Tutorial on next generation sequencing (NGS) data analysis with emphasis on RNA sequencing (RNA-seq) technique

Prepared by:

Kennedy Mwangi …, MSc student, MKU/JKUAT

John Gitau……, MSc student, MKU/UoN

Mtakai Ngara, PhD, MKU/Karolinska Institutet

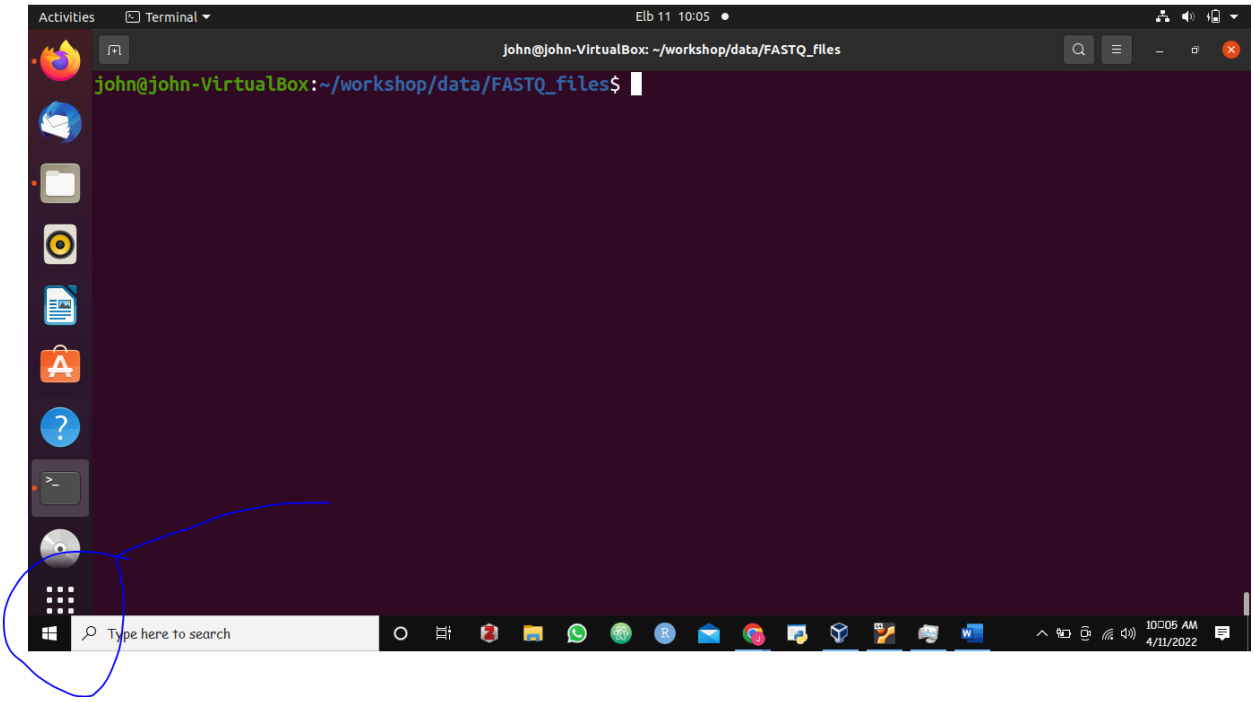
April 6, 2022

Scope

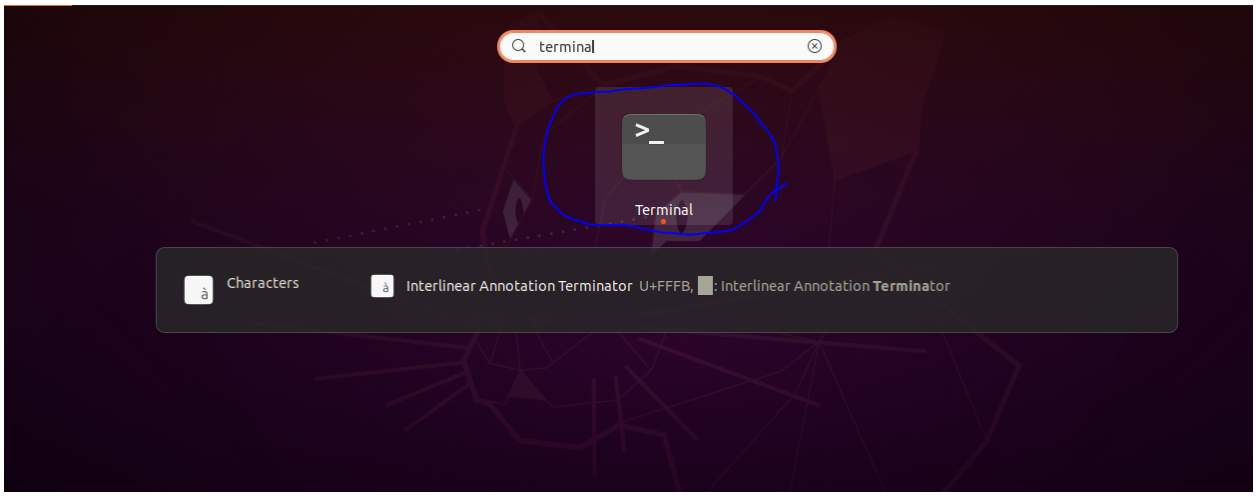
- Bioinformatics background and resources
- Next-generation sequencing (NGS) concepts and applications
- Gene expression profiling using RNA-sequencing technique
- Basic command line for Unix/Linux users
- Conda environment installation and basics
- NGS datasets
- Read trimming
- Quality control (QC) analysis
- Reference mapping
- Mapping quality evaluation
- Gene expression quantification
- Differential expression analysis
- Basic visualization

For participants using Windows and have installed Linux/Unix OS (ubuntu):

Once you have installed virtual box, click the Applications Icon on the lower left (shown below).

