter server port: **:8888**


Home Page - Select or create
Not Secure | 35.231.191.114:8888/tree?
Apps
Reading List
jupyter
Quit
Logout
Files
Running
Clusters
Select items to perform actions on them.
Upload
New
0
/
Name
Last Modified
File size
pm4ngs_venv
40 minutes ago

CHAPTER

FIVE

PM4NGS FOR RNA-SEQ DATA ANALYSIS

Differential expression (DE) analysis allows the comparison of RNA expression levels between multiple conditions.

The differential gene expression and GO enrichment pipeline is comprised of five steps, shown in next Table. The first step involves downloading samples from the NCBI SRA database, if necessary, or executing the pre-processing quality control tools on local samples. Subsequently, sample trimming, alignment, and quantification processes are executed. Once all samples are processed, groups of differentially expressed genes are identified per condition, using DESeq and EdgeR. Over- and under-expressed genes are reported by each program, and the interception of their results is computed.

Finally, once differentially expressed genes are identified, a GO enrichment analysis is executed to provide key biological processes, molecular functions, and cellular components for identified genes.

Table 1: DGA and Go enrichment, RNASeq pipeline

Step	Jupyter Notebook	Workflow CWL	Tool	Input	Output	CWL Tool
Sample Download and Quality Control	*Pre-processing QC*	download_quality_control.cwl	SRA-Tools	SRA accession	Fastq	fastq-dump.cwl
			FastQC	Fastq	FastQC HTML and Zip	fastqc.cwl
Trimming	*Samples trimming*		Trimmomatic	Fastq	Fastq	trimmomatic-PE.cwl trimmomatic-SE.cwl
Alignments and Quantification	*Alignment and Quantification*	rnaseq-alignment-quantification.cwl	STAR	Fastq	BAM	star.cwl
			Samtools	SAM	BAM Sorted BAM BAM index BAM stats BAM flagstats	samtools-view.cwl samtools-sort.cwl samtools-index.cwl samtools-stats.cwl samtools-flagstat.cwl
			IGVtools	Sorted BAM	TDF	igvtools-count.cw
			RSeQC	Sorted BAM	Alignment quality control (TXT and PDF)	rseqc-bam_stat.cwl rseqc-infer_experiment.cwl rseqc-junction_annotation.cwl rseqc-junction_saturation.cwl rseqc-read_distribution.cwl rseqc-read_quality.cwl
			TPMCalculator	Sorted BAM	Read counts (TSV) TPM values (TSV)	tpmcalculator.cwl
Differential Gene Analysis	*Differential Gene Expression Analysis*		DeSEq2	Read Matrix	TSV	deseq2-2conditions.cwl
			EdgeR	Read Matrix	TSV	edgeR-2conditions.cwl
Go enrichment	*Gene Ontology Enrichment*		goenrichment	gene IDs	TSV	Python code in the notebook

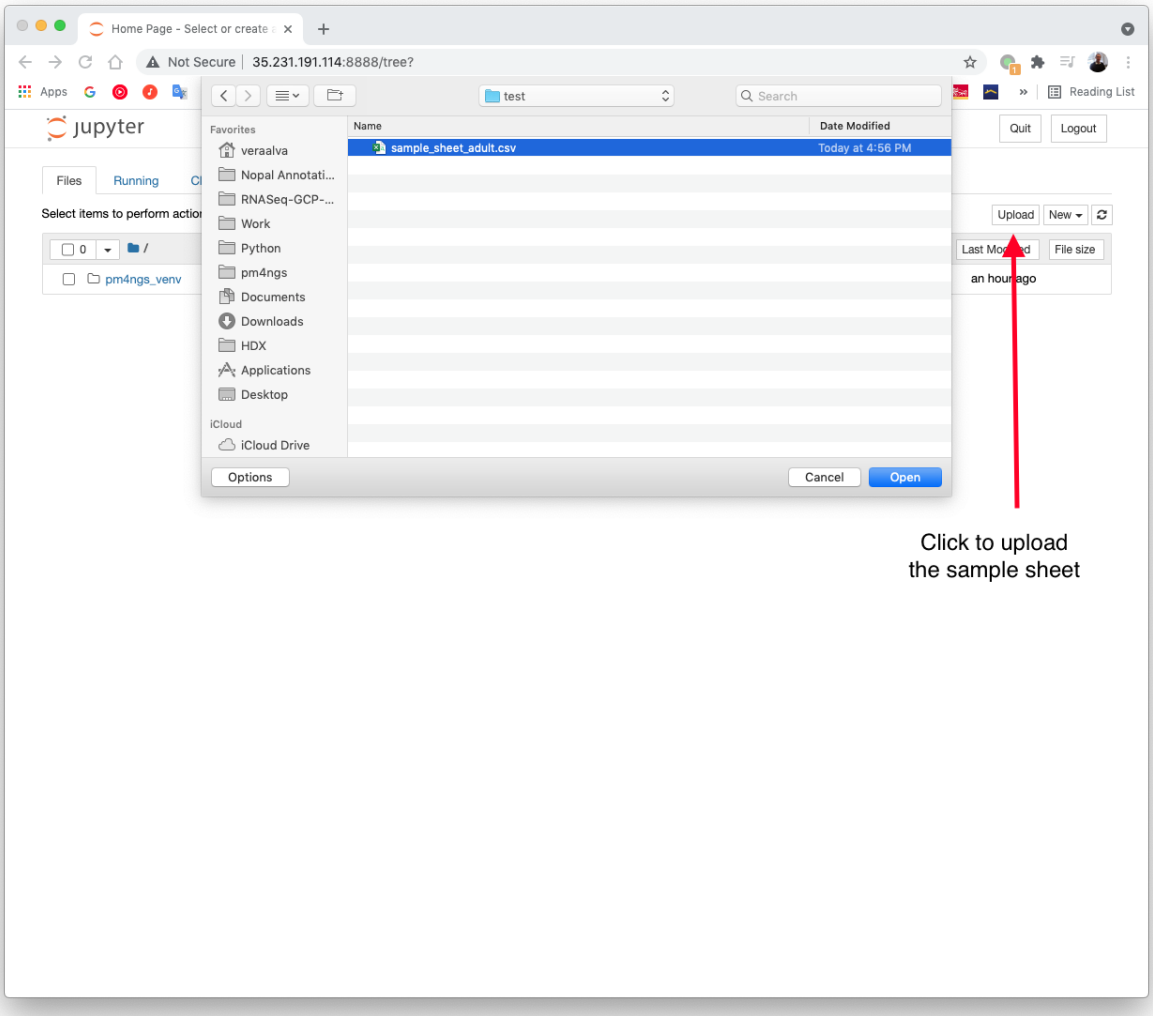
5.1 Sample sheet

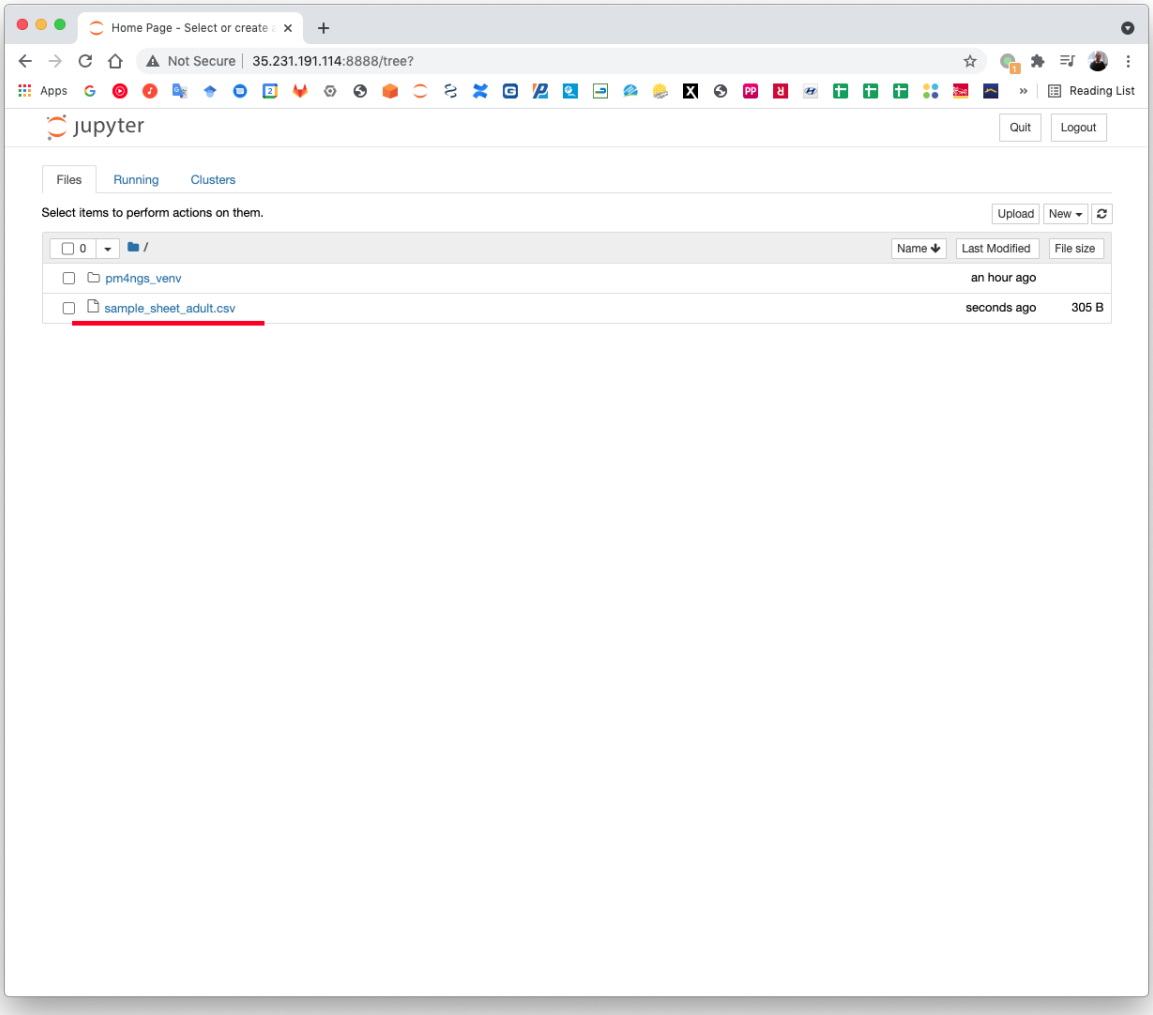
Sample Sheet for Drosophila BioProject PRJDB5673

sample_name	file	description	replicate	condition
DRR092341		Wild type, adult male head, replicate 1	1	WTAdult
DRR092342		Wild type, adult male head, replicate 2	2	WTAdult
DRR092343		Lol mutant, adult male head, replicate 1	1	LolAdult
DRR092344		Lol mutant, adult male head, replicate 2	2	LolAdult

Download the sample sheet file to your local computer: sample_sheet_adult.csv

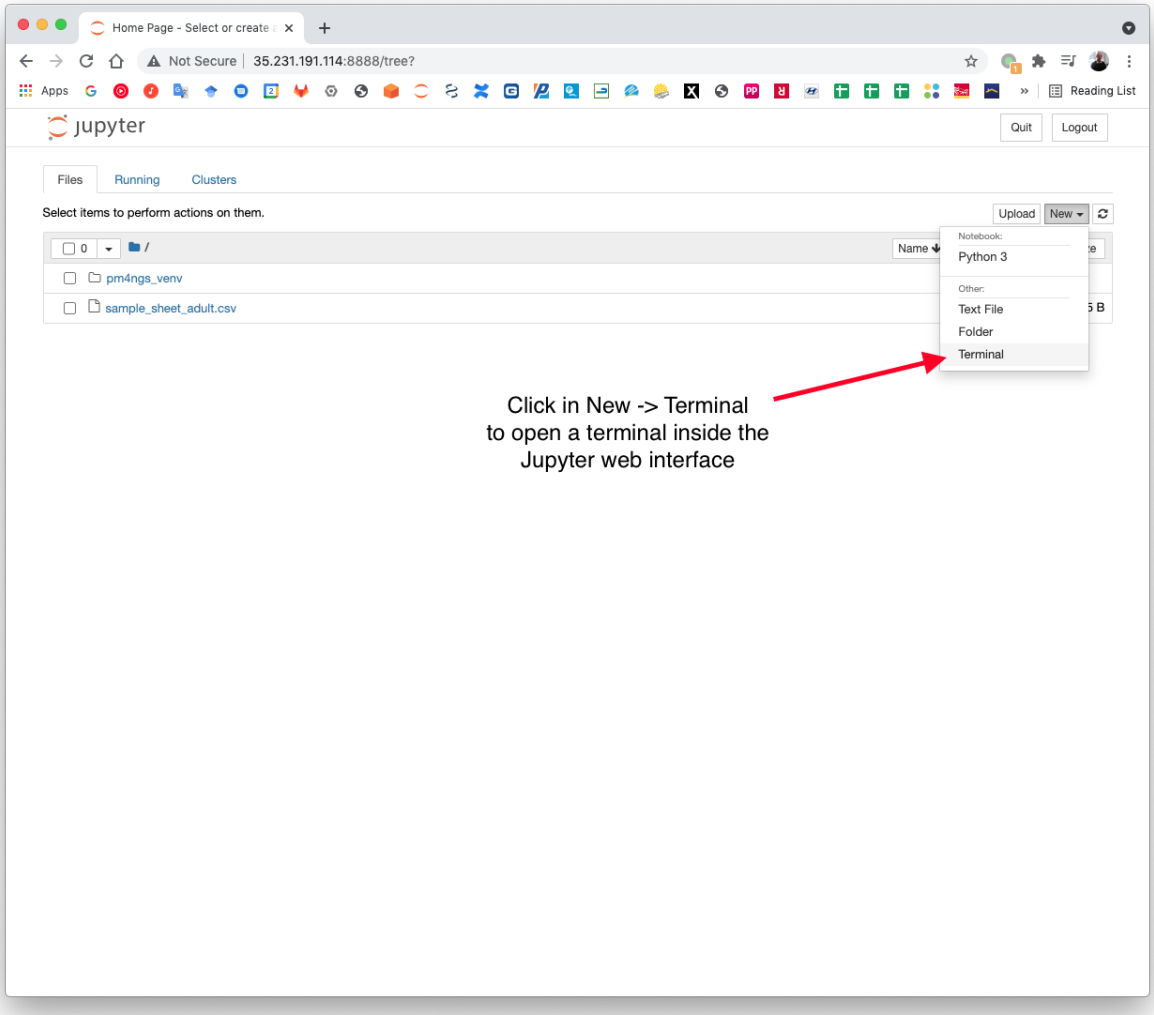
Upload the sample sheet file to the GCP instance through the Jupyter server





5.2 Creating the PM4NGS RNA-Seq project

Open a terminal to execute *pm4ngs-rnaseq*. This command will create an organizational data structure for the analysis.



In the terminal, activate the *pm4ngs_venv* virtual environment and execute the *pm4ngs-rnaseq* command.

The **pm4ngs-rnaseq** command line executed with the **--sample-sheet** option will let you type the different variables required for creating and configuring the project. The default value for each variable is shown in the brackets. After all questions are answered, the CWL workflow files will be cloned from the github repo ncbi/cwl-ngs-workflows-cbb to the folder **bin/cwl**.

```
r78v10a07@instance-veraalva:~$ source pm4ngs_venv/bin/activate
(pm4ngs_venv) r78v10a07@instance-veraalva:~$ ls
pm4ngs_venv  sample_sheet_adult.csv
(pm4ngs_venv) r78v10a07@instance-veraalva:~$ pm4ngs-rnaseq --sample-sheet sample_sheet_
↪adult.csv
```

Note:

- **author_name:** Use your full name
- **email:** Use your email
- **project_name:** Name of the project with no space nor especial characters. This will be used as project folder's

name.

Use: Dros_lol_mut

- **dataset_name:** Dataset to process name with no space nor especial characters. This will be used as folder name to group the data. This folder will be created under the **data/{{dataset_name}}** and **results/{{dataset_name}}**.

