



LEIDEN UNIVERSITY MEDICAL CENTER

Getting started with NGS data analysis

Michiel van Galen

Leiden Genome Technology Center

Department of Human Genetics

Center for Human and Clinical Genetics



Overview

- How to work with Linux
- Working with NGS tools
- Cheatsheet is available in the course repository
- Practice during the practical

Before we start...

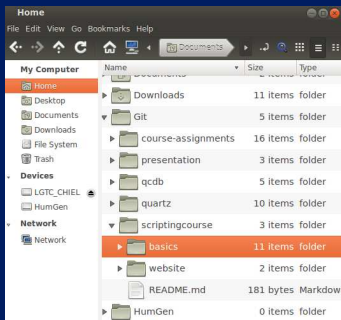
- Linux is case sensitive
- Tab completion: start typing then TAB
- To end command line execution:
 - 'ctrl-c'
- To show the manual of a command: 'man COMMAND'

Using a terminal

Folder structure

- Always keep the directory structure in mind

Figure 1 : Tree structure



Folder structure

```
1 $ pwd
2 /home/mvangalen/
3 $ cd ..
4 $ pwd
5 /home/
6 $ cd mvangalen
7 $ pwd
8 /home/mvangalen/
```

Listing 1 : Navigation

- 'pwd' Print working dir
- 'cd <dir>' Change dir to <dir>
- Try 'cd' and 'cd -'

Inspect files

- Various commands are useful when inspecting files
- less, more
 - View the file and allow scrolling
 - Press 'q' to stop these processes
- head, tail
 - Show the first or last 10 lines
 - Or more, try -n50 for example
- 'cat' dumps the complete file to screen

Copy and move

- Copying and moving files is fairly simple
- Call the command and supply the source(s) and target

```
1  $ ls
2  original.txt work
3  $ cp original.txt another.txt
4  $ ls
5  another.txt original.txt work
6  $ mv original.txt archive.txt
7  $ ls
8  archive.txt another.txt work
9  $ mv archive.txt another.txt work/
10 $ ls
11 work
12 $ ls work
13 archive.txt another.txt
```

Listing 2 : Copy and move

Create links

- Instead of a copy we can also make links
- Useful when you need big files in another folder
- Example of linking a file from /data/raw/big.fq :

```
1 $ pwd
2 /data/analysis
3 $ ln -s /data/raw/big.fq big.fastq
4 $ ls -l
5 big.fastq -> /data/raw/big.fq
6 $ rm big.fastq;ls
7 $ ls /data/raw
8 big.fq
```

Listing 3 : Links

Create and remove folders

- For folders the same syntax applies
- ; allows more commands on one line
- -r removes recursively

```

1  $ mkdir important
2  $ ls
3  important
4  $ cd important
5  $ touch secret.txt
6  $ pwd;ls
7  /home/mvangalen/important
8  secret.txt
9  $ cd ..
10 $ rm -r important

```

Listing 4 : Folders

Rename files

- Rename a file can be done with the mv command
 - 'mv file old new'
- There is also rename. Ideal for multiple files
- Syntax : rename 's/old/new/' targets
- Regular expression

```
1  rename -fvn 's/fastq/fq/' *.fastq
2
3  -f force overwriting.
4  -v verbose.
5  -n dry run.
```

Listing 5 : Rename

Redirecting output to files

- The output on your screen is a so called stream
- You can redirect this stream to a file
- `>` creates a new file, overwriting existing one
- `>>` Appends to a file, preserving the data

```

1  $ ls
2  Docs
3  $ ls > list.txt; cat list.txt
4  Docs
5  $ ls >> list.txt; cat list.txt
6  Docs
7  Docs

```

Listing 6 : Appending output stream

Two different streams

- In fact there are multiple streams: standard out and standard error
- Standard error usually displays notifications
- We can redirect this stream to standard out using `&>`
- This way we can catch the error output in a file as well

```

1  $ ls *mp3 *avi &> my_list.txt
2  $ cat my_list.txt
3  ls: cannot access *avi: No such file or directory
4  dancing-queen.mp3
5  waterloo.mp3

