Advanced Bioinformatics (7BBG2016): Practical Bioinformatics Data Skills

Student ID: M2108139

1. Basic Linux and the command Line (20pts – 15% of final mark, each question provides 1 point)

1.1 What does ./../.. stand for?

A. Current directory

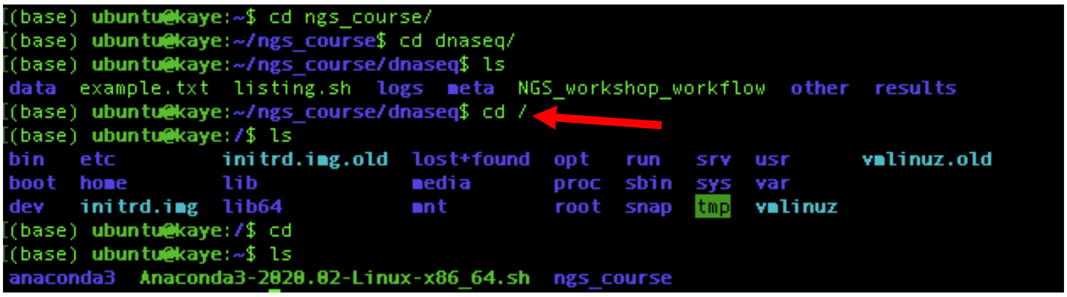
B. Up one directory

C. Up two directories ←

D. None of Above

1.2 What does cd / mean in UNIX? Please explain what the cd command does.

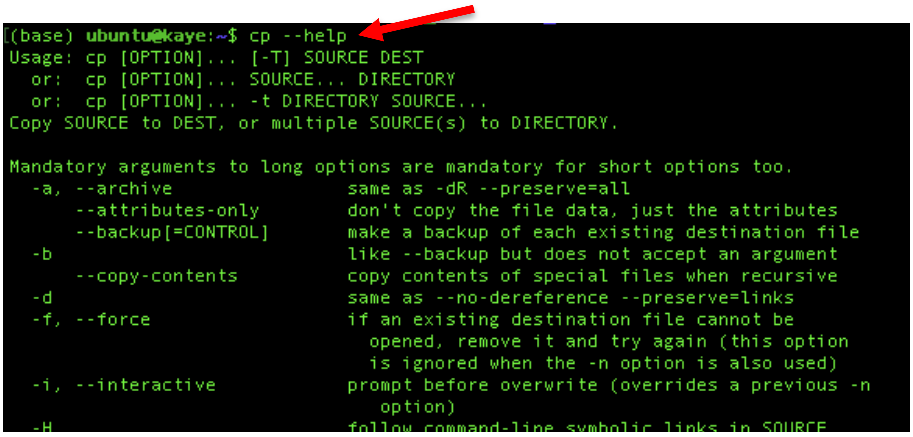
In UNIX, the cd command is used to change the current working directory, using either absolute or relative path names. Absolute path names start with the system root /, and the relative path begins at your current directory. The cd / command changes the current directory to the root directory. For example:



```
[(base) ubuntu@kaye:~$ cd ngs_course/
[(base) ubuntu@kaye:~/ngs_course$ cd dnaseq/
[(base) ubuntu@kaye:~/ngs_course/dnaseq$ ls
data  example.txt  listing.sh  logs  meta  NGS_workshop_workflow  other  results
[(base) ubuntu@kaye:~/ngs_course/dnaseq$ cd /
[(base) ubuntu@kaye:/$ ls
bin   etc          initrd.img.old  lost+found  opt   run   srv  usr      vmlinuz.old
boot  home         lib             media       proc  sbin  sys  var
dev   initrd.img   lib64           mnt         root  snap  tmp  vmlinuz
[(base) ubuntu@kaye:/$ cd
[(base) ubuntu@kaye:~$ ls
anaconda3  Anaconda3-2020.02-Linux-x86_64.sh  ngs_course
```

1.3 What command would you use to get help about the command cp? (please provide an example command)

The cp is used to copy files or directories from one place to the other.
To get help about command cp, the command cp -- help can be used

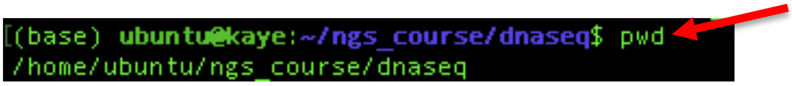


```
[(base) ubuntu@kaye:~$ cp --help
Usage: cp [OPTION]... [-T] SOURCE DEST
  or:  cp [OPTION]... SOURCE... DIRECTORY
  or:  cp [OPTION]... -t DIRECTORY SOURCE...
Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

Mandatory arguments to long options are mandatory for short options too.
  -a, --archive                same as -dR --preserve=all
      --attributes-only        don't copy the file data, just the attributes
      --backup[=CONTROL]       make a backup of each existing destination file
  -b                           like --backup but does not accept an argument
      --copy-contents          copy contents of special files when recursive
  -d                           same as --no-dereference --preserve=links
  -f, --force                  if an existing destination file cannot be
                                 opened, remove it and try again (this option
                                 is ignored when the -n option is also used)
  -i, --interactive            prompt before overwrite (overrides a previous -n
                                  option)
  -H                           follow command-line symbolic links in SOURCE
```