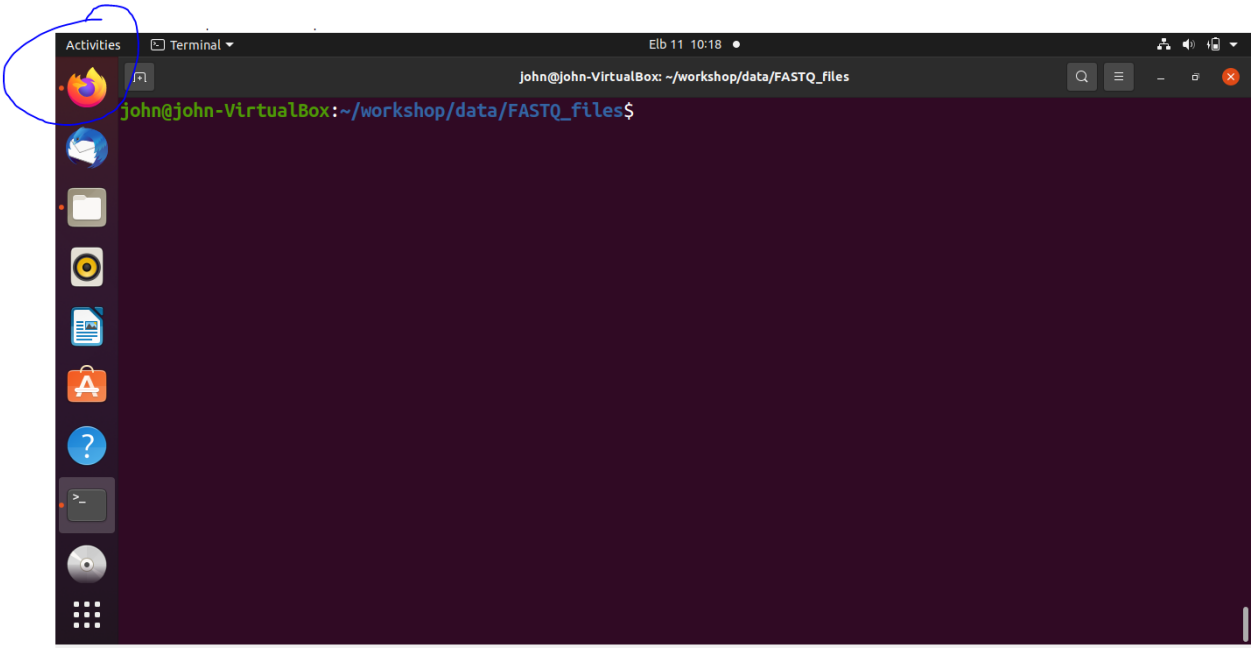


In the resultant search box, type terminal and click it to open it (see below)

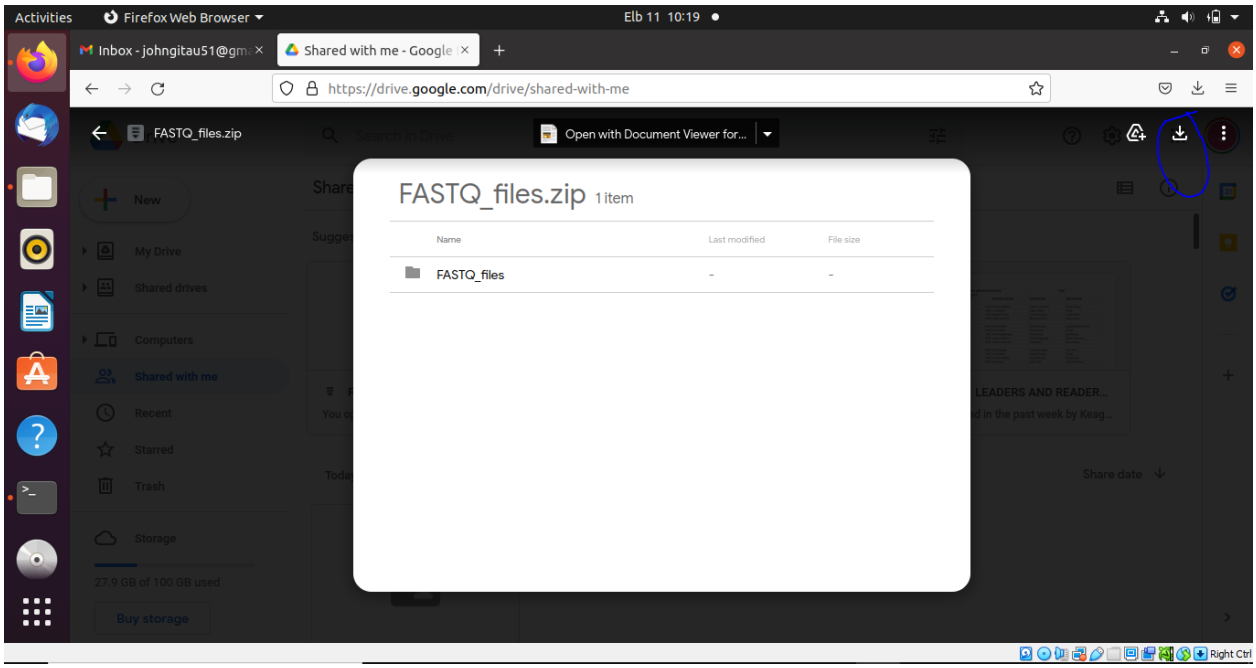


Once the terminal is open, type conda activate ngs where all tools as outlined in Software_installation_instructions.pdf- are contained.

To get the data into the working directory (data) in Linux, log in to your gmail using the Firefox browser provided in your virtual box (see below).