```

Listing 7 : Appending both streams

Two different streams

- Standard error uses stream 2 and can therefor also be redirected individually
- Some ideas for redirecting streams:

```

1  $ ls *mp3 *avi >list.txt
2  ls: cannot access *avi: No such file or directory
3
4  $ ls *mp3 *avi 2>errors.txt
5  dancing-queen.mp3
6  waterloo.mp3
7
8  $ ls *mp3 *avi 2>errors.txt >list.txt
9
10 $ ls &> /dev/null

```

Listing 8 : Appending both streams

Piping and redirecting

- Pipes are useful to use output from one command as input for another
- Can be used for most of the UNIX commands
- Check if output is valid input for the next command
- Use a hyphen - for a command which requires a file

```
1  $ ls
2  abba_poster.jpg dancing_queen.FLAC
3  list.txt sos.mp3 waterloo.mp3
4  $ ls | grep 'mp3' > my_music_collection.txt
5  $ ls | grep 'FLAC' >> my_music_collection.txt
6
7  $ samtools view -b dna.sam | samtools sort - dna.srt
```

Listing 9 : Navigation

Tee

- Duplicate the stream to a file with ‘tee’
- Displays the output and writes

```

1  $ ls | tee list.txt
2  happy_new_year.mp3 mama_mia.mp3
3  $ cat list.txt
4  happy_new_year.mp3
5  mama_mia.mp3

```

Listing 10 : Navigation

Writing a shell script

Do something for each something

- Many times it's helpful to do something for each something
- A simple for-one-liner in bash can help us out
- for this in that; do something; done

```
1  for i in 1 2 3; do echo $i; done
2
3  for line in $(cat ignore); do echo $line; done
```

Listing 11 : For loop

Writing a shell script

Editor

- nano - Nano's ANOther editor
 - Simple editor, doesn't rely on the mouse
 - Many useful features, invoked by holding the ctrl key
 - For example: ^X Exit means ctrl-x to exit

Figure 2 : Nano



Writing a shell script

Writing a simple script

- Shebang line
- Comment

```
1  #!/bin/bash
2  clear
3  # Say something original here.
4  echo "Hello world"
```

Listing 12 : Hello world

Running a script

- Typing : `./world.sh`
 - Error, permission denied
- Make it executable first!
 - `'chmod 755 world.sh'`
- What also works is typing : `'bash ./world.sh'`

Writing a shell script

Arguments in a bash script

- Sometimes it can be nice to pass an argument to a script
- In bash we can reference them with the dollar sign and a number

```

1  $ cat parser.sh
2  #!/bin/bash
3  echo "Parsed $1 and $2."
4
5  $ bash parser.sh ABBA rocks
6  Parsed ABBA and rocks.
```

Listing 13 : Argument parsing.

Tip of the iceberg...

- There are many more cool tricks available
- Read the man page or try one of these:
 - `uniq`
 - `paste`
 - `split`
- Don't forget the cheat sheet!

Analysis workflow

For this course we will go over the steps to analyze a human dataset short reads for variants:

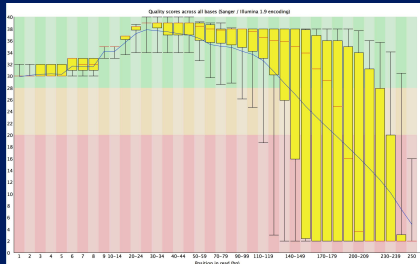
- Quality control
- Quality trimming
- Alignment
- Variant calling
- Annotation

In the practical we will practice how to apply these steps using standard tools

Quality control

- FastQC: A quality control tool for high throughput sequence data.
- Assess the quality of your data in a fastq file

Figure 3 : FastQC



Quality trimming

- Sickle: A windowed adaptive trimming tool for FASTQ files using quality.
- Only maintain the high-quality bases in a given window size
- Works on single and paired end

```

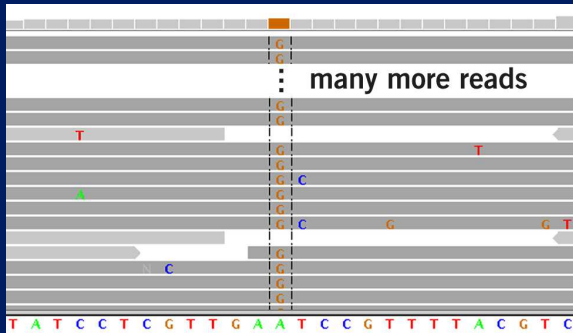
1  @Header
2  ACGTACGTACGT
3  +
4  !# II!JJJI##!
5
6  Will result in:
7  —GTACGTA—