1.4 What does the command pwd do?

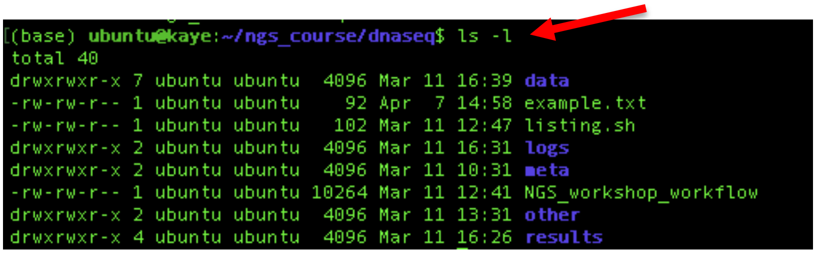
The command pwd stands for “print working directory’. This will show the complete path for the current working directory. For examples:



```
[(base) ubuntu@kaye:~/ngs_course/dnaseq$ pwd
/home/ubuntu/ngs_course/dnaseq
```

1.5 How do you display a listing of file details such as date, size, and access permissions in a given directory? (please providcd e an example command)

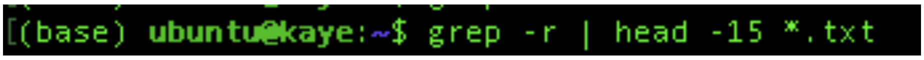
To display a listing of file details such as date, size, and access permissions in a given directory, the command ls -lh or ls -l can be used. For example:



```
[(base) ubuntu@kaye:~/ngs_course/dnaseq$ ls -l
total 40
drwxrwxr-x 7 ubuntu ubuntu  4096 Mar 11 16:39 data
-rw-rw-r-- 1 ubuntu ubuntu    92 Apr  7 14:58 example.txt
-rw-rw-r-- 1 ubuntu ubuntu   102 Mar 11 12:47 listing.sh
drwxrwxr-x 2 ubuntu ubuntu  4096 Mar 11 16:31 logs
drwxrwxr-x 2 ubuntu ubuntu  4096 Mar 11 10:31 meta
-rw-rw-r-- 1 ubuntu ubuntu 10264 Mar 11 12:41 NGS_workshop_workflow
drwxrwxr-x 2 ubuntu ubuntu  4096 Mar 11 13:31 other
drwxrwxr-x 4 ubuntu ubuntu  4096 Mar 11 16:26 results
```

1.6 How do you print on the terminal the first 15 lines of all files ending by .txt? (please provide an example command)

The grep -r (recursive) command and the head (head -15) command can be used for this. For example:

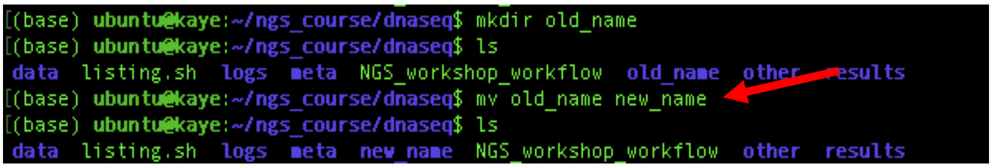


```
[(base) ubuntu@kaye:~$ grep -r | head -15 *.txt
```

*.txt means look at all files ending in .txt

1.7 How do you rename a file from new to old? (please provide an example command)

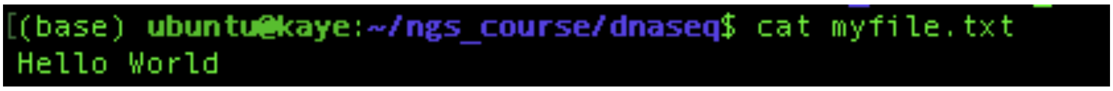
The mv command can be used to rename files. E.g. mv old_name new_name will move files from old_name into new_name while deleting old_name. Example command:



```
[(base) ubuntu@kaye:~/ngs_course/dnaseq$ mkdir old_name
[(base) ubuntu@kaye:~/ngs_course/dnaseq$ ls
 data  listing.sh  logs  meta  NGS_workshop_workflow  old_name  other  results
[(base) ubuntu@kaye:~/ngs_course/dnaseq$ mv old_name new_name
[(base) ubuntu@kaye:~/ngs_course/dnaseq$ ls
 data  listing.sh  logs  meta  new_name  NGS_workshop_workflow  other  results
```