  Use: PRJDB5673

- **is_data_in_SRA:** If the data is in the SRA set this to y. A CWL workflow to download the data from the SRA database to the folder **data/{{dataset_name}}** and execute FastQC on it will be included in the **01 - Pre-processing QC.ipynb** notebook.

  Set this option to **n**, if the fastq files names and location are included in the sample sheet.

  Use: y

- **Select sequencing_technology:** Select one of the available sequencing technologies in your data.

  Values: 1 - single-end, 2 - paired-end

  Use: 1

- **create_demo:** If the data is downloaded from the SRA and this option is set to y, only the number of spots specified in the next variable will be downloaded. Useful to test the workflow.

  Use: n

- **number_spots:** Number of sport to download from the SRA database. It is ignored is the **create_demo** is set to **n**.

  Press Enter

- **organism:** Organism to process, e.g. human. This is used to link the selected genes to the NCBI gene database.

  Use: drosophila

- **genome_name:** Genome name , e.g. hg38 or mm10.

  Use: dm6

- **genome_dir:** Absolute path to the directory with the genome annotation (genome.fa, genes.gtf) to be used by the workflow or the name of the genome.

  If the name of the genome is used, PM4NGS will include a cell in the **03 - Alignments and Quantification.ipynb** notebook to download the genome files. The genome data will be at **data/{{dataset_name}}/{{genome_name}}/**

  Press Enter

- **aligner_index_dir:** Absolute path to the directory with the genome indexes for STAR.

  If **{{genome_name}}/STAR** is used, PM4NGS will include a cell in the **03 - Alignments and Quantification.ipynb** notebook to create the genome indexes for STAR.

  Press Enter

- **genome_fasta:** Absolute path to the genome fasta file

  If **{{genome_name}}/genome.fa** is used, PM4NGS will use the downloaded fasta file.

  Press Enter

- **genome_gtf:** Absolute path to the genome GTF file

  If **{{genome_name}}/genes.gtf** is used, PM4NGS will use the downloaded GTF file.

Press Enter

- **genome_bed:** Absolute path to the genome BED file

  If **{{genome_name}}/genes.bed** is used, PM4NGS will use the downloaded BED file.

  Press Enter

- **fold_change:** A real number used as fold change cutoff value for the DG analysis, e.g. 2.0.

  Press Enter

- **fdr:** Adjusted P-Value to be used as cutoff in the DG analysis, e.g. 0.05.

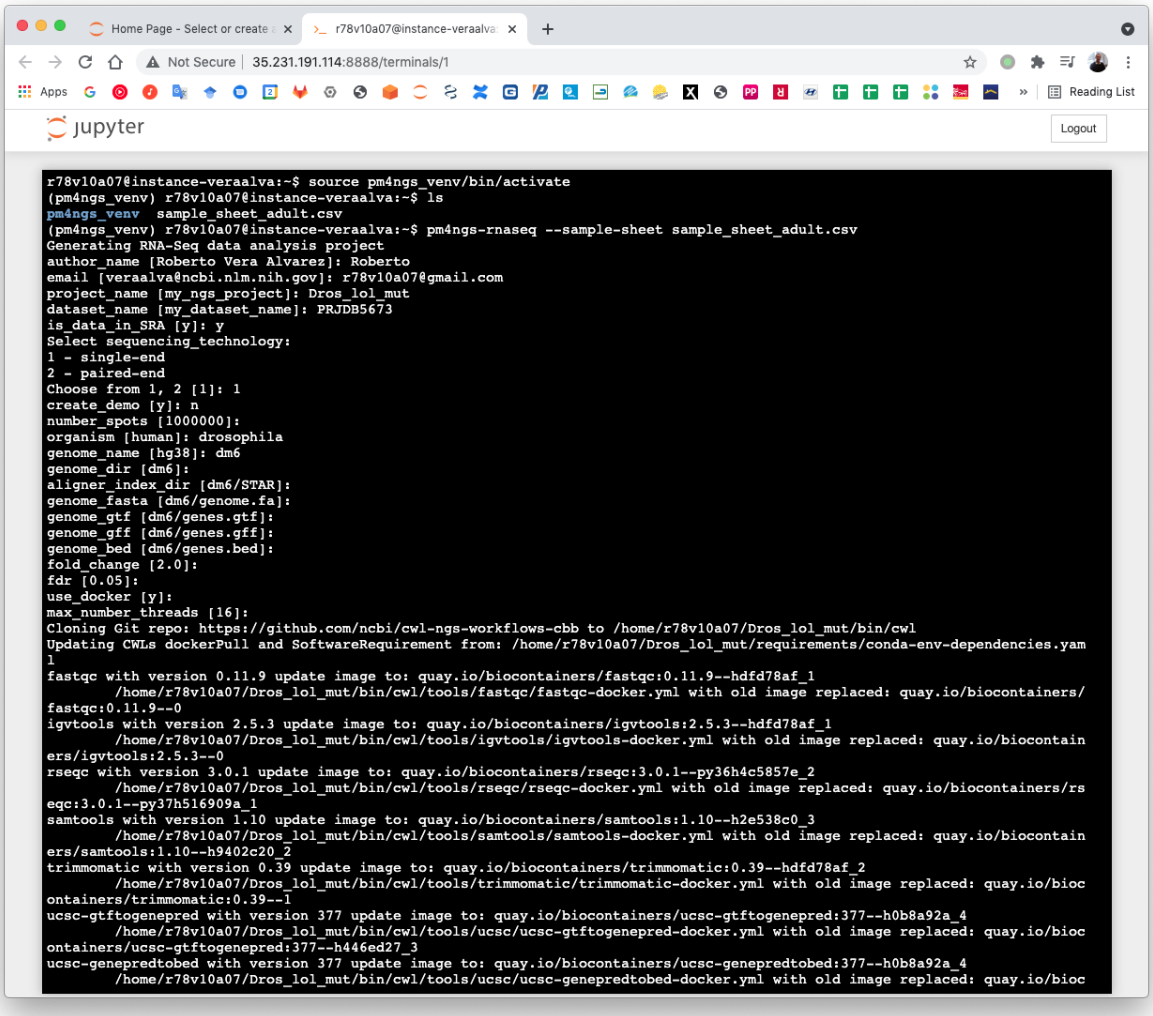
  Press Enter

- **use_docker:** Set this to y if you will be using Docker. Otherwise Conda needs to be installed in the computer.

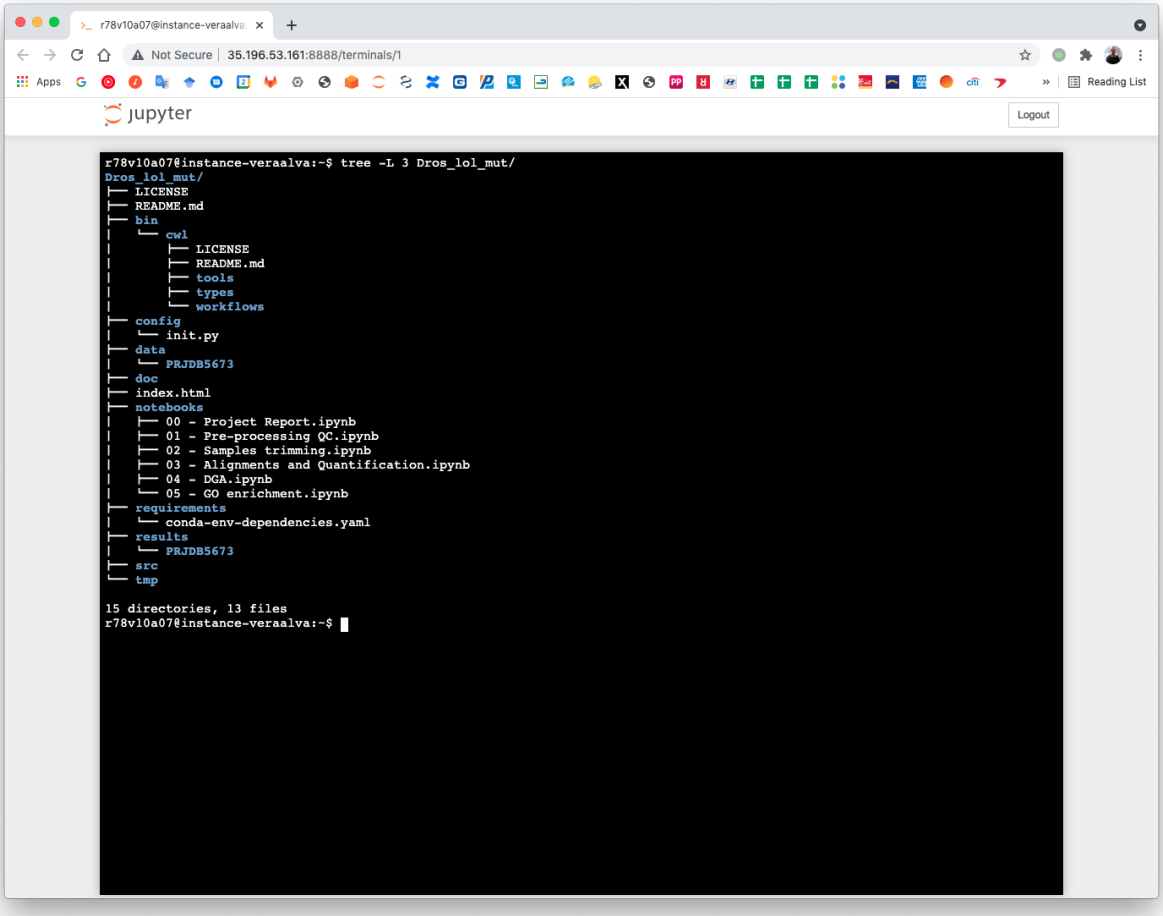
  Press Enter

- **max_number_threads:** Number of threads available in the computer.

  Press Enter

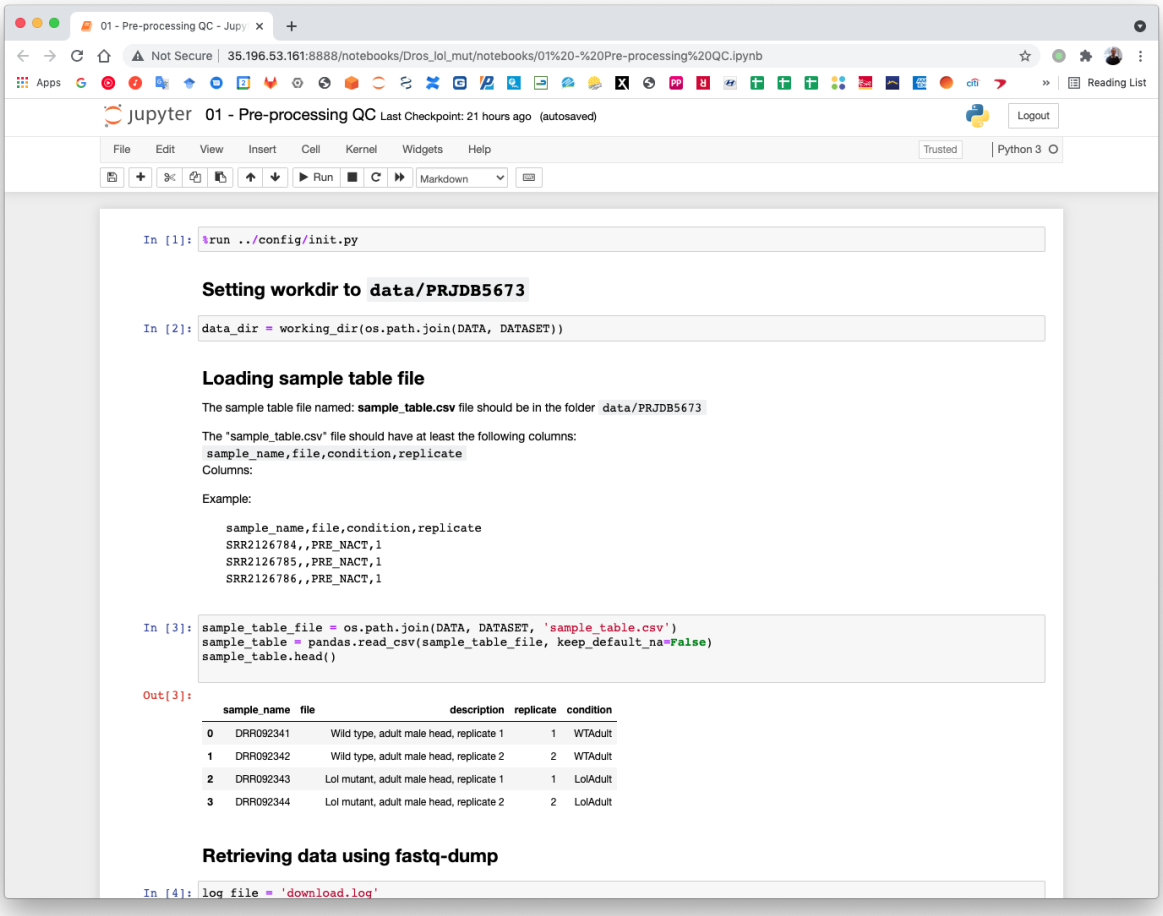


The project organizational data structure is:

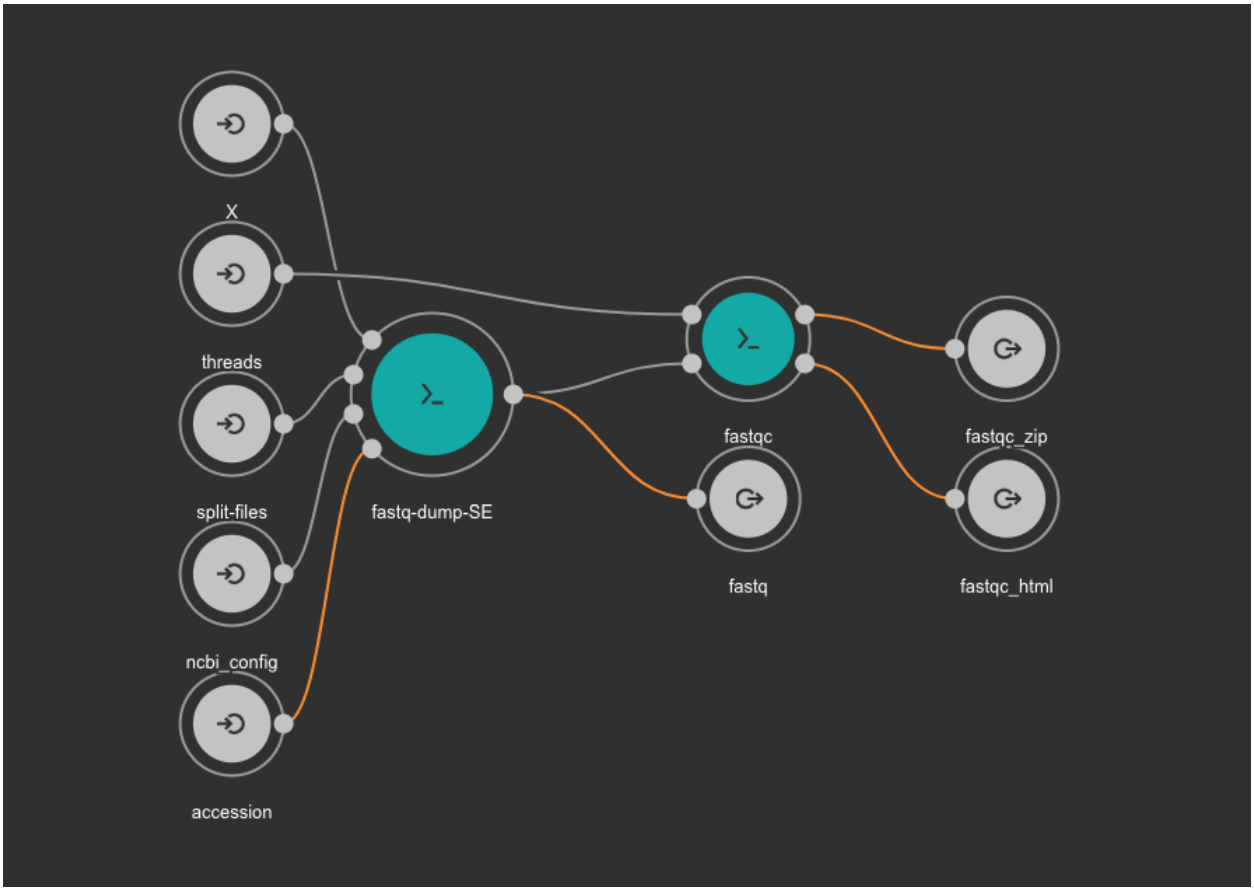


5.3 Pre-processing QC

The first notebook download the SRA data using the accessions defined in the sample sheet. Execute all cells until the **Retrieving data using fastq-dump**. This cell will submit the CWL workflow. Open a terminal to check that the **fastq-dump** command is working.