```


Alignment

High throughput mapping of reads to a known reference genome

- Allow mismatches



Alignment

- Many, many different tools available
- Input, output, paired end and multi threading are usually supported
- Underlying methods and technical background vary
 - Gapped or ungapped alignment
 - CPU or RAM focus
 - Indexing of genome
- SAM/BAM output is widely adapted and used for many downstream tools
- In the practical we will use Bowtie

http://en.wikipedia.org/wiki/List_of_sequence_alignment_software

Variant calling

- Multiple options available each with their own pros and cons
 - GATK pipeline
 - Samtools
 - Varscan
- For the practical we will stick to Samtools
- Variant Call Format (vcf) is widely used

Annotation

Adding some context to your variants greatly benefits the interpretation

ID	Chr	Position	Ref	Alt	Gene	Function	Occur	Refseq	Change
rs1050171	chr7	55249063	G	A	EGFR	synonymous	HET	NM_005228	Q787Q
rs1873778	chr4	55141055	A	G	PDGFRA	synonymous	HOM	NM_006206	P567P
rs2228230	chr4	55152040	C	T	PDGFRA	synonymous	HET	NM_006206	V824V
rs41115	chr5	112175770	G	A	APC	synonymous	HET	NM_000038	T1493T
rs41115	chr5	112175770	G	A	APC	synonymous	HET	NM_001127510	T1493T
rs41115	chr5	112175770	G	A	APC	synonymous	HET	NM_001127511	T1475T
rs7688609	chr4	1807894	G	A	FCFR3	synonymous	HOM	NM_000142	T651T
rs7688609	chr4	1807894	G	A	FCFR3	synonymous	HOM	NM_022965	T539T
rs7688609	chr4	1807894	G	A	FCFR3	synonymous	HOM	NM_001163213	T653T

Annotation

- Ensembl, Variant Effect Predictor
 - On- and offline, different sources, many options

A survey of tools for variant analysis of next-generation genome sequencing data

Table 2
Variant annotation

Name	OS	Input	Output	SNP	INDEL	CNV	GUI	CLI	Web	Function/Location Parameters	DB IDs	Number of scores
ANNOVAR	Lin, Mac, Win, web interface	VCF, pileup, CompleteGenomics, GFF3- SOLID, SOAPsnp, MAQ, CASAVA	TXT	Yes	Yes	Yes	No	Yes	No	9 (func) + 11(exonic-func)	Yes	GERP++ conservation, LRT, MutationTaster, PhyloP conservation, PolyPhen, SIFT
AnnTools	Lin, Mac	VCF, pileup, TXT	VCF	Yes	Yes	Yes	No	Yes	No	5 (position) + 4 (functional class)	Yes	–
NGS-SNP	Lin, Mac	VCF, pileup, MAQ, diBayes, TXT	TXT	Yes	No	No	No	Yes	No	17	Yes	Condel, PolyPhen, SIFT
SeattleSeq	web interface	VCF, MAQ, CASAVA, GATK BED, custom	VCF, SeattleSeq	Yes	Yes	No	No	No	Yes	11(dbSNP) + 5 (GVS)	Yes	GERP, Grantham, phastCons, PolyPhen
snpEff	Lin, Mac, Win	VCF, pileup/TXT (deprecated)	VCF, TXT, HTML overview	Yes	Yes	No	No	Yes	No	34	Yes	–
SVA	Lin	VCF, SV.events file, BCO	CSV	Yes	Yes	Yes	Yes	Yes	No	17 (SNP), 17 (INDEL), 10 (CNV)	Yes	–
VARIANT	web interface	VCF, GFF2, BED	web report, TXT	Yes	Yes	No	No	Yes	Yes	26	Yes	–
VEP	Lin, web interface	VCF, pileup, HGVS, TXT, variant identifiers	TXT	Yes	Yes	No	No	Yes	Limited	28	Yes	Condel, PolyPhen, SIFT

Summary

- You have learned the basics to work in a Linux terminal
- You have gotten an idea of a small selection of NGS tools
- In the practical you will learn how to use and combine these tools
- This knowledge can be applied to install and use the tools of your choice

Michiel van Galen
Jeroen Laros