1.8 How do you display the contents of a file myfile.txt? (please provide an example command)

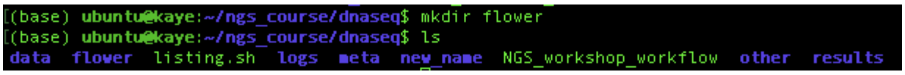
You can do this by using the command cat or less (cat myfile.txt). For example:



```
[(base) ubuntu@kaye:~/ngs_course/dnaseq$ cat myfile.txt
 Hello World
```

1.9 How do you create a new directory called flower? (please provide an example command)

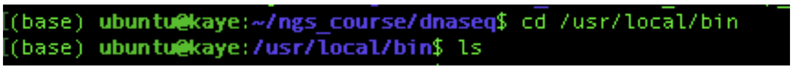
This can be done by using the command mkdir (mkdir flower). For example:



```
[(base) ubuntu@kaye:~/ngs_course/dnaseq$ mkdir flower
[(base) ubuntu@kaye:~/ngs_course/dnaseq$ ls
data  flower  listing.sh  logs  meta  new_name  NGS_workshop_workflow  other  results
```

1.10 How do you change the current directory to /usr/local/bin? (please provide an example command)

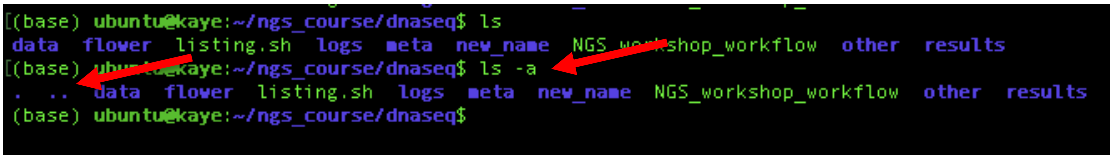
This can be achieved using the cd command. For example:



```
[(base) ubuntu@kaye:~/ngs_course/dnaseq$ cd /usr/local/bin
[(base) ubuntu@kaye:/usr/local/bin$ ls
```

1.11 How can you display a list of all files in the current directory, including the hidden files? (please provide an example command)

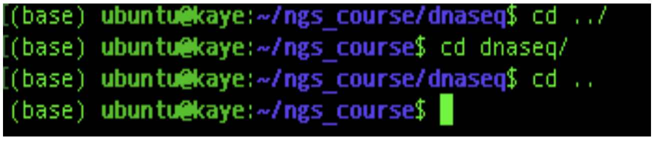
To do this use the ls -a command



```
[(base) ubuntu@kaye:~/ngs_course/dnaseq$ ls
 data  flower  listing.sh  logs  meta  new_name  NGS_workshop_workflow  other  results
[(base) ubuntu@kaye:~/ngs_course/dnaseq$ ls -a
 .  ..  data  flower  listing.sh  logs  meta  new_name  NGS_workshop_workflow  other  results
 (base) ubuntu@kaye:~/ngs_course/dnaseq$
```

1.12 What command do you have to use to go to the parent directory? (please provide an example command)

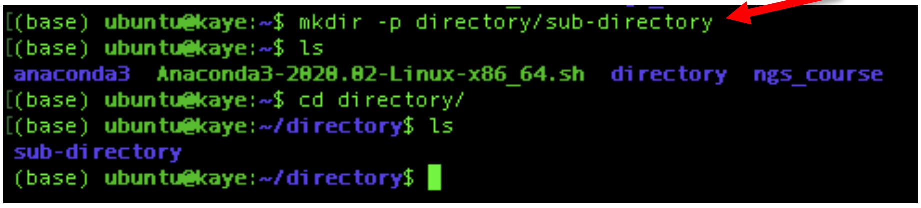
To go to the parent directory, use the command cd ../ or cd ..
For example:



```
[(base) ubuntu@kaye:~/ngs_course/dnaseq$ cd ../
[(base) ubuntu@kaye:~/ngs_course$ cd dnaseq/
[(base) ubuntu@kaye:~/ngs_course/dnaseq$ cd ..
 (base) ubuntu@kaye:~/ngs_course$ █
```

1.13 Which command would you use to create a sub-directory in your home directory? (please provide an example)

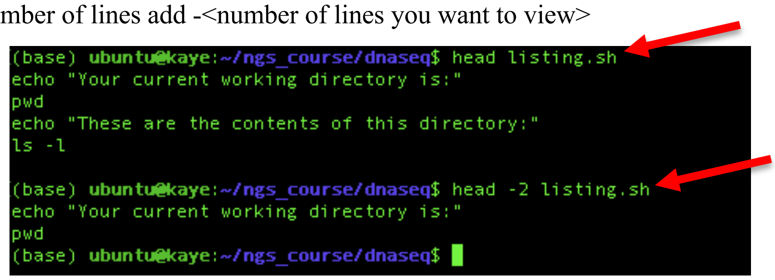
The command mkdir -p can be used. For example:



```
[(base) ubuntu@kaye:~$ mkdir -p directory/sub-directory
[(base) ubuntu@kaye:~$ ls
 anaconda3  Anaconda3-2020.02-Linux-x86_64.sh  directory  ngs_course
[(base) ubuntu@kaye:~$ cd directory/
[(base) ubuntu@kaye:~/directory$ ls
 sub-directory
 (base) ubuntu@kaye:~/directory$ █
```