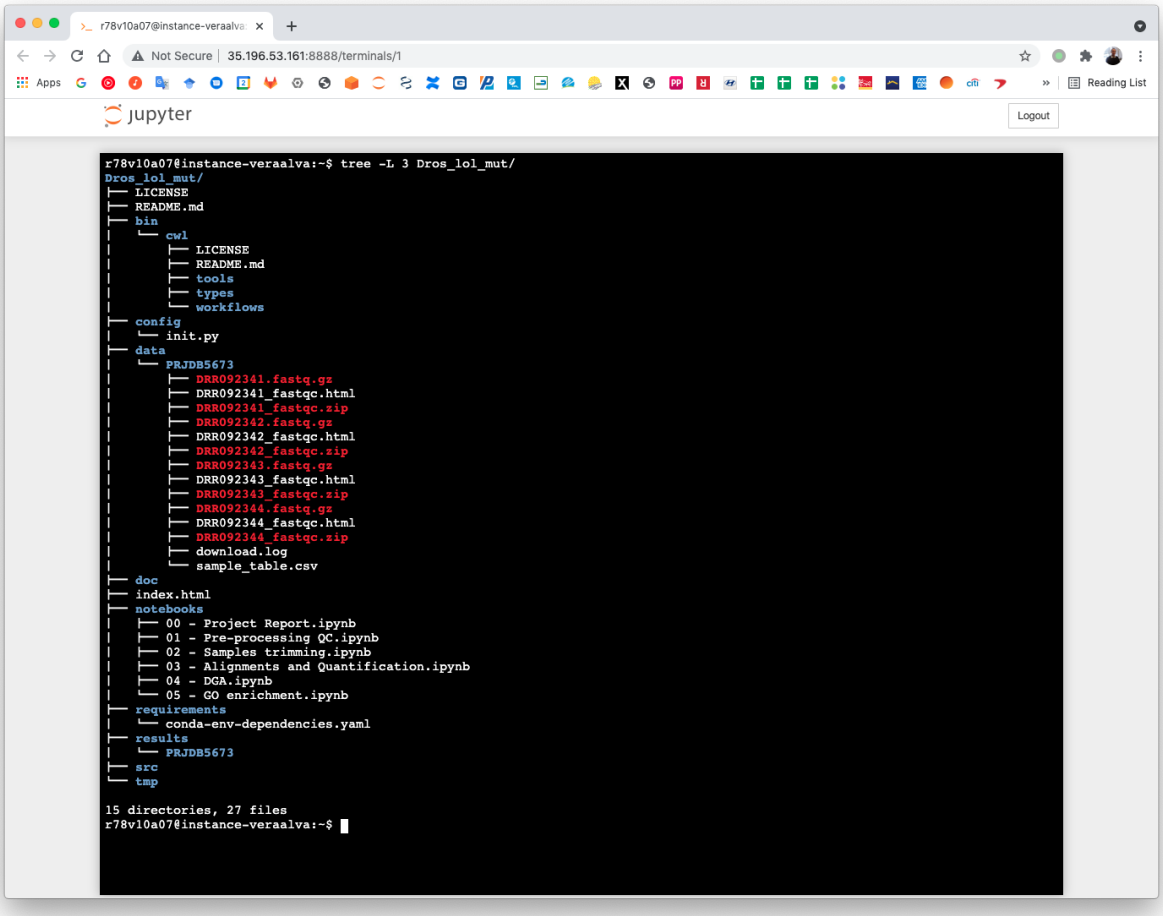


The CWL workflow for this step is: download_quality_control.cwl

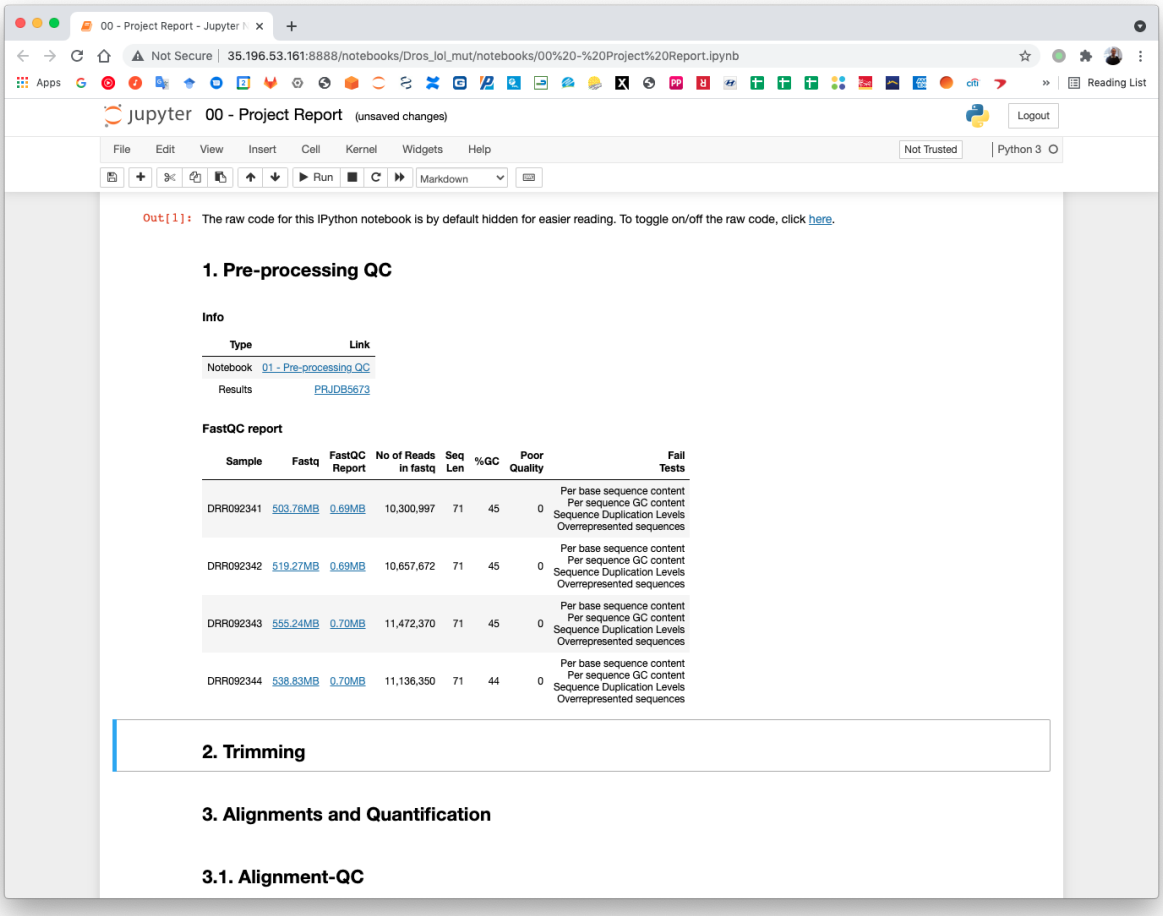


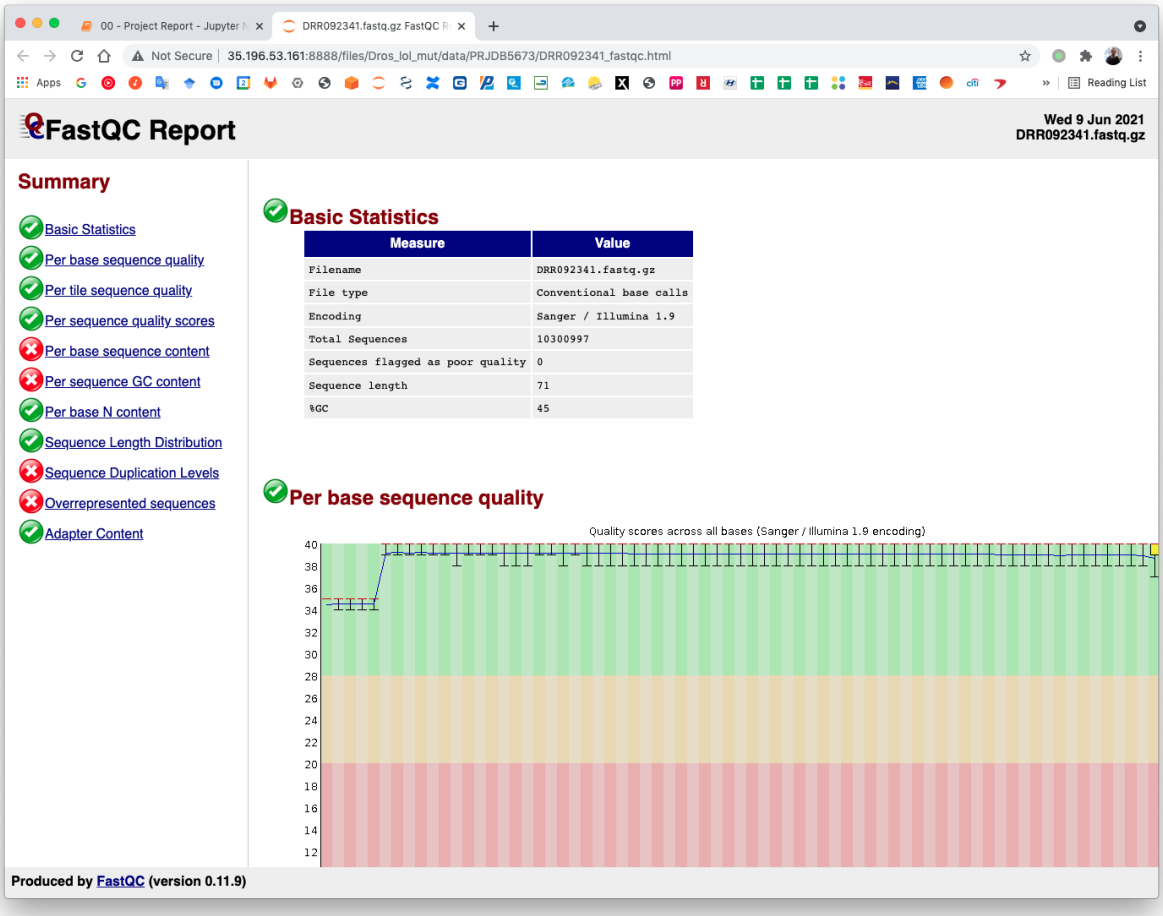
Once all cells are execute completely the *fastq* samples will be available at the **data/PRJDB5673** directory. Run the **tree** command to visualize the data structure.

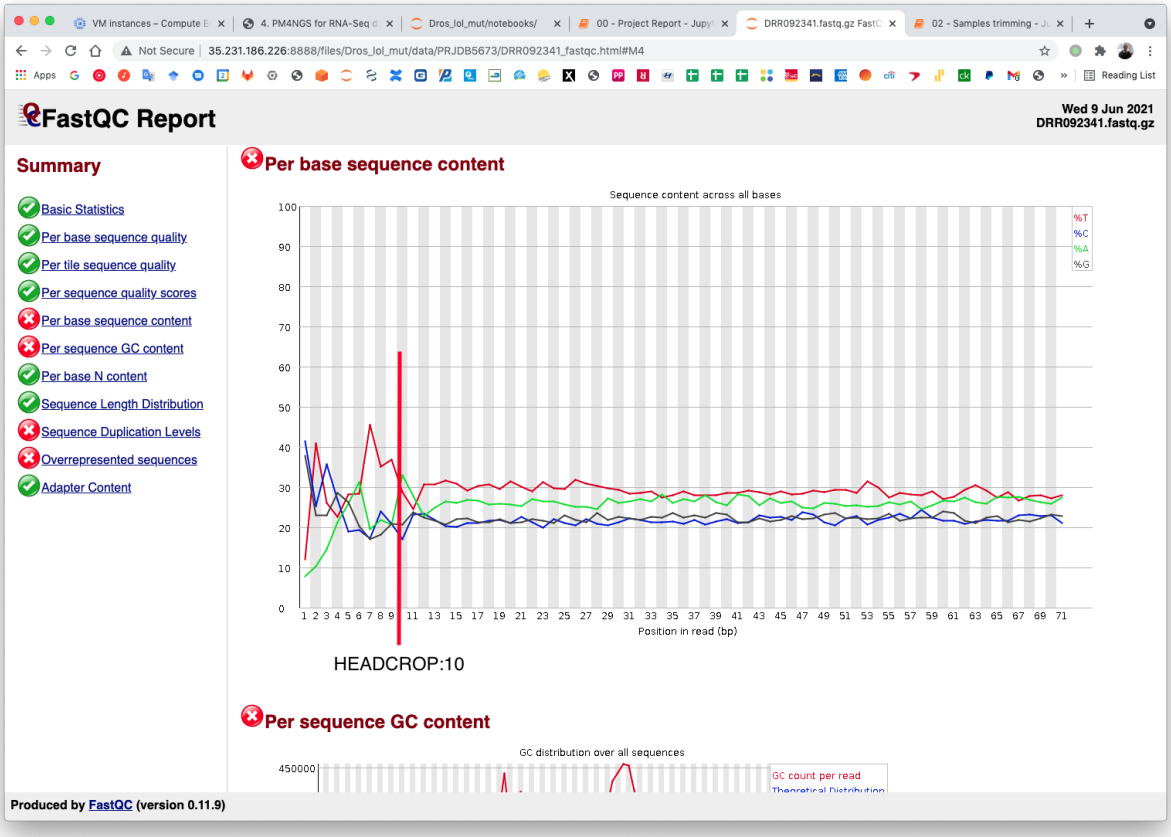
```
(pm4ngs_venv) r78v10a07@instance-veraalva:~$ tree -L 3 Dros_lol_mut/
```

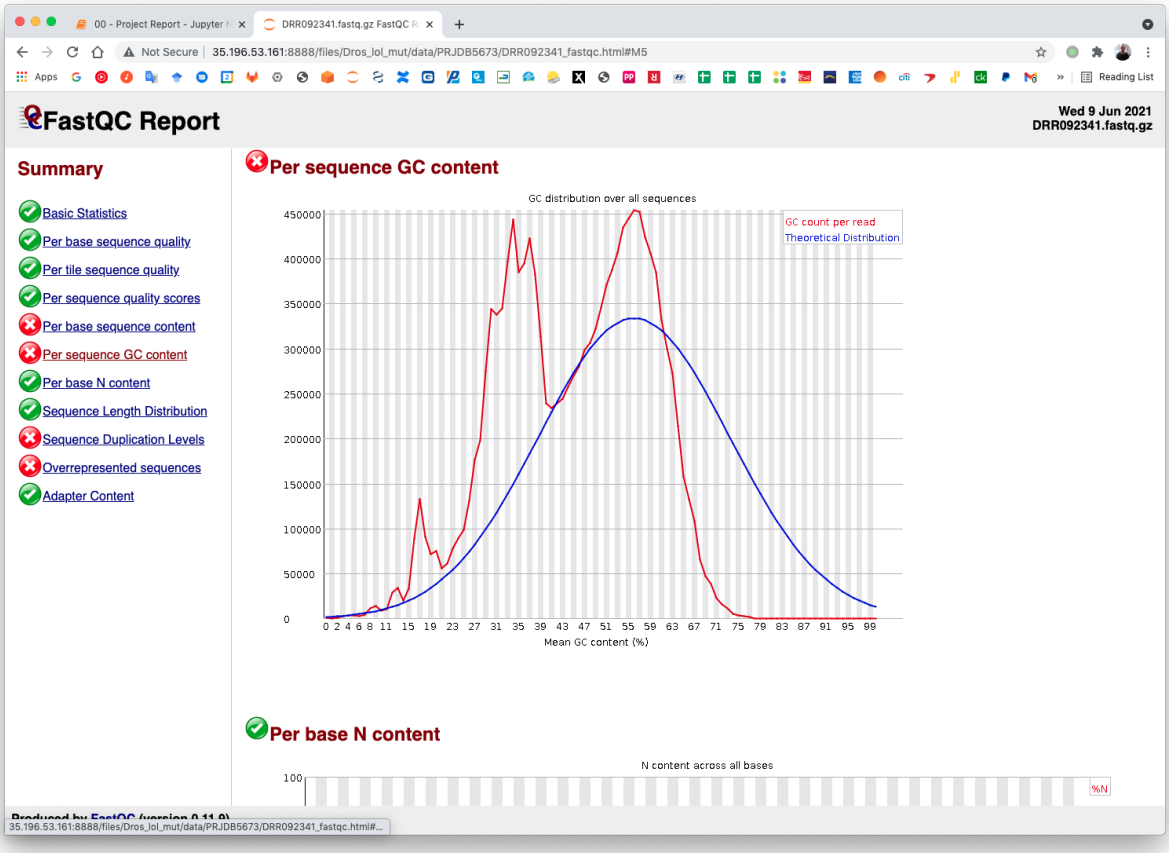


The pre-processing table is available in the *00 - Project Report* notebook. The links in the table gives access to the FastQC reports for each sample. The reports are used to select the trimming parameters.