Within the shared directory, click on the data subdirectory and download the FASTQ_files.zip



Navigate to the downloads section and move the downloaded zipped folder to your data directory on your terminal.

Basic command line (CLI) and conda package manager

Basic command line (CLI)

This tutorial is mainly based on a Linux/Unix command-line. Using the command-line requires a Linux/Unix operating system (OS).
For those running Windows OS, you have an option to run any of the linux OSes on virtual box (VB) or installing Ubuntu on windows. See the instructions on how to download and install VB from the earlier shared pdf named 'Software_installation_instructions.pdf'.

We will go through some introductory material explaining the shell syntax. This is bash syntax but most of it will work on other shells (tcsh, zsh) as well.

What is a shell?

Wikipedia:
"In computing, a shell is a user interface for access to an operating system's services. In general, operating system shells use either a command-line interface (CLI) or graphical user interface (GUI), depending on a computer's role and particular operation…"

"CLI shells allow some operations to be performed faster in some situations, especially when a proper GUI has not been or cannot be created. However, they require the user to memorize all commands and their calling syntax, and also to learn the shell-specific scripting language, for example bash script."

Notes on Unix/Linux filesystem:

- The directory structure in a Linux system is not much different from any other system you worked with, e.g., Windows, MacOSX.
- However, on the command-line we navigate via commands and not via mouse clicks. Why is it necessary to use the command-line in the first place? Strictly speaking it is not, if you do not want to make use of programs on the command-line. However, the power of the Linux system becomes only obvious once we learn to make use of the command-line, thus navigating the directory structure via commands is one of the most important skills for you to learn.
- Examples of Linux environments: Ubuntu, Mint, Debian, Fedora etc.

Open a terminal

Open a terminal window and you are are ready to go.
On your linux desktop find: System Tools –> QTerminal (for LXDE environment) or type "Terminal" in the search box.

Useful resources (1, 2).

1. The Unix shell for novice: https://swcarpentry.github.io/shell-novice/
2. Access other courses online: https://software-carpentry.org/

Conda package manager

We will use the package/tool managing system called conda (3) to install some programs that we will use during the course. We assume you have conda installed, if not follow the instructions from 'Software_installation_instructions.pdf'.

Ensure conda is updated

```
$ conda update --yes conda
```

Ensure the essential conda channels are installed to make access to tools available

```
# Install some conda channels
# A channel is where conda looks for packages
$ conda config --add channels defaults
$ conda config --add channels bioconda
$ conda config --add channels conda-forge
```

Creating environments

```
# create an environment
$ conda create -n ngs python=3
# activate the environment
$ conda activate ngs
```

Install modules (software)

```
# Install more tools into the environment
$ conda install package
```

General conda commands

```
# to search for packages
$ conda search [package]
# To update all packages
$ conda update --all –yes
# List all packages installed
$ conda list [-n env]

# conda list environments
$ conda env list
# create new env
$ conda create -n [name] package [package] ...
# activate env
$ conda activate [name]

# deactivate env
$ conda deactivate
```

3. http://conda.pydata.org/miniconda.html

QC analysis of NGS data analysis

There are a few steps one need to do when getting the raw sequencing data from the sequencing facility:

1. Remove PhiX sequences (we are not going to do this)
2. Adapter trimming
3. Quality trimming of reads
4. Quality assessment

Get the data

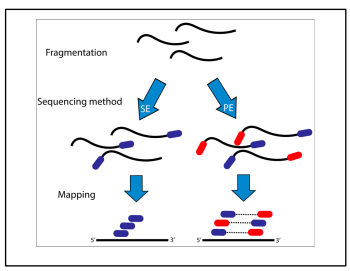
Downloading genomics data is important for reuse and certain tools such as SRA Explorer (https://sra-explorer.info/) can be useful especially when processing a limited number of samples. Using the accession e.g., 'PRJNA229998' from in search field and select all the runs to the collection and download the ftp links. You should have a text file named: 'sra_explorer_fastq_urls.txt' and using the URLs you may download the compressed fastq.gz files with the corresponding meta data.