1.14 Which command would you use to list the first lines in a text file? (please provide an example)

The command head can be used for this – e.g. head <filename>
For a specific number of lines add -<number of lines you want to view>
For example:



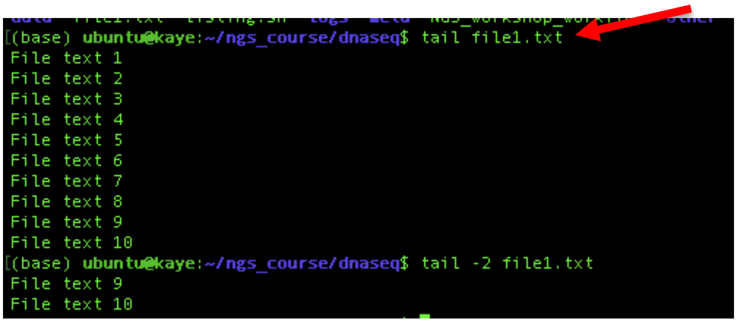
```
(base) ubuntu@kaye:~/ngs_course/dnaseq$ head listing.sh
echo "Your current working directory is:"
pwd
echo "These are the contents of this directory:"
ls -l

(base) ubuntu@kaye:~/ngs_course/dnaseq$ head -2 listing.sh
echo "Your current working directory is:"
pwd
(base) ubuntu@kaye:~/ngs_course/dnaseq$
```

1.15 Which command will display the last lines of the text file file1? (please provide an example)

The tail command can be used for this – e.g. tail <filename>
For a specific number of lines add -<number of lines you want to view>

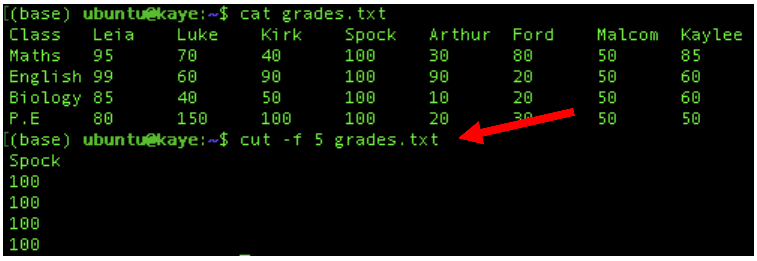
For example:



```
[(base) ubuntu@kaye:~/ngs_course/dnaseq$ tail file1.txt
 File text 1
 File text 2
 File text 3
 File text 4
 File text 5
 File text 6
 File text 7
 File text 8
 File text 9
 File text 10
[(base) ubuntu@kaye:~/ngs_course/dnaseq$ tail -2 file1.txt
 File text 9
 File text 10
```

1.16 Which command is used to extract a column from a text file? (please provide an example)

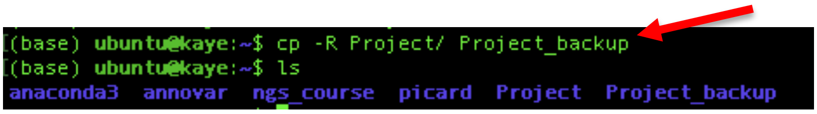
The command cut can be used, followed by the column of interest to be taken out (-f <column_number>). In the example below, the cut command gives us all the rows for column "5" (-f 5). These contain grades for "Spock" and prints these grades as a STDOUT.



```
[(base) ubuntu@kaye:~$ cat grades.txt
 Class   Leia    Luke    Kirk    Spock   Arthur  Ford    Malcom  Kaylee
 Maths   95      70      40      100     30      80      50      85
 English 99      60      90      100     90      20      50      60
 Biology 85      40      50      100     10      20      50      60
 P.E     80      150     100     100     20      30      50      50
[(base) ubuntu@kaye:~$ cut -f 5 grades.txt
 Spock
 100
 100
 100
 100
```