00 - Project Report - Jupyter
Not Secure
35.196.53.161:8888/notebooks/Dros_lol_mut/notebooks/00%20-%20Project%20Report.ipynb
jupyter
00 - Project Report
(unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3
Run
Markdown
Out[1]:
The raw code for this IPython notebook is by default hidden for easier reading. To toggle on/off the raw code, click here.
1. Pre-processing QC
Info
Type
Link
Notebook
01 - Pre-processing QC
Results
PRJDB5673
FastQC report
Sample
Fastq
FastQC Report
No of Reads in fastq
Seq Len
%GC
Poor Quality
Fail Tests
DRR092341
503.76MB
0.69MB
10,300,997
71
45
0
Per base sequence content
Per sequence GC content
Sequence Duplication Levels
Overrepresented sequences
DRR092342
519.27MB
0.69MB
10,657,672
71
45
0
Per base sequence content
Per sequence GC content
Sequence Duplication Levels
Overrepresented sequences
DRR092343
555.24MB
0.70MB
11,472,370
71
45
0
Per base sequence content
Per sequence GC content
Sequence Duplication Levels
Overrepresented sequences
DRR092344
538.83MB
0.70MB
11,136,350
71
44
0
Per base sequence content
Per sequence GC content
Sequence Duplication Levels
Overrepresented sequences
2. Trimming
3. Alignments and Quantification
3.1. Alignment-QC


00 - Project Report - Jupyter
DRR092341.fastq.gz FastQC R
Not Secure
35.196.53.161:8888/files/Dros_lol_mut/data/PRJDB5673/DRR092341_fastqc.html
Reading List
FastQC Report
Wed 9 Jun 2021
DRR092341.fastq.gz
Summary
Basic Statistics
Per base sequence quality
Per tile sequence quality
Per sequence quality scores
Per base sequence content
Per sequence GC content
Per base N content
Sequence Length Distribution
Sequence Duplication Levels
Overrepresented sequences
Adapter Content
Basic Statistics
Measure
Value
Filename
DRR092341.fastq.gz
File type
Conventional base calls
Encoding
Sanger / Illumina 1.9
Total Sequences
10300997
Sequences flagged as poor quality
0
Sequence length
71
%GC
45
Per base sequence quality
Quality scores across all bases (Sanger / Illumina 1.9 encoding)
40
38
36
34
32
30
28
26
24
22
20
18
16
14
12
Produced by FastQC (version 0.11.9)


FastQC Report
Wed 9 Jun 2021
DRR092341.fastq.gz
Summary
Basic Statistics
Per base sequence quality
Per tile sequence quality
Per sequence quality scores
Per base sequence content
Per sequence GC content
Per base N content
Sequence Length Distribution
Sequence Duplication Levels
Overrepresented sequences
Adapter Content
Per base sequence content
Sequence content across all bases
%T
%C
%A
%G
Position in read (bp)
HEADCROP:10
Per sequence GC content
GC distribution over all sequences
GC count per read
Produced by FastQC (version 0.11.9)



5.4 Samples trimming

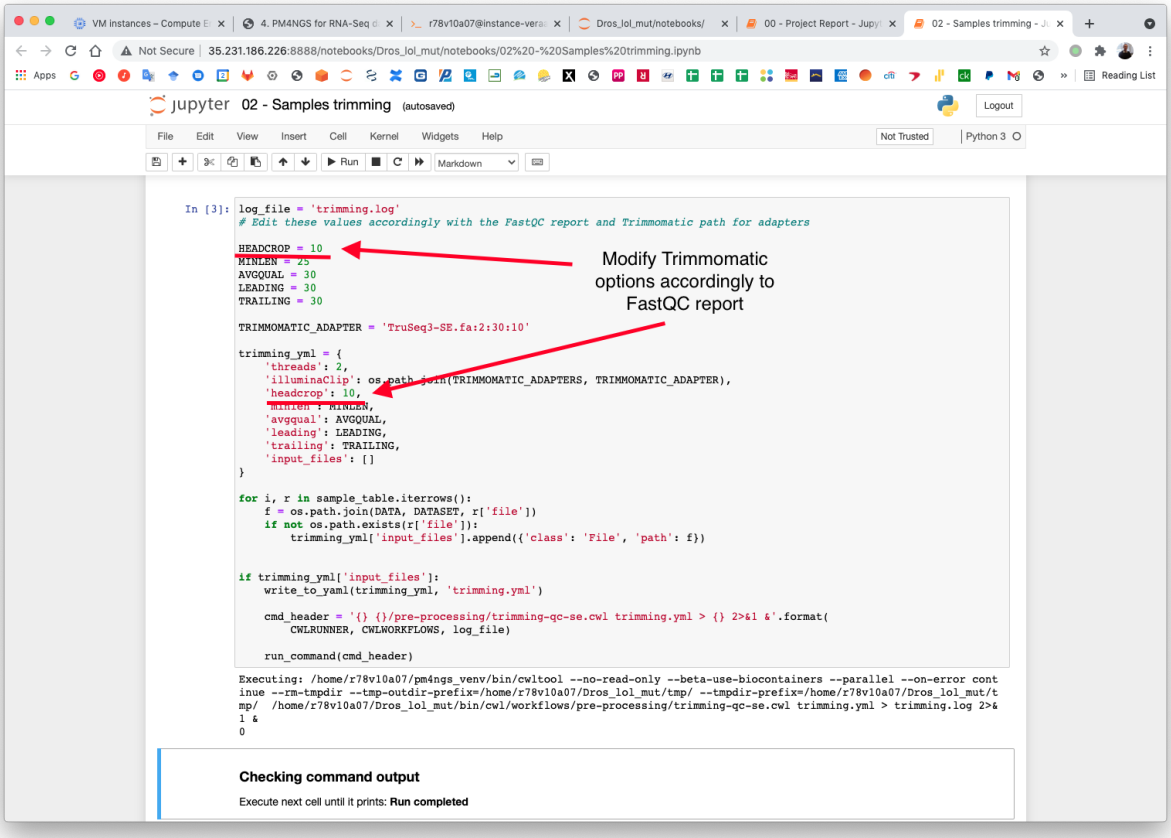
Trimmomatic performs a variety of useful trimming tasks for illumina paired-end and single ended data.The selection of trimming steps and their associated parameters are supplied on the command line.

The current trimming steps are:

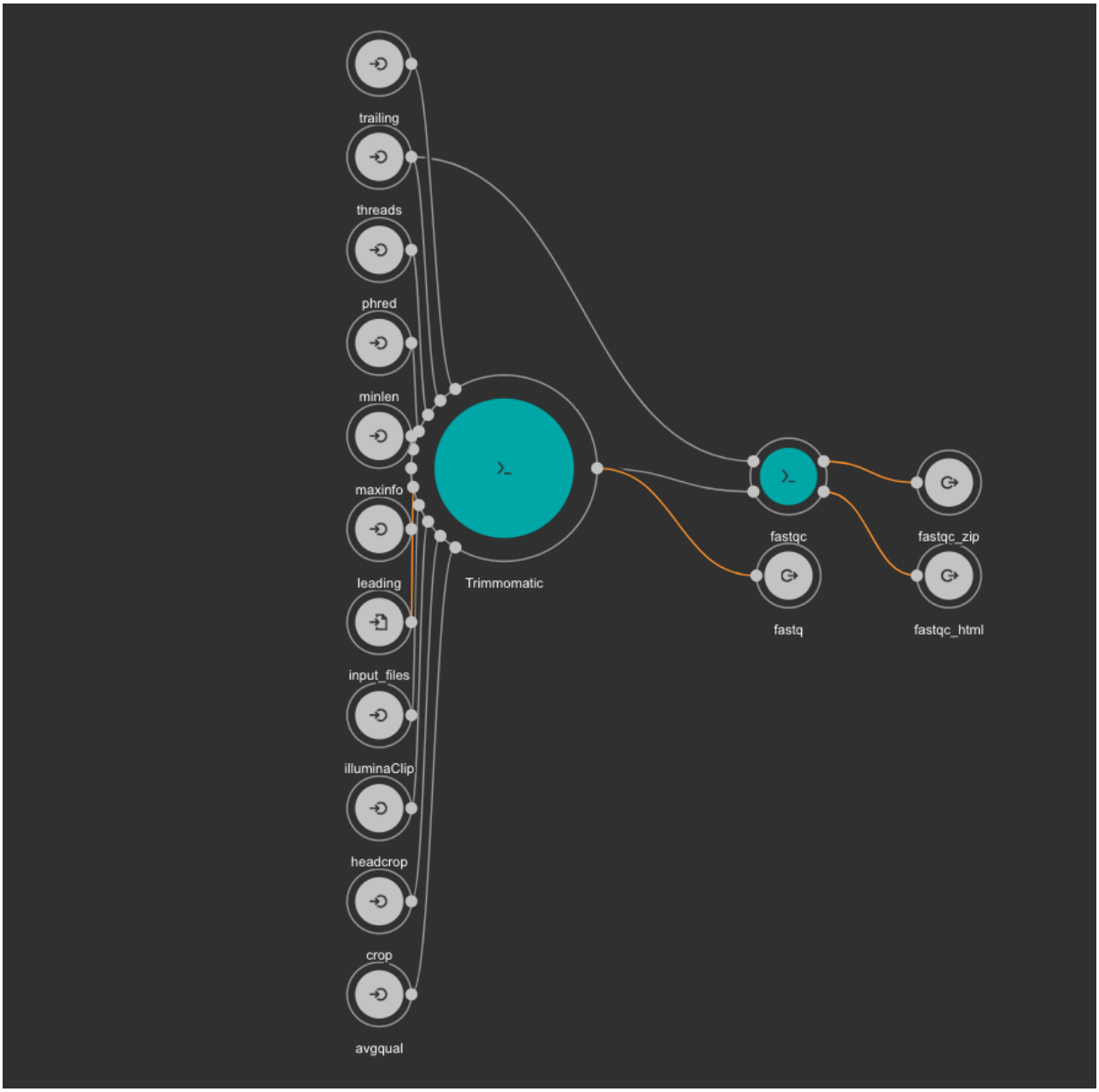
- ILLUMINACLIP: Cut adapter and other illumina-specific sequences from the read.
- SLIDINGWINDOW: Perform a sliding window trimming, cutting once the average quality within the window falls below a threshold.
- LEADING: Cut bases off the start of a read, if below a threshold quality
- TRAILING: Cut bases off the end of a read, if below a threshold quality
- CROP: Cut the read to a specified length
- HEADCROP: Cut the specified number of bases from the start of the read
- MINLEN: Drop the read if it is below a specified length
- TOPHRED33: Convert quality scores to Phred-33
- TOPHRED64: Convert quality scores to Phred-64

It works with FASTQ (using phred + 33 or phred + 64 quality scores, depending on the Illumina pipeline used), either uncompressed or gzipp'ed FASTQ. Use of gzip format is determined based on the .gz extension.

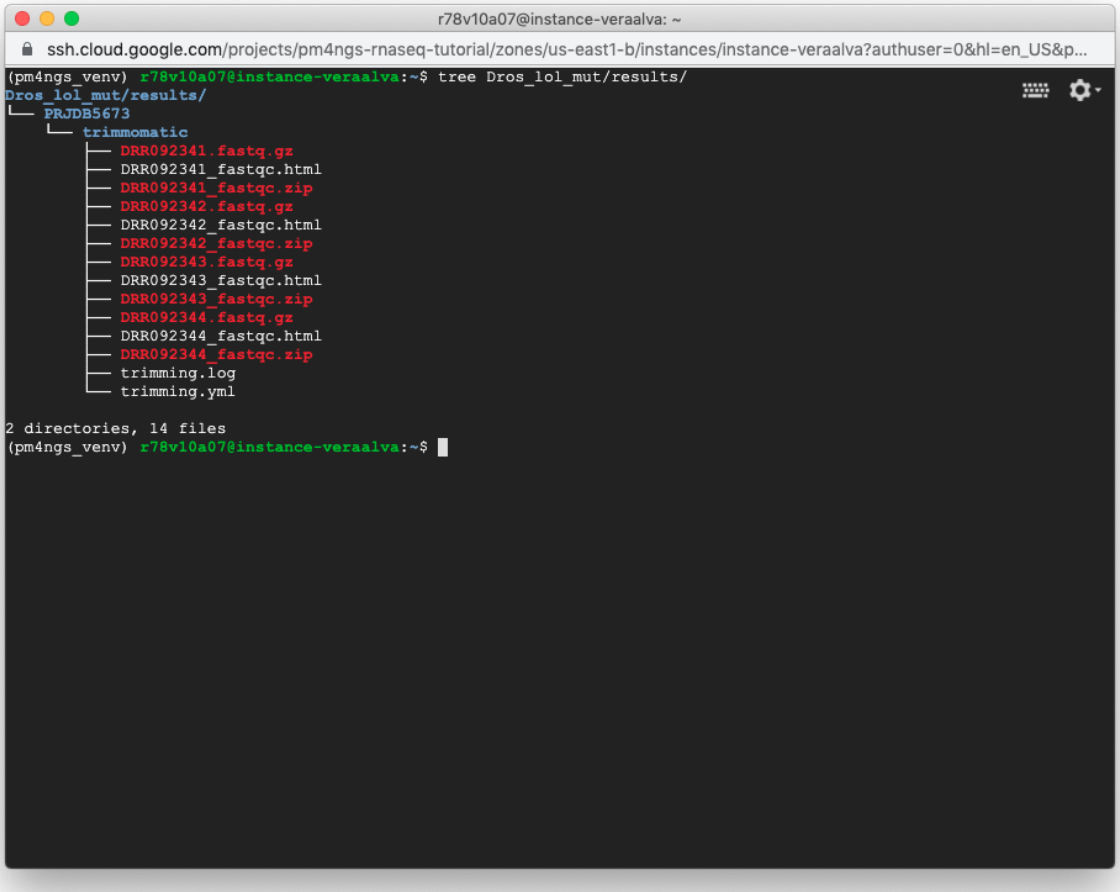
PM4NGS uses standard options for Trimmomatic but in this example we need to add **HEADCROP:10** as it is shown in next figure:



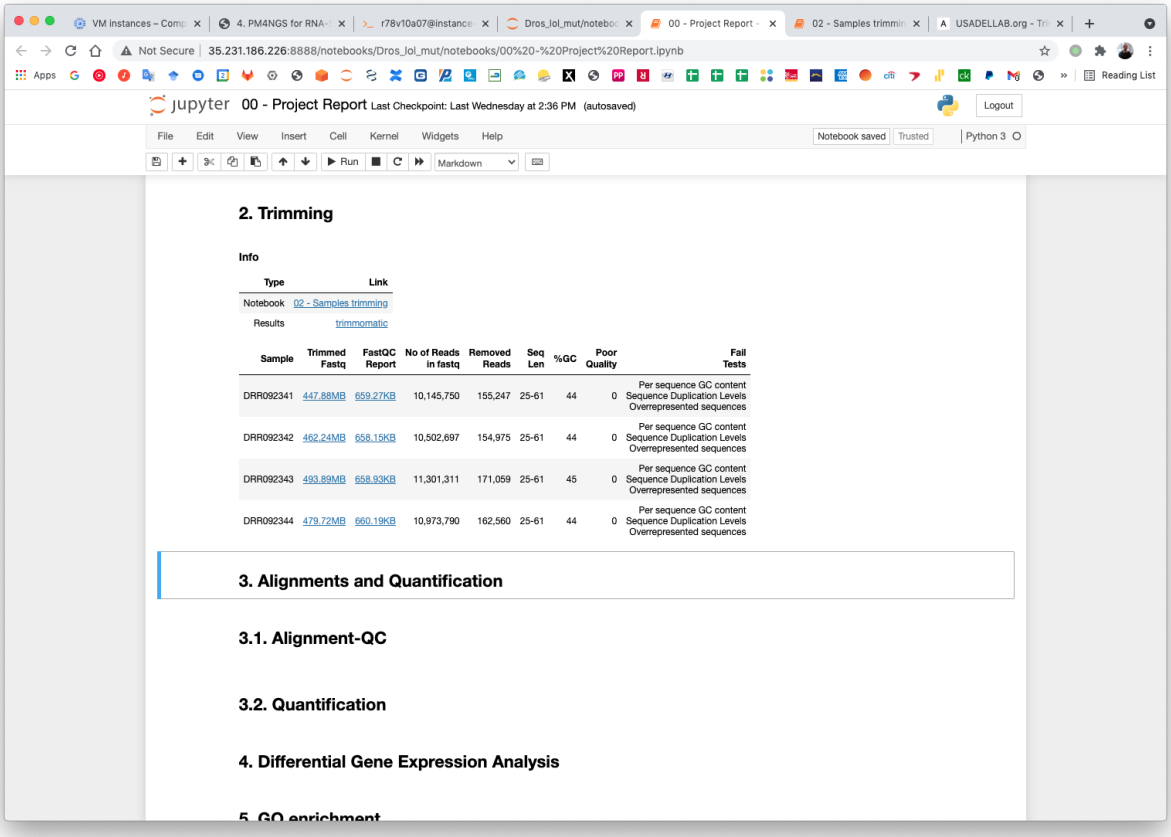
The CWL workflow for this step is: trimming-qc-se.cwl



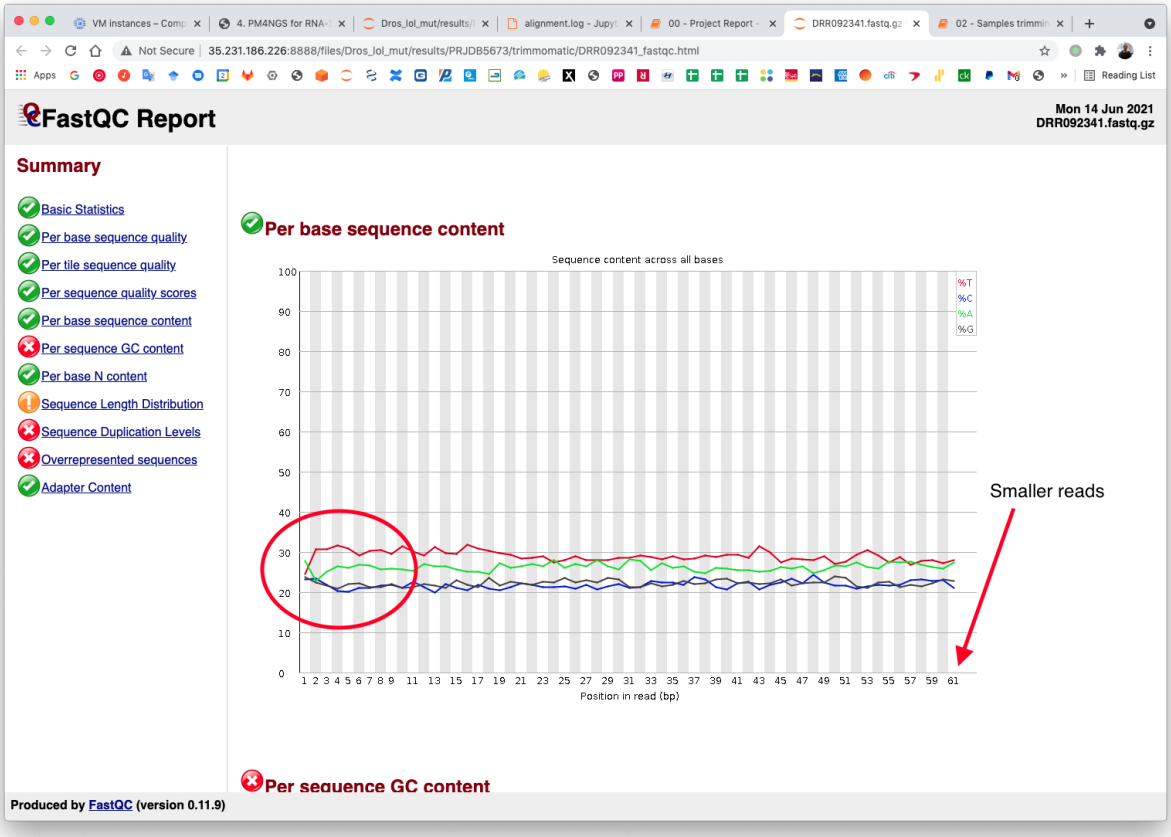
The result files for the trimming workflow are available at: **results/PRJDB5673/trimmomatic**



Updating the *00 - Project Report* notebook.



Check the FastQC reports to check if the trimming reduced the distortion in the first 10 bases.



5.5 Alignment and Quantification

The alignment and quantification workflow uses STAR as aligner, Samtools for filtering and sorting BAM files, TPM-Calculator for RNA-Seq quantification, RSeQC for post processing quality control and IGVtools for creating visualization files.

In this tutorial we are analysing *Drosophila* samples. Therefore, we need the *Drosophila* genome sequence and annotations. PM4NGS provides the **dm6** genome pre-formatted for the alignment and quantification workflow. For a complete list of PM4NGS pre-formatted genomes see: https://pm4ngs.readthedocs.io/en/latest/pipelines/genomes.html

Creating Genome indexes if they don't exists

In [3]:
```
if not os.path.exists(GENOME):
    working_dir(GENOME)
    !curl -o {GENOME_NAME}.tar.gz https://ftp.ncbi.nlm.nih.gov/pub/pm4ngs/resources/{GENOME_NAME}.tar.gz
    !tar xzfv {GENOME_NAME}.tar.gz --strip 1
    !rm {GENOME_NAME}.tar.gz
    working_dir(ALIGNER_INDEX)
    log_file = 'star_index.log'
    index_cmd = '{0} {1}/star/star-index.cwl --runThreadN {2} --genomeDir . '.format(CWLRUNNER, CWLTOOLS, 16)
    index_cmd += '--genomeFastaFiles ../genome.fa  --sjdbGTFfile ../genes.gtf > index.log 2>&1 &'
    run_command(index_cmd)
```

```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 47.8M  100 47.8M    0     0  44.9M      0  0:00:01  0:00:01 --:--:-- 44.9M
dm6/dm6-blacklist.bed
dm6/dm6.chrom.sizes
dm6/genes.bed
dm6/genes.genePred
dm6/genes.gtf
dm6/genome.fa
dm6/refGene.txt
Executing: /home/r78v10a07/pm4ngs_venv/bin/cwltool --no-read-only --beta-use-biocontainers --parallel --on-error cont
inue --rm-tmpdir --tmp-outdir-prefix=/home/r78v10a07/Dros_lol_mut/tmp/ --tmpdir-prefix=/home/r78v10a07/Dros_lol_mut/t
mp/  /home/r78v10a07/Dros_lol_mut/bin/cwl/tools/star/star-index.cwl --runThreadN 16 --genomeDir . --genomeFastaFiles
../genome.fa  --sjdbGTFfile ../genes.gtf > index.log 2>&1 &
0
```

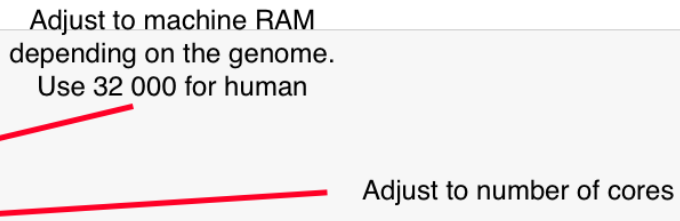
In [5]:
```
if os.path.exists('index.log'):
    check_cwl_command_log('index.log')
```

```
Process completed.
Proceed to next cells
```

This workflow is the most time consuming part and require setting proper computer resources like number of cores and RAM. For this tutorial we need at least 64 GB of RAM and 16 cores. GCP machine type *n1-standard-16* provides those resources.

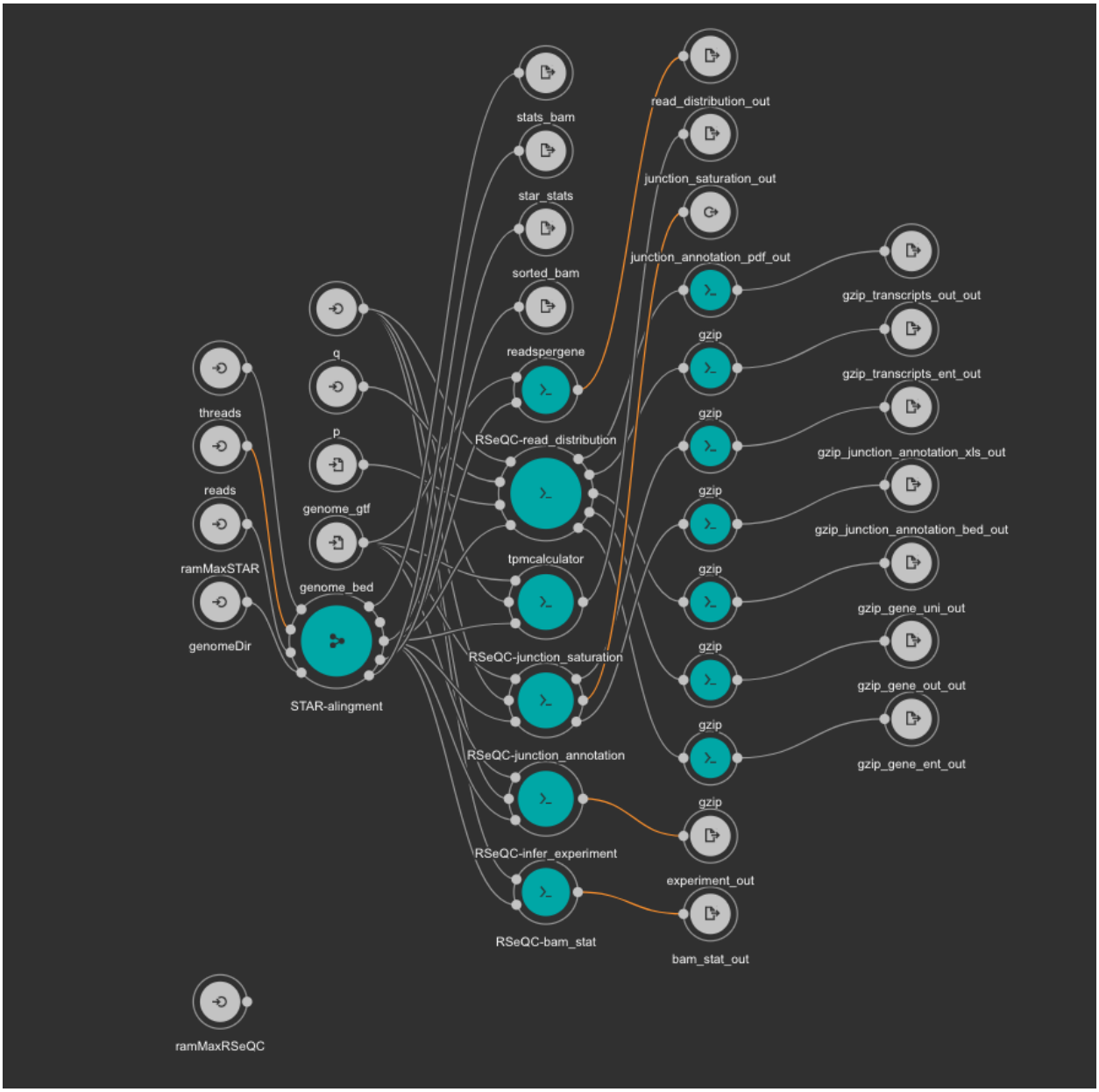
Processing samples

In [6]:
```
working_dir(result_dir)
log_file = 'alignment.log'

alignment_yml = {

    'ramMaxRSeQC': 16000,
    'ramMaxSTAR': 16000,


    'threads': 16,
    'genomeDir': {'class': 'Directory', 'path': ALIGNER_INDEX },
    'genome_bed': {'class': 'File', 'path': GENOME_BED },
    'genome_gtf': {'class': 'File', 'path': GENOME_GTF },
    'q': 255,
    'reads': []
}


for i, r in sample_table.iterrows():
    f = os.path.join(data_dir, r['file'])
    if not os.path.exists(r['file'].replace('.fastq.gz', '_genes.out.gz')):
        alignment_yml['reads'].append([
            {'class': 'File', 'path': f}])


if alignment_yml['reads']:
    write_to_yaml(alignment_yml, 'alignment.yml')
    cmd_header = '{} {}/RNA-Seq/rnaseq-alignment-quantification.cwl alignment.yml > {} 2>&1 &'.format(
        CWLRUNNER, CWLWORKFLOWS, log_file)
    run_command(cmd_header)
```

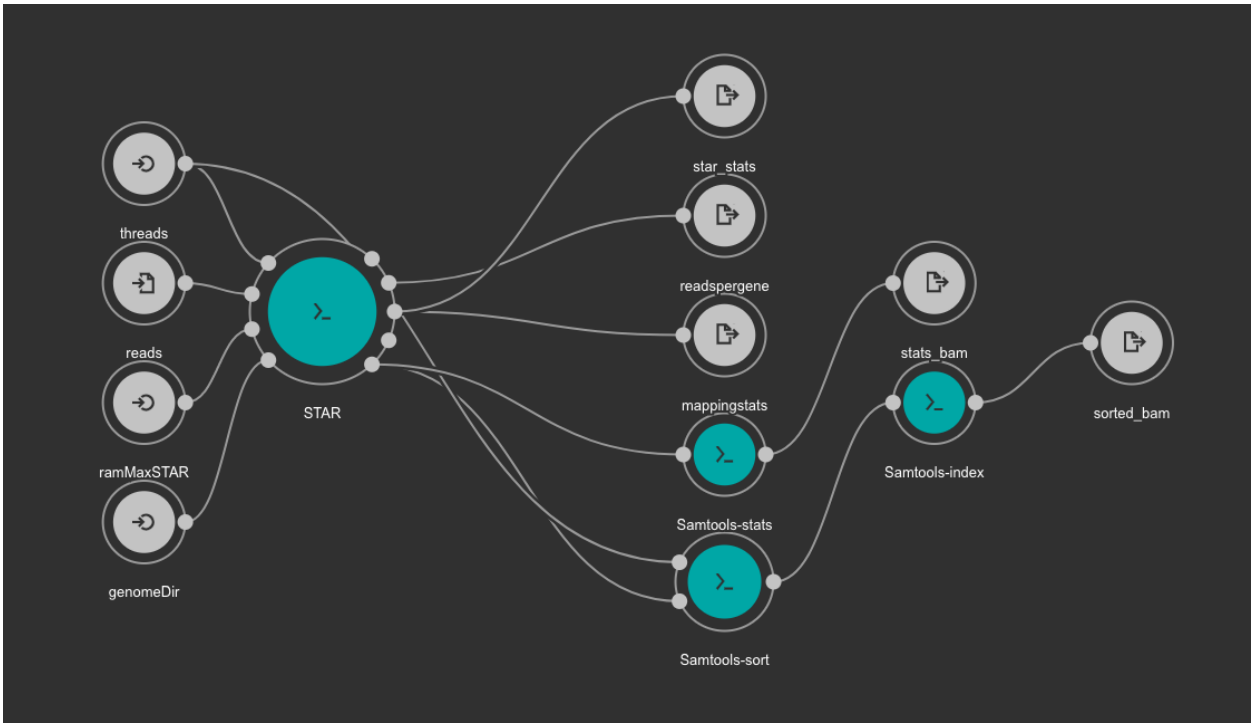



```
Executing: /home/r78v10a07/pm4ngs_venv/bin/cwltool --no-read-only --beta-use-biocontainers --parallel --on-error cont
inue --rm-tmpdir --tmp-outdir-prefix=/home/r78v10a07/Dros_lol_mut/tmp/ --tmpdir-prefix=/home/r78v10a07/Dros_lol_mut/t
mp/  /home/r78v10a07/Dros_lol_mut/bin/cwl/workflows/RNA-Seq/rnaseq-alignment-quantification.cwl alignment.yml > align
ment.log 2>&1 &
0
```