However, for this tutorial we will use a subsampled data from 4 samples (2 treated and 2 controls) from the published study (https://pubmed.ncbi.nlm.nih.gov/24926665/). Hence from google drive link (https://drive.google.com/drive/folders/1958ujeoJauMVIABgr6_DE1jSC0mTIzlP?usp=sharing) download the 'data' folder unzip it.

For those using terminal:

```
# create a directory you work in
$ mkdir analysis
# change into the directory
$ cd analysis
#create a directory for the data
$mkdir data
# change into the data directory
$cd data
#Copy the sequencing data files (compressed FASTQs) into the data directory
$cp data/*.fastq.gz .
```

The data is from a paired-end sequencing run data (see Fig. 1.0 below) from an Illumina HiSeq 2000 platform (4,5). Thus, we have two files, one for each end of the read.



4. Illumina platforms: https://www.illumina.com/systems/sequencing-platforms.html
5. S. Goodwin, et al, Coming of age: ten years of next-generation sequencing technologies. Nature Reviews Genetics (2016).

Investigate the data

1. Use the command-line to get some ideas about the file.
2. What kind of files are we dealing with?
3. How many sequence reads are in the file?

The FASTQ file format

The data we receive from the sequencing is in FASTQ format which is a text-based format that stores both the sequence (normally nucleotide) and the corresponding quality scores in ASCII encoded fashion.

A FASTQ file normally has four lines:

- Line 1 begins with a '@' character and is followed by a sequence identifier and an optional description (like a FASTA title line).
- Line 2 is the raw sequence letters.
- Line 3 begins with a '+' character and is optionally followed by the same sequence identifier (and any description) again.
- Line 4 encodes the quality values for the sequence in Line 2, and must contain the same number of symbols as letters in the sequence.

A useful tool to decode base qualities can be found here:
http://broadinstitute.github.io/picard/explain-qualities.html

 # change into the directory
$ cd analysis

QC process

PhiX genome

PhiX is a nontailed bacteriophage with a single-stranded DNA and a genome with 5386 nucleotides. PhiX is used as a quality and calibration control for sequencing runs. PhiX is often added at a low known concentration, spiked in the same lane along with the sample or used as a separate lane. As the concentration of the genome is known, one can calibrate the instruments. Thus, PhiX genomic sequences need to be removed before processing your data further as this constitutes a deliberate contamination. The steps involve mapping all reads to the “known” PhiX genome and removing all of those sequence reads from the data.

However, your sequencing provider might not have used PhiX, thus you need to read the protocol carefully, or just do this step in any case.

Adapter trimming

The process of sequencing DNA via Illumina technology requires the addition of some adapters to the sequences. These get sequenced as well and need to be removed as they are artificial and do not belong to the species we try to sequence. We have to deal with a trade-off between accuracy of adapter removal and speed of the process since trimming does take some time when handling larger sequencing data.

Also, we have generally two different approaches when trimming adapter:

1. We can use a tool that takes an adapter or list of adapters and removes these from each sequence read.
2. We can use a tool that predicts adapters and removes them from each sequence read.

For the first approach we need to know the adapter sequences that were used during the sequencing of our samples. Normally, you should ask your sequencing provider, who should be providing this information to you. Illumina itself provides a document that describes the adapters used for their different technologies. Also, tools such as FastQC (6) tool, provides a collection of contaminants and adapters.

Here, we are going to use the second approach with a tool called fastp to trim adapters and do quality trimming. Fastp (7) has a few characteristics which make it a great tool, most importantly: it is pretty fast, provides good information after the run, and can do quality trimming as well, thus saving us to use another tool to do this.