1.17 How do you copy an entire directory structure? E.g. from Project to Project.backup (please provide an example)

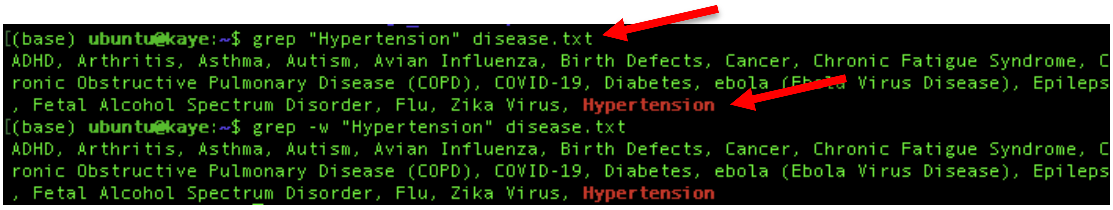
To copy an entire directory structure the copy file command can be used with the cop-R or -r option. Example:



```
(base) ubuntu@kaye:~$ cp -R Project/ Project_backup
(base) ubuntu@kaye:~$ ls
anaconda3  annovar  ngs_course  picard  Project  Project_backup
```

1.18 How would you search for the string Hypertension at the end of the line in a file called diseases.txt? (please provide an example)

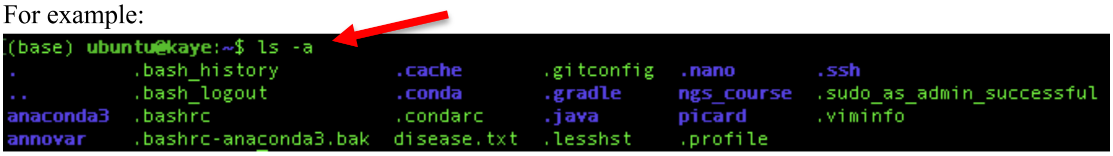
The grep command can be used, either alone or with -w to find the whole word.



```
(base) ubuntu@kaye:~$ grep "Hypertension" disease.txt
ADHD, Arthritis, Asthma, Autism, Avian Influenza, Birth Defects, Cancer, Chronic Fatigue Syndrome, C
ronic Obstructive Pulmonary Disease (COPD), COVID-19, Diabetes, ebola (Ebola Virus Disease), Epileps
, Fetal Alcohol Spectrum Disorder, Flu, Zika Virus, Hypertension
(base) ubuntu@kaye:~$ grep -w "Hypertension" disease.txt
ADHD, Arthritis, Asthma, Autism, Avian Influenza, Birth Defects, Cancer, Chronic Fatigue Syndrome, C
ronic Obstructive Pulmonary Disease (COPD), COVID-19, Diabetes, ebola (Ebola Virus Disease), Epileps
, Fetal Alcohol Spectrum Disorder, Flu, Zika Virus, Hypertension
```

1.19 How do you see hidden files in your home directory? (please provide an example)

To see hidden files in your home directory (cd ~), go to the home directory and use the ls command can be used with the -a flag which allows all files in a directory to be viewed.
For example:



```
(base) ubuntu@kaye:~$ ls -a
.           .bash_history          .cache       .gitconfig  .nano       .ssh
..          .bash_logout           .conda       .gradle     ngs_course  .sudo_as_admin_successful
anaconda3   .bashrc                .condarc     .java       picard      .viminfo
annovar     .bashrc-anaconda3.bak  disease.txt  .lesshst    .profile
```

1.20 How do you run a job that will continue running even if you are logged out? (please provide an example)

To do this the nohup command or screen tool can be used, followed by the Ctrl-a d command.

- The nohup command (which means no hang up) → In this case a job run with nohub will keep running until it finished even after you are logged out.
  For example: $ nohup ./script_to_be_run.sh
- Screen tool, open another screen that will run the script while you are logged out.
  For example:
  $ screen
  $ ./script_to_be_run

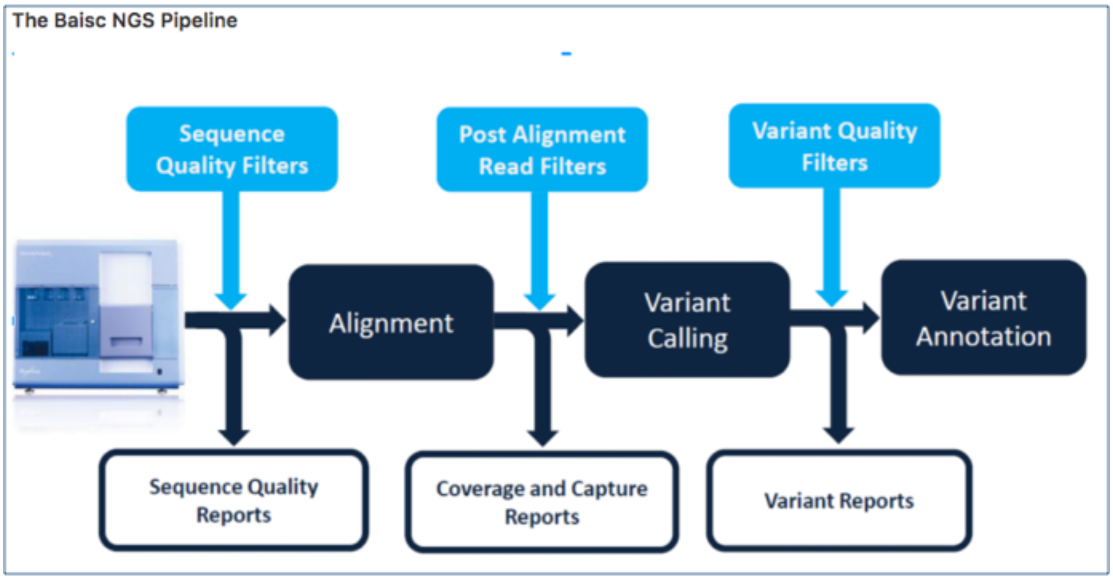
Next press Ctrl-a d, and you can log out while the script runs until it finishes. For example:
$ ./script_to_be_run (start script)
$ ctrl_z (pause program and go back to shell window)
$ bg (run job in background)
$ jobs (for rosalind this can be found in squeue -u followed by knumber – displays current jobs running)
$ disown -h %(job number/squeue number) → to disown job, so it will still be running