The CWL workflow for this step is: rnaseq-alignment-quantification.cwl



It is based in another workflow for the alignment: star-alignment.cwl



Once finished, the alignment and quantification workflow will create for each sample these files:

Note: STAR alignment stats

- DRR092341Aligned.out.stats
- DRR092341Log.final.out
- DRR092341ReadsPerGene.out.tab

Alignments files sorted from SAMtools

- DRR092341_sorted.bam
- DRR092341_sorted.bam.bai

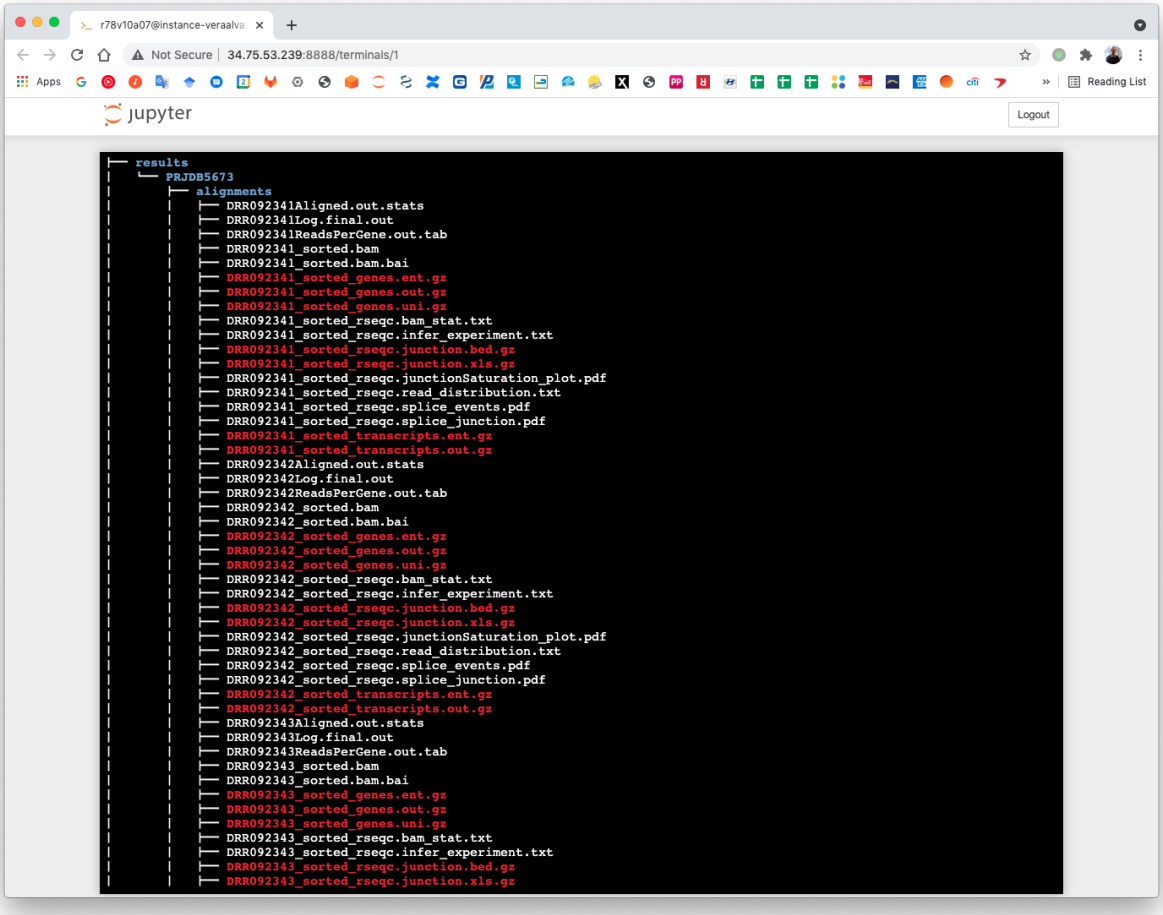
Read counts and TPMs from TPMCalculator

- DRR092341_sorted_genes.ent.gz
- DRR092341_sorted_genes.out.gz
- DRR092341_sorted_genes.uni.gz
- DRR092341_sorted_transcripts.ent.gz
- DRR092341_sorted_transcripts.out.gz

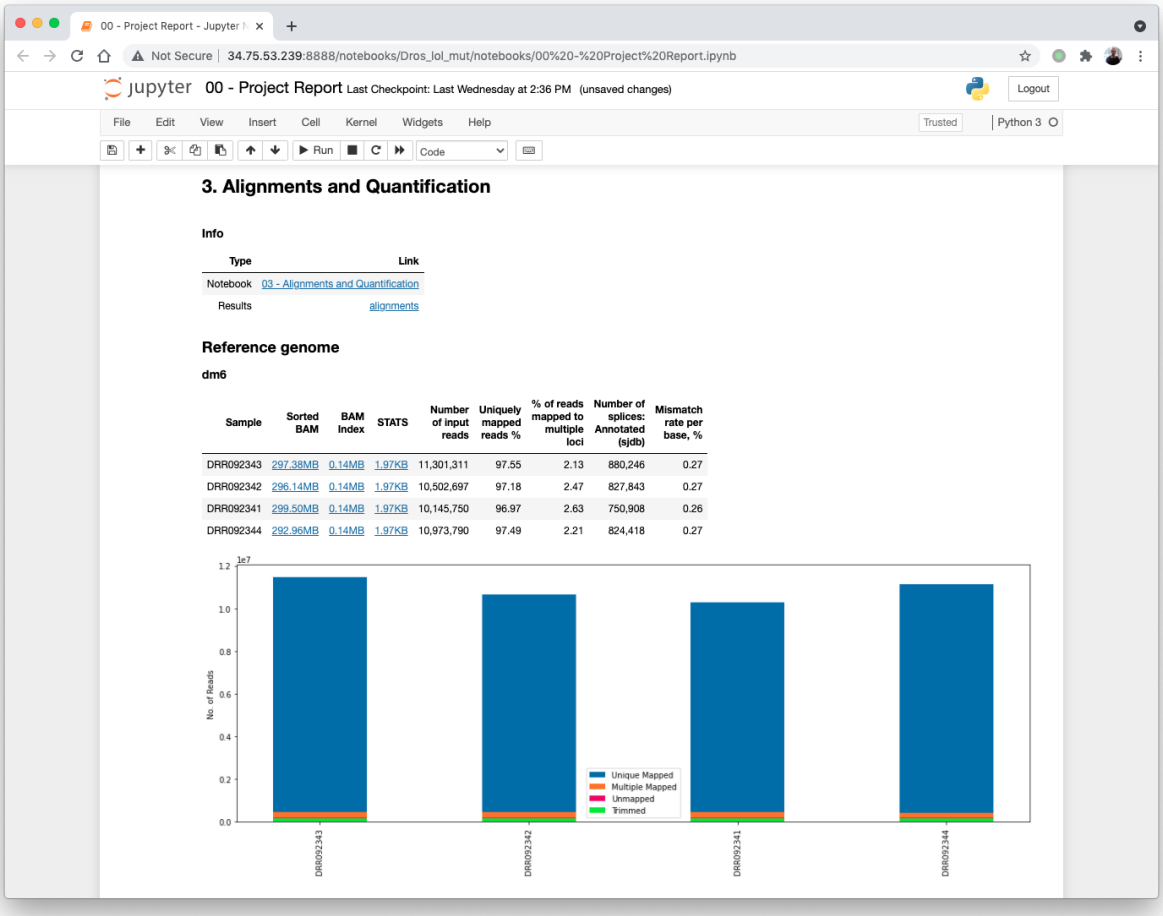
** Post-processing QC from RSeQC**

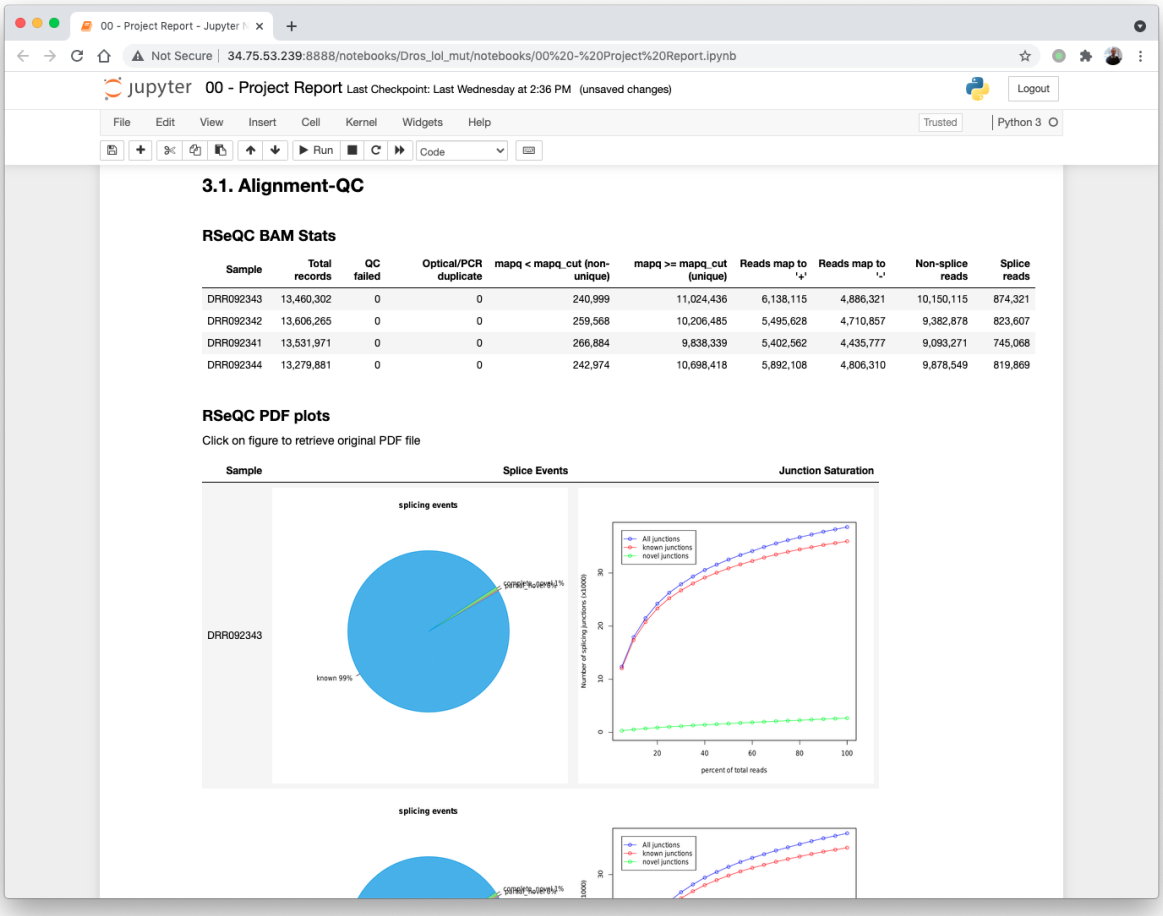
- DRR092341_sorted_rseqc.bam_stat.txt
- DRR092341_sorted_rseqc.infer_experiment.txt
- DRR092341_sorted_rseqc.junction.bed.gz
- DRR092341_sorted_rseqc.junction.xls.gz
- DRR092341_sorted_rseqc.junctionSaturation_plot.pdf

- DRR092341_sorted_rseqc.read_distribution.txt
- DRR092341_sorted_rseqc.splice_events.pdf
- DRR092341_sorted_rseqc.splice_junction.pdf