Quality trimming of our sequencing reads will remove bad quality called bases from our reads.

```
# create env and install tools
$ conda create --yes -n qc fastp fastqc multiqc
# activate env
$ conda activate qc
# create directory for the placing the trimmed output
$ mkdir trimmed
```

Here, as an example we are trimming the sequence reads:

```
$ fastp --detect_adapter_for_pe --overrepresentation_analysis --correction --cut_right --
thread 2 --html trimmed/GSM1275862.fastp.html --json trimmed/GSM1275862.fastp.json
-i data/GSM1275862_R1.fastq.gz -I data/GSM1275862_R2.fastq.gz -o
trimmed/GSM1275862_R1.fastq.gz -O trimmed/GSM1275862_R2.fastq.gz
```

- **--detect_adapter_for_pe**: Specifies that we are dealing with paired-end data.
- **--overrepresentation_analysis**: Analyse the sequence collection for sequences that appear too often.
- **--correction**: Will try to correct bases based on an overlap analysis of read1 and read2.
- **--cut_right**: Will use quality trimming and scan the read from start to end in a window. If the quality in the window is below what is required, the window plus all sequence towards the end is discarded and the read is kept if it's still long enough.
- **--thread**: Specify how many concurrent threads the process can use.
- **--html** and **--json**: We specify the location of some statistics files.
- **-i** data/GSM1275862_R1.fastq.gz **-I** data/ GSM1275862_R2.fastq.gz: Specifies the two input read files
- **-o** trimmed/GSM1275862_R1.fastq.gz **-O** trimmed/GSM1275862_R2.fastq.gz: Specifies the two desired output read files

---

6. FastQC: http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

7. Fastp: https://github.com/OpenGene/fastp

To-do: Now trim the remaining PE samples namely: GSM1275866, GSM1275867 and GSM1275875.

Quality assessment of sequencing reads

FastQC

FastQC (8) is a very simple program to run that provides information about sequence read quality. It aims to provide a simple way to do some quality control checks on raw sequence data coming from high throughput sequencing pipelines. It provides a modular set of analyses which you can use to give a quick impression of whether your data has any problems of which you should be aware before doing any further analysis.

$ fastqc –help

The basic command:

$ fastqc -o RESULT-DIR INPUT-FILE.fq (.gz) ...

- -o RESULT-DIR is the directory where the result files will be written
- INPUT-FILE.fq is the sequence file to analyze, can be more than one file

$ mkdir fastqc_report

$ fastqc -o fastqc_report data/GSM1275862_R1.fastq.gz

NB: The result will be a HTML page per input file that can be opened in a web-browser.

For some help on FastQC results and interpretation see the links (10,11).

To-do: now use FastQC to check the quality of the remaining samples.

$ fastqc -o fastqc_report data/*fastq.gz

---

8. FastQC: http://www.bioinformatics.babraham.ac.uk/projects/fastqc/
9. Fastp: https://github.com/OpenGene/fastp
10. http://www.bioinformatics.babraham.ac.uk/projects/fastqc/
11. http://www.bioinformatics.babraham.ac.uk/projects/fastqc/Help/3%20Analysis%20Modules/

MultiQC

MultiQC (12) is an excellent tool to put FastQC (and other tool) results of different samples into context. It compiles all FastQC (13) results and fastp statistics into one nice webpage. The use of MultiQC (12) is simple. Just provide the command with directories where multiple results are stored and it will compile a nice report, for example:

$ mkdir multiqc_report

$ multiqc -o multiqc_report fastqc_report

Run FastQC and MultiQC on the trimmed data

To do:

1. Create a directory for the results –> trimmed_fastqc
2. Run FastQC on all trimmed files.
3. Visit the FastQC (13) website and read about sequencing QC reports for good and bad Illumina sequencing runs.
4. Run MultiQC on the trimmed_fastqc and trimmed directories
5. Compare your results examples and write down your observations with regards to the data.