2. The NGS Pipeline (65pts – 45% of final mark)

2.0 From raw data to alignment and variant calls (20pts)

The assessment is designed to:

- Test your ability to run standard NGS pipeline using the command line on a Linux system.
- Test your ability to create a Bash script that executes your NGS pipeline
- Test your basic knowledge of a standard NGS pipeline.



You have been provided with paired end fastq data and an annotation bed file from an Illumina HiSeq 2500 run. Using the assigned Openstack instance (please contact the module leaders if you have any problems with your Openstack instance), install the necessary tools and execute a standard Bioinformatics NGS pipeline to perform read alignment, variant discovery and annotation as described in the following NGS Pipeline section. **You are required to share a bash script that runs the workflow and takes the provided sequencing data as input (links provided below) with the examiner by uploading it with this report.** Please make sure the bash script lines are adequately commented to provide a clear description of what it is doing. **The script will be evaluated by the examiner and up to 20pts will be given for a fully running and easy to read script.** Based on your pipeline, provide the following information and answer each question.

Fastq Read 1 (~750MB): https://s3-eu-west-1.amazonaws.com/workshopdata2017/NGS0001.R1.fastq.qz

Fastq Read 2 (~750MB):https://s3-eu-west-1.amazonaws.com/workshopdata2017/NGS0001.R2.fastq.qz

Annotation File (10M): https://s3-eu-west-1.amazonaws.com/workshopdata2017/annotation.bed

2.1 Install the tools and dependencies of your pipeline (using Miniconda when possible) and Download the input files (10 pts)

1. **List the command lines to install all dependencies necessary to run the pipeline (3 pts)**

# Anaconda was installed locally to OpenStack instance (ubuntu@10.200.111.236) and used to install Trimmomatic, Fastqc, Samtools, Picard, Bedtools, BWA, Freebayes, and vcflib

# Go to home directory

```
$ cd ~/
```

# Download Anaconda using wget command

```
$ wget https://repo.anaconda.com/archive/Anaconda3-2020.02-Linux-x86_64.sh
```

# Make executable

```
$ chmod +x ./Anaconda3-2020.02-Linux-x86_64.sh
```

# Run the Anaconda script using bash command

```
$ bash ./Anaconda3-2020.02-Linux-x86_64.sh
$ source ~/.bashrc
```

# Configure aspects of conda

```
$ conda config --add channels defaults
$ conda config --add channels bioconda
$ conda config --add channels conda-forge
```

# Install the tools needed in this assignment

```
$ conda install samtools
$ conda install bwa
$ conda install freebayes
$ conda install picard
$ conda install bedtools
$ conda install trimmomatic
$ conda install fastqc
$ conda install vcflib
```

2. **List all command lines necessary to download the input files (e.g. fastqs, reference genomes, etc) (2 pts)**

# The project was organised prior to downloading the input files. A directory called ngs_course was made in the home directory, containing the sub-directory dna_seq_assignment

```
$ cd ~/
$ mkdir ngs_course
$ mkdir ngs_course/dna_seq_assignment
```

```
# Four directories were made within dna_seq_assignment to keep files organised.
$ cd ngs_course/dna_seq_assignment
$ mkdir data meta results logs
```

```
# Subdirectories were created in the data directory for trimmed and untrimmed reads.
$ cd ~/ngs_course/dna_seq_assignment/data
$ mkdir untrimmed_fastq
$ mkdir trimmed_fastq
```

```
# Assignment fastq files (Fastq Read 1 (approx. 750MB) +Fastq Read 2 (approx.. 750MB)) were downloaded
$ cd ~/ngs_course/dna_seq_assignment/data/untrimmed_fastq
$ wget https://s3-eu-west-1.amazonaws.com/workshopdata2017/NGS0001.R1.fastq.qz
$ wget https://s3-eu-west-1.amazonaws.com/workshopdata2017/NGS0001.R2.fastq.qz
```

```
# Files were checked to have been downloaded
$ cd ~/ngs_course/dna_seq_assignment/data/untrimmed_fastq
$ ls -lF
```

```
# Annotation.bed file provided in assignment was downloaded
$ cd ~/ngs_course/dna_seq_assignment/data
$ wget https://s3-eu-west-1.amazonaws.com/workshopdata2017/annotation.bed
```

```
# Reference file to map against data in alignment step was downloaded. (hg19.fa.gz)
$ cd ~/ngs_course/dna_seq_assignment/data
$ wget http://hgdownload.cse.ucsc.edu/goldenPath/hg19/bigZips/hg19.fa.gz
```