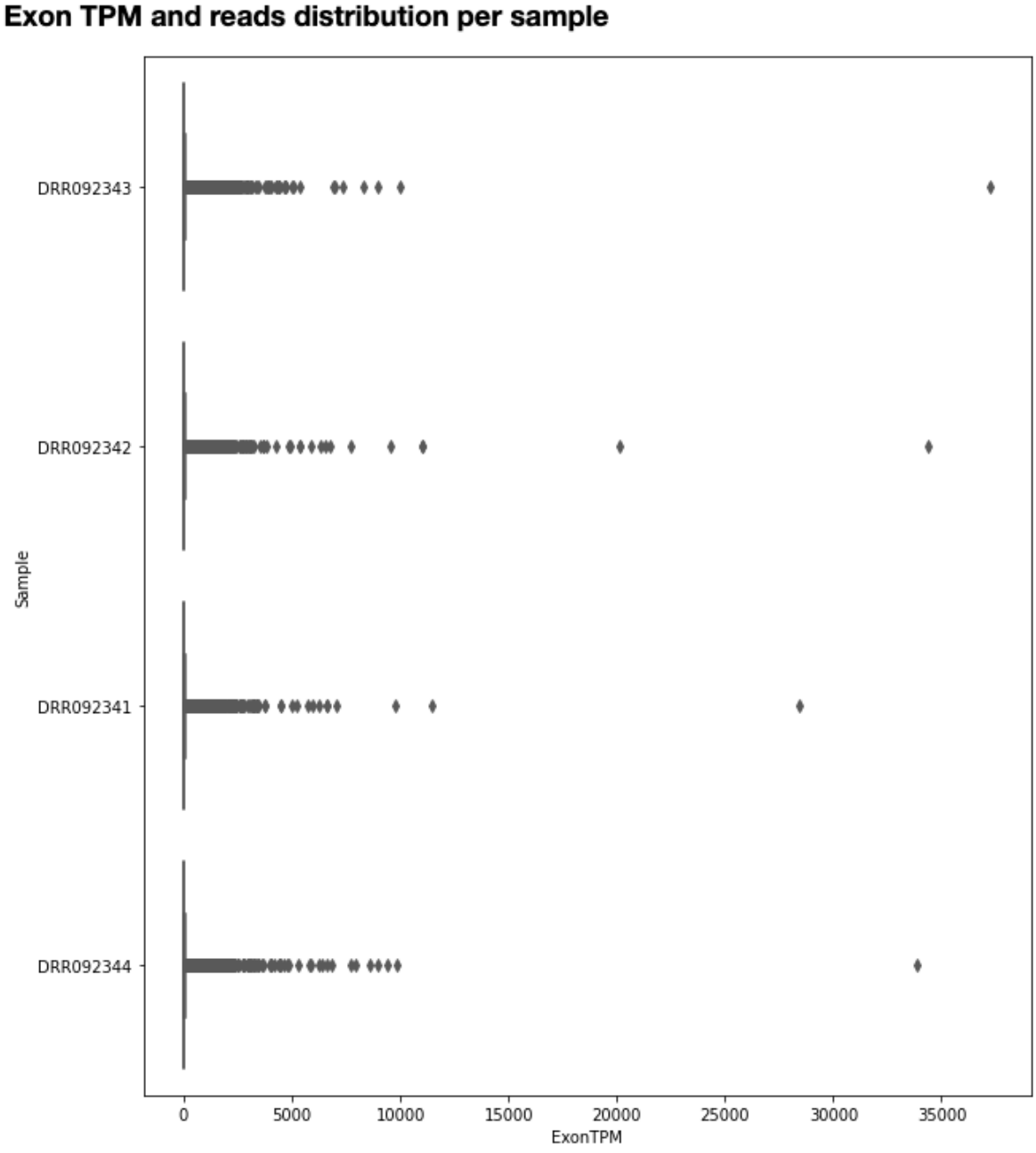


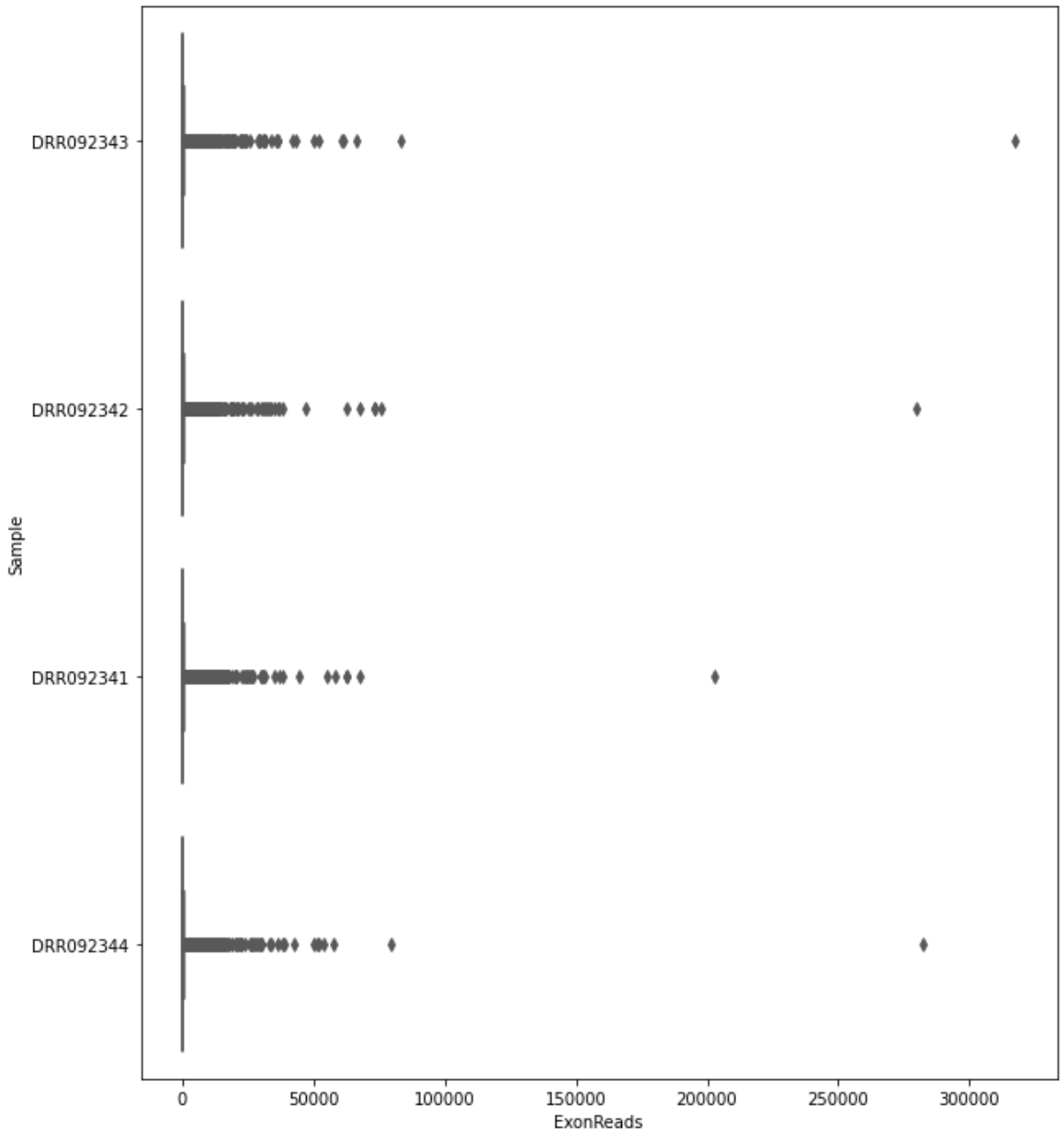
Updating the *00 - Project Report* notebook.


00 - Project Report - Jupyter
Not Secure
34.75.53.239:8888/notebooks/Dros_lol_mut/notebooks/00%20-%20Project%20Report.ipynb
jupyter
00 - Project Report
Last Checkpoint: Last Wednesday at 2:36 PM (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
3. Alignments and Quantification
Info
Type
Link
Notebook
03 - Alignments and Quantification
Results
alignments
Reference genome
dm6
Sample
Sorted BAM
BAM Index
STATS
Number of input reads
Uniquely mapped reads %
% of reads mapped to multiple loci
Number of splices: Annotated (sjdb)
Mismatch rate per base, %
DRR092343
297.38MB
0.14MB
1.97KB
11,301,311
97.55
2.13
880,246
0.27
DRR092342
296.14MB
0.14MB
1.97KB
10,502,697
97.18
2.47
827,843
0.27
DRR092341
299.50MB
0.14MB
1.97KB
10,145,750
96.97
2.63
750,908
0.26
DRR092344
292.96MB
0.14MB
1.97KB
10,973,790
97.49
2.21
824,418
0.27
1e7
No. of Reads
Unique Mapped
Multiple Mapped
Unmapped
Trimmed
DRR092343
DRR092342
DRR092341
DRR092344



The quantification values read counts and TPM are shown in a boxplot for easy comparison.


Exon TPM and reads distribution per sample
DRR092343
DRR092342
DRR092341
DRR092344
Sample
0
5000
10000
15000
20000
25000
30000
35000
ExonTPM



5.6 Differential Gene Expression Analysis

The DGE analysis is executed in the **04 - DGA** notebook. The workflow uses DESeq2 and EdgeR for the identification of differentially expressed genes. R scripts are available at deseq2-2conditions.cwl and edgeR-2conditions.cwl.

In our approach, genes are designated as differentially expressed if they are identified by DESeq2 and EdgeR. Those genes are shown under the **Intersection** results.

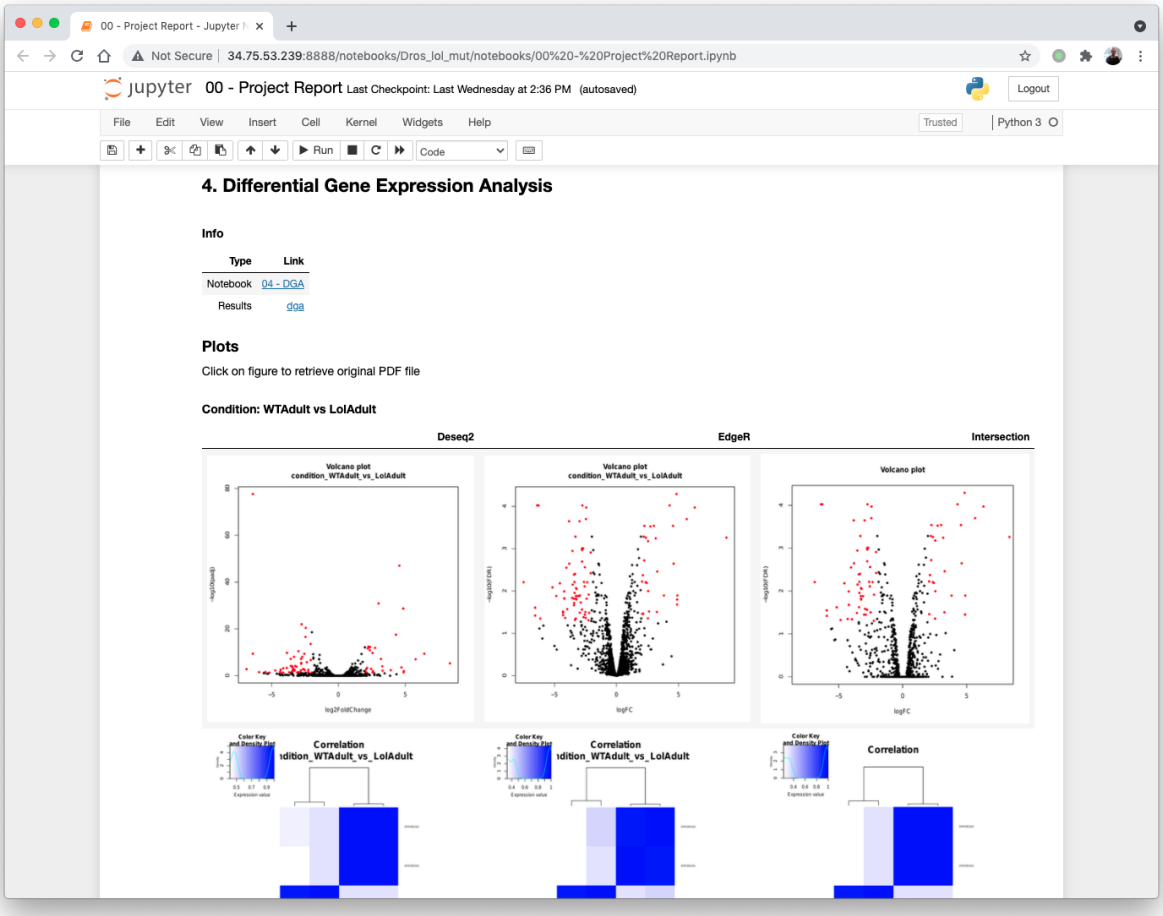
```
results/PRJDB5673/dga/
├── WTAdult_vs_LolAdult_deseq2_dga.log
├── WTAdult_vs_LolAdult_edge_dga.log
```

(continues on next page)

(continued from previous page)

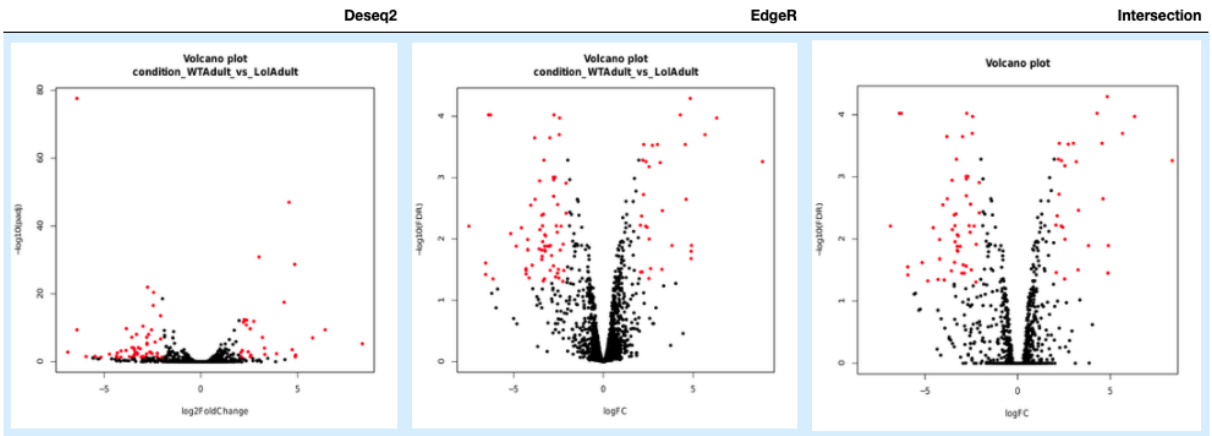
```
├── WTAdult_vs_LolAdult_heatmap_union.log
├── WTAdult_vs_LolAdult_volcano_union.log
├── condition_WTAdult_vs_LolAdult_deseq2.csv
├── condition_WTAdult_vs_LolAdult_deseq2_correlation_heatmap.pdf
├── condition_WTAdult_vs_LolAdult_deseq2_expression_heatmap.pdf
├── condition_WTAdult_vs_LolAdult_deseq2_pca.csv
├── condition_WTAdult_vs_LolAdult_deseq2_pca.pdf
├── condition_WTAdult_vs_LolAdult_deseq2_volcano.pdf
├── condition_WTAdult_vs_LolAdult_edgeR.csv
├── condition_WTAdult_vs_LolAdult_edgeR_correlation_heatmap.pdf
├── condition_WTAdult_vs_LolAdult_edgeR_expression_heatmap.pdf
├── condition_WTAdult_vs_LolAdult_edgeR_pca.csv
├── condition_WTAdult_vs_LolAdult_edgeR_pca.pdf
├── condition_WTAdult_vs_LolAdult_edgeR_volcano.pdf
├── condition_WTAdult_vs_LolAdult_intersection.csv
├── condition_WTAdult_vs_LolAdult_intersection_correlation_heatmap.pdf
├── condition_WTAdult_vs_LolAdult_intersection_expression_heatmap.pdf
├── condition_WTAdult_vs_LolAdult_intersection_over-expressed.csv
├── condition_WTAdult_vs_LolAdult_intersection_pca.pdf
├── condition_WTAdult_vs_LolAdult_intersection_under-expressed.csv
└── condition_WTAdult_vs_LolAdult_intersection_volcano.pdf

0 directories, 23 files
```

Updating the *00 - Project Report* notebook.

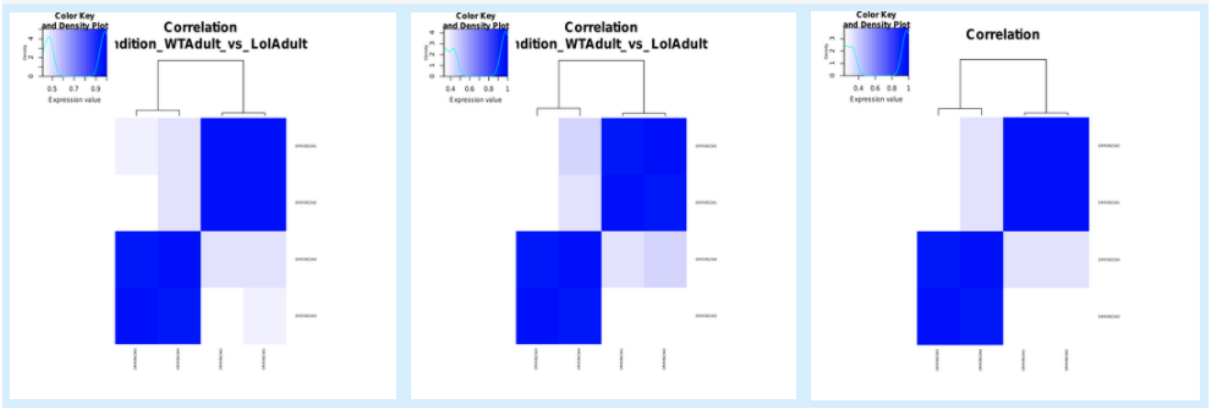


DGE plots are automatically generated.