Implement and run the following NGS Pipeline (please provide the command lines to run the following steps of your pipeline and comment/explain the choice of options):

2.2. Pre-Alignment QC (4 pts)

Perform quality assessment and trimming (2pt)

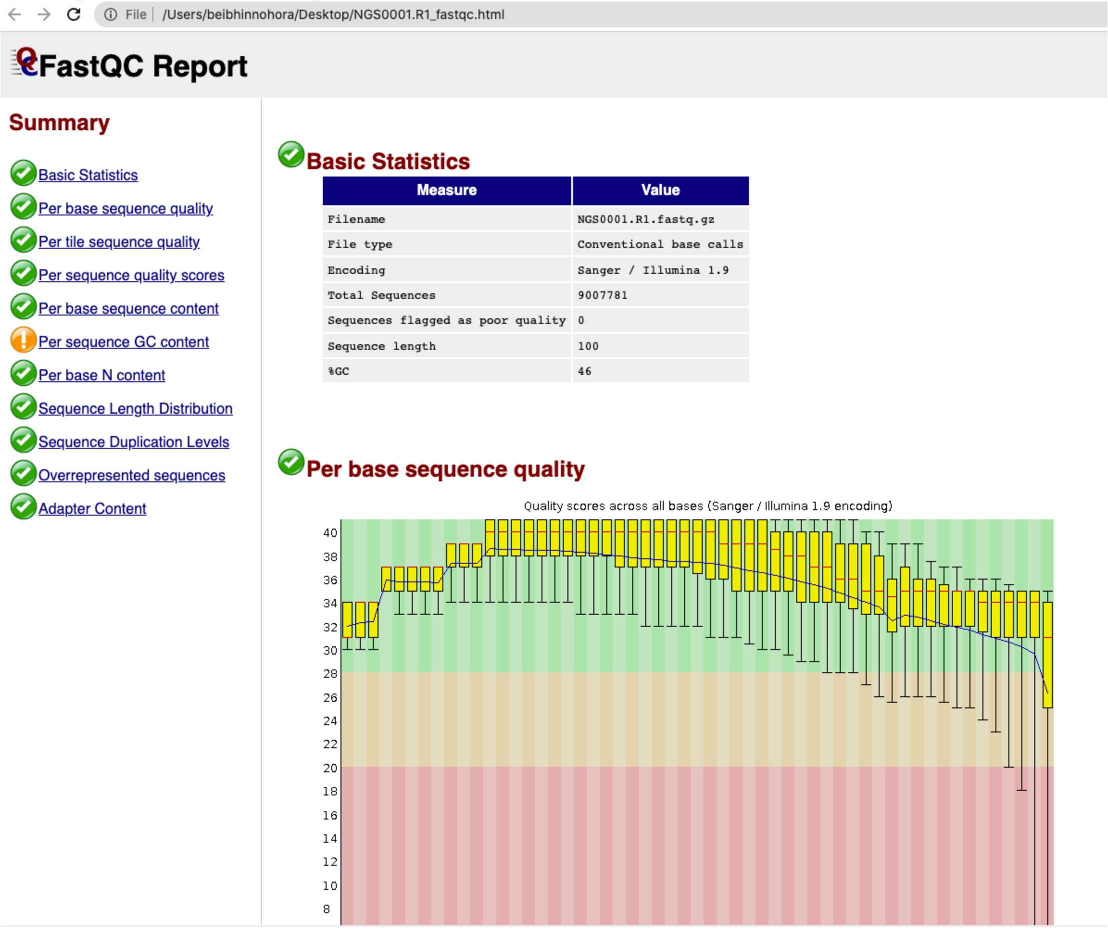
```
# Quality control of the untrimmed FASTQ files is important address the quality of the data and perform any quality control metrics necessary to improve data quality e.g. trimming. FASTQC can be used to do some quality control checks on raw sequencing data from high throughput sequencing. It gives a summary of whether the data has any problems that needs to be addressed in the form of summary graphics and tables.
```

```
# FastQC was run to do quality control checks on raw sequencing data (fix mistake that files ended in .qz instead of  .gz).
$ cd ~/ngs_course/dna_seq_assignment/data/untrimmed_fastq
$ mv NGS0001.R1.fastq.qz NGS0001.R1.fastq.gz
$ mv NGS0001.R2.fastq.qz NGS0001.R2.fastq.gz
$ fastqc *.fastq.gz
```

```
# Move FastQC results to results directory
$ mkdir ~/ngs_course/dna_seq_assignment/results/fastqc_untrimmed_reads
$ cd ~/ngs_course/dna_seq_assignment/data/untrimmed_fastq
$ mv *fastqc* ~/ngs_course/dna_seq_assignment/results/fastqc_untrimmed_reads/
```

```
# View details of files generated by FastQC:
$ ls -lh ~/ngs_course/dna_seq_assignment/results/fastqc_untrimmed_reads/
```