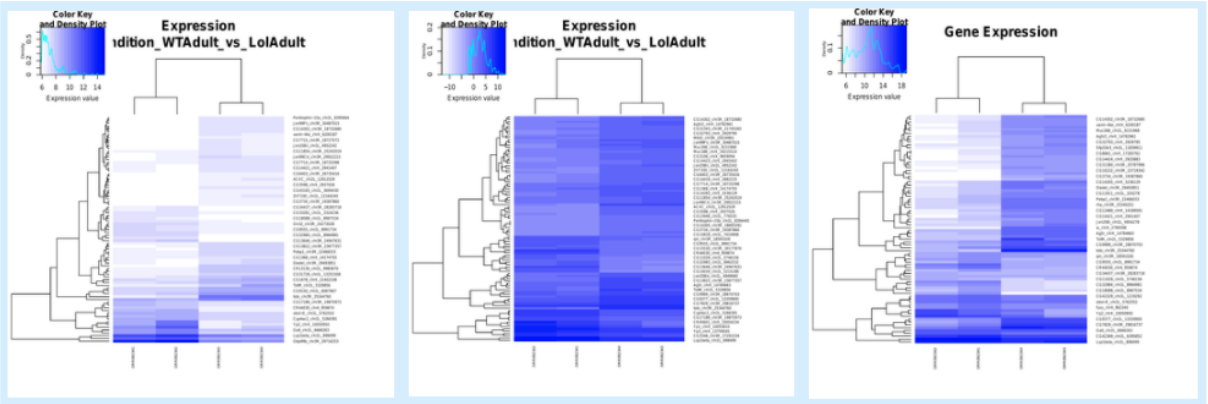
5.6.1 Volcano Plots



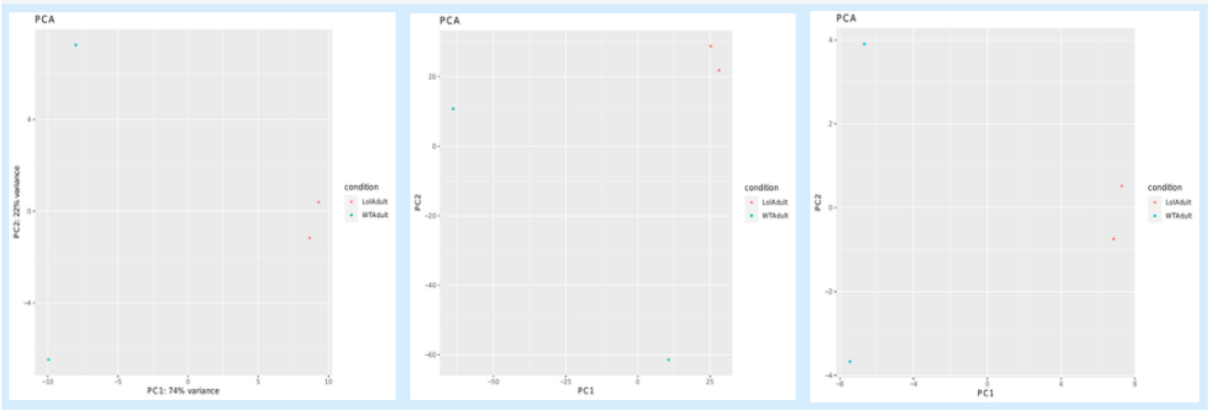
5.6.2 Sample correlation



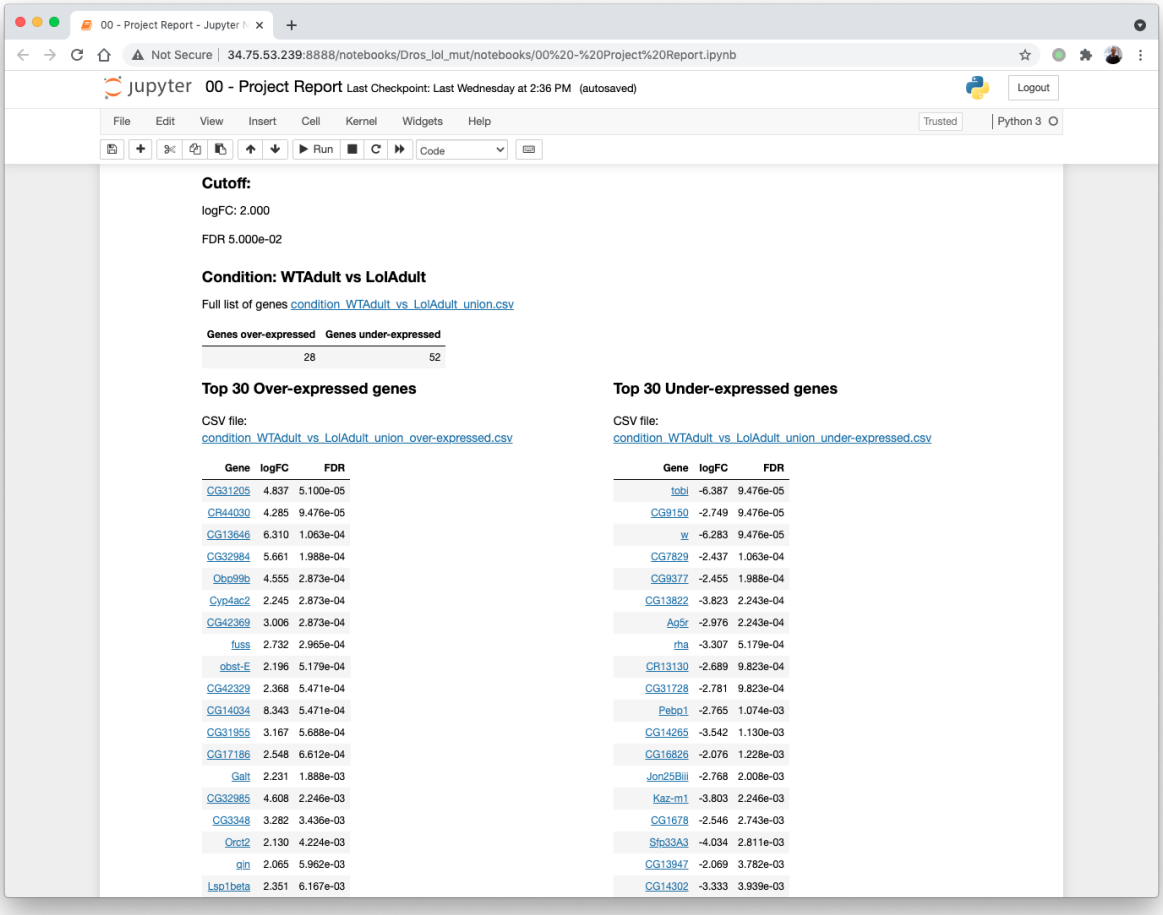
5.6.3 Expression correlation



5.6.4 PCA plots



5.6.5 List of differentially expressed genes



00 - Project Report - Jupyter

Not Secure | 34.75.53.239:8888/notebooks/Dros_lol_mut/notebooks/00%20-%20Project%20Report.ipynb

jupyter 00 - Project Report Last Checkpoint: Last Wednesday at 2:36 PM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

Cutoff:

logFC: 2.000

FDR 5.000e-02

Condition: WTAdult vs LolAdult

Full list of genes condition_WTAdult_vs_LolAdult_union.csv

Genes over-expressed	Genes under-expressed
28	52

Top 30 Over-expressed genes

CSV file:
condition_WTAdult_vs_LolAdult_union_over-expressed.csv

Gene	logFC	FDR
CG31205	4.837	5.100e-05
CR44030	4.285	9.476e-05
CG13646	6.310	1.063e-04
CG32984	5.661	1.988e-04
Obp99b	4.555	2.873e-04
Cyp4ac2	2.245	2.873e-04
CG42369	3.006	2.873e-04
fuss	2.732	2.965e-04
obst-E	2.196	5.179e-04
CG42329	2.368	5.471e-04
CG14034	8.343	5.471e-04
CG31955	3.167	5.688e-04
CG17186	2.548	6.612e-04
Galt	2.231	1.888e-03
CG32985	4.608	2.246e-03
CG3348	3.282	3.436e-03
Orct2	2.130	4.224e-03
gin	2.065	5.962e-03
Lsp1beta	2.351	6.167e-03

Top 30 Under-expressed genes

CSV file:
condition_WTAdult_vs_LolAdult_union_under-expressed.csv

Gene	logFC	FDR
tobi	-6.387	9.476e-05
CG9150	-2.749	9.476e-05
w	-6.283	9.476e-05
CG7829	-2.437	1.063e-04
CG9377	-2.455	1.988e-04
CG13822	-3.823	2.243e-04
Ag5r	-2.976	2.243e-04
rha	-3.307	5.179e-04
CR13130	-2.689	9.823e-04
CG31728	-2.781	9.823e-04
Pebp1	-2.765	1.074e-03
CG14265	-3.542	1.130e-03
CG16826	-2.076	1.228e-03
Jon25Biii	-2.768	2.008e-03
Kaz-m1	-3.803	2.246e-03
CG1678	-2.546	2.743e-03
Sfp33A3	-4.034	2.811e-03
CG13947	-2.069	3.782e-03
CG14302	-3.333	3.939e-03

5.7 Gene Ontology Enrichment

Gene Ontoogy enrichment analysis is executed with an *in-house* developed python package named goenrichment. This tool creates a database by integrating information from Gene Ontology, NCBI Gene using the files gene_info and gene2go. Read more in the package web page to create a database for another organism.

Available databases can be downloaded from https://ftp.ncbi.nlm.nih.gov/pub/goenrichment/

Next image shows the pre-defined code in PM4NGS for downloading the GO enrichment database for *Drosophila*. Current version of the database contains **43 987** GO terms cros-referenced to **13 741** genes.

Loading Gene2GO database from NCBI FTP

This database is created with the GO-enrichment python package: https://gitlab.com/r78v10a07/goenrichment

There are pre-computed database at: ftp://ftp.ncbi.nlm.nih.gov/pub/goenrichment

Loading Gene2GO database for human ¶

```
In [4]: goenrichDB = "https://ftp.ncbi.nlm.nih.gov/pub/goenrichment/goenrichDB_drosophila.pickle"
        godb = load_goenrichdb(goenrichDB)
        print('There are %d alternative ids in database' % (len(godb['alt_id'])))
        print('There are %d GO terms' % (len(godb['graph'].nodes())))
        print('There are %d genes in database' % (godb['M']))

        Loading go enrichment DB from: https://ftp.ncbi.nlm.nih.gov/pub/goenrichment/goenrichDB_drosophila.pickle
        There are 3297 alternative ids in database
        There are 43987 GO terms
        There are 13741 genes in database
```

The GO enrichment tool uses three parameters to identify differential GO terms between the two conditions.

Note: min_category_depth

Min GO term graph depth to include in the report. Default: 4

min_category_size

Min number of gene in a GO term to include in the report. Default: 3

max_category_size

Max number of gene in a GO term to include in the report. Default: 500 This 500 is good for very well annotated organism as human or mouse.

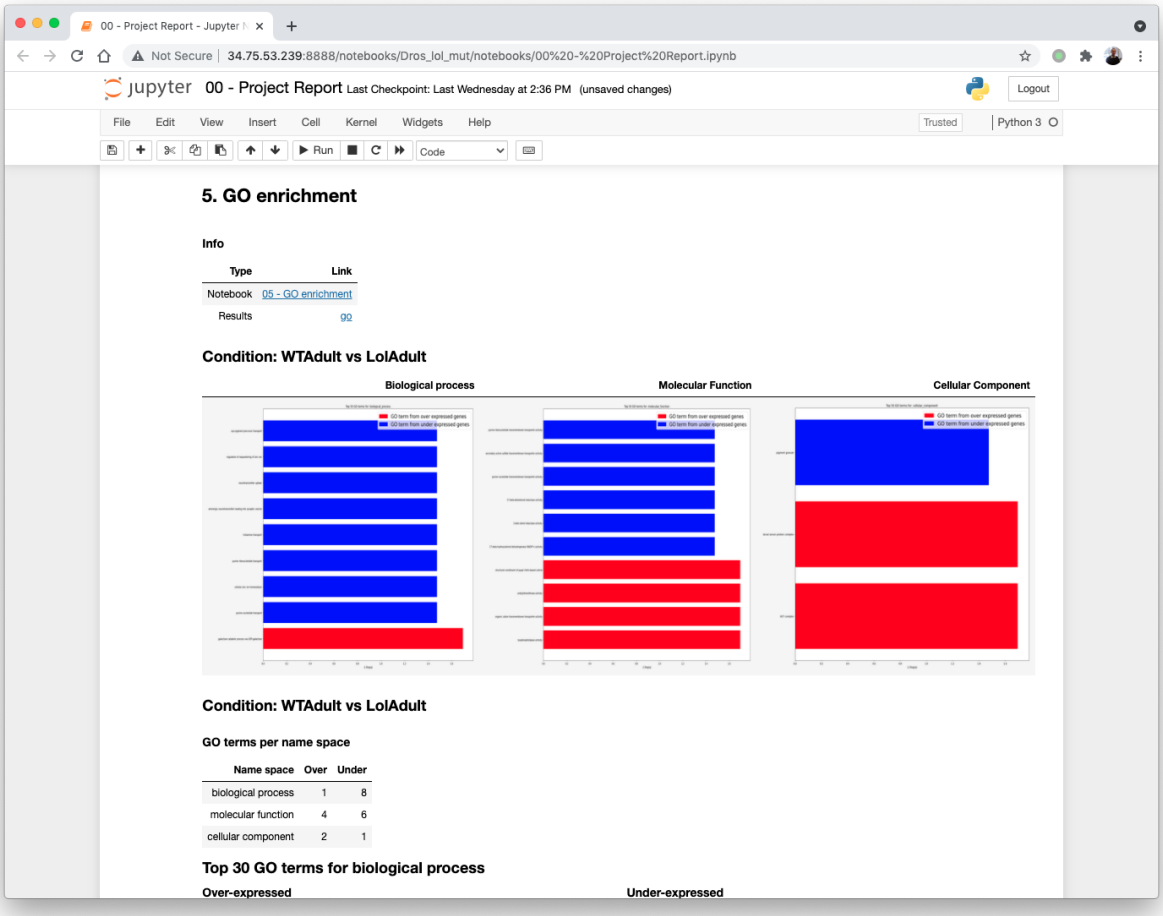
In the case of *Drosophila* use 10 to get relevant results.

In the Jupyter Notebook change the values to

GO enrichment parameters

```
In [33]: min_category_depth=4
         min_category_size=3
         max_category_size=10
```

Updating the *00 - Project Report* notebook.



The list of differential GO terms are available in the tables per GO name space.

Top 30 GO terms for biological process

Over-expressed

CSV file:
raw file

GO term	Name	FDR
GO:0033499	galactose catabolic process via UDP-gala	2.020e-02

Under-expressed

CSV file:
raw file

GO term	Name	FDR
GO:0015865	purine nucleotide transport	3.356e-02
GO:0006882	cellular zinc ion homeostasis	3.356e-02
GO:0015868	purine ribonucleotide transport	3.356e-02
GO:0051608	histamine transport	3.356e-02
GO:0015842	aminergic neurotransmitter loading into	3.356e-02
GO:0001504	neurotransmitter uptake	3.356e-02
GO:0061088	regulation of sequestering of zinc ion	3.356e-02
GO:0006856	eye pigment precursor transport	3.356e-02

Top 30 GO terms for molecular function

Over-expressed

CSV file:
raw file

GO term	Name	FDR
GO:0004622	lysophospholipase activity	2.020e-02
GO:0015101	organic cation transmembrane transporter	2.020e-02
GO:0070569	uridylyltransferase activity	2.020e-02
GO:0008011	structural constituent of pupal chitin-b	2.020e-02

Under-expressed

CSV file:
raw file

GO term	Name	FDR
GO:0072582	17-beta-hydroxysteroid dehydrogenase (NA	3.356e-02
GO:0000253	3-keto sterol reductase activity	3.356e-02
GO:0072555	17-beta-ketosteroid reductase activity	3.356e-02
GO:0015216	purine nucleotide transmembrane transpor	3.356e-02
GO:0008271	secondary active sulfate transmembrane t	3.356e-02
GO:0005346	purine ribonucleotide transmembrane tran	3.356e-02

Top 30 GO terms for cellular component

Over-expressed

CSV file:
raw file

GO term	Name	FDR
GO:0016590	ACF complex	2.020e-02
GO:0005616	larval serum protein complex	2.020e-02

Under-expressed

CSV file:
raw file

GO term	Name	FDR
GO:0048770	pigment granule	3.356e-02

CHAPTER

SIX

INTRODUCTION

This tutorial will show how to create and configure a Google Cloud Platform instance for Next Generation Sequencing (NGS) data analysis. PM4NGS is a program manager for NGS data analysis that is designed to generate a standard organizational structure including directory structures for the project, Jupyter notebooks for data management and CWL workflows for the pipeline execution.

CHAPTER

SEVEN

BACKGROUND READING

1. Vera Alvarez R, Pongor LS, Mariño-Ramírez L and Landsman D. PM4NGS, a project management framework for next-generation sequencing data analysis, GigaScience, Volume 10, Issue 1, January 2021, giaa141, https://doi.org/10.1093/gigascience/giaa141

2. Vera Alvarez R, Mariño-Ramírez L and Landsman D. Transcriptome annotation in the cloud: complexity, best practices, and cost, GigaScience, Volume 10, Issue 2, February 2021, giaa163, https://doi.org/10.1093/gigascience/giaa163

CHAPTER

EIGHT

HELP AND SUPPORT

For feature requests or bug reports, please open an issue on our GitHub Repository.