# html files produced by FastQC can be viewed locally on desktop using FileZilla. To view summaries on the command line of FastQC data the following steps were taken. Example of FASTQC report:



FastQC Report

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	NGS0001.R1.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	9007781
Sequences flagged as poor quality	0
Sequence length	100
%GC	46

✓ 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
10
8

```
# Unzip FastQC zip files and view what information is contained in them.
$ for zip in *.zip; do unzip $zip; done
$ cd ~/ngs_course/dna_seq_assignment/results/fastqc_untrimmed_reads/NGS0001.R1_fastqc
$ ls -lh
$ head summary.txt
$ cd ~/ngs_course/dna_seq_assignment/results/fastqc_untrimmed_reads/NGS0001.R2_fastqc
$ ls -lh
$ head summary.txt
```

```
# Save record of FastQC summaries
$ cd ~/ngs_course/dna_seq_assignment/results/fastqc_untrimmed_reads
$ cat */summary.txt > ~/ngs_course/dna_seq_assignment/logs/fastqc_untrimmed_summaries.txt
```

```
# Improve quality of reads by trimming adapter and filtering out poor quality read scores using Trimmomatic. Trimmomatic is a java based program that can remove sequence specific reads and nucleotides that fall below a certain threshold.
```

```
# Go to untrimmed fastq data location and run Trimmomatic
$ cd ~/ngs_course/dna_seq_assignment/data/untrimmed_fastq
$ trimmomatic PE  \
  -threads 4 \
  -phred33 \

/home/ubuntu/ngs_course/dna_seq_assignment/data/untrimmed_fastq/NGS0001.R1.fastq.gz
/home/ubuntu/ngs_course/dna_seq_assignment/data/untrimmed_fastq/NGS0001.R2.fastq.gz \
  -baseout
/home/ubuntu/ngs_course/dna_seq_assignment/data/trimmed_fastq/NGS0001_trimmed_R \
  ILLUMINACLIP:/home/ubuntu/anaconda3/pkgs/trimmomatic-0.39-hdfd78af_2/share/trimmomatic-0.39-2/adapters/NexteraPE-PE.fa:2:30:10 \
  TRAILING:25 MINLEN:50
```

```
# View trimmed data in trimmed directory
$ cd ~/ngs_course/dna_seq_assignment/data/trimmed_fastq
$ ls -lh
```

Perform basic quality assessment of paired trimmed sequencing data (2pt)

```
##Run FastQC on trimmed paired data.
$ cd ~/ngs_course/dna_seq_assignment/data/trimmed_fastq
$ fastqc NGS0001_trimmed_R_1P
$ fastqc NGS0001_trimmed_R_2P
```

```
## Move FastQC results on paired trimmed data to results directory
$ mkdir ~/ngs_course/dna_seq_assignment/results/fastqc_trimmed_reads
$ cd ~/ngs_course/dna_seq_assignment/data/trimmed_fastq
$ mv *fastqc* ~/ngs_course/dna_seq_assignment/results/fastqc_trimmed_reads/
```

## View details of files generated by FastQC:

```
$ ls -lh ~/ngs_course/dna_seq_assignment/results/fastqc_trimmed_reads/
```

## html files produced by FastQC for paired trimmed reads can be viewed locally on desktop using FileZilla. To view summaries on the command line of FastQC data the following steps were taken

# Unzip FastQC zip files and view what information is contained in them.

```
$ for zip in *.zip; do unzip $zip; done
$ cd ~/ngs_course/dna_seq_assignment/results/fastqc_trimmed_reads/NGS0001_trimmed_R_1P_fastqc
$ ls -lh
$ head summary.txt
$ cd ~/ngs_course/dna_seq_assignment/results/fastqc_trimmed_reads/NGS0001_trimmed_R_2P_fastqc
$ ls -lh
$ head summary.txt
```

# Save record of FastQC summaries

```
$ cd ~/ngs_course/dna_seq_assignment/results/fastqc_trimmed_reads
$ cat */summary.txt > ~/ngs_course/dna_seq_assignment/logs/fastqc_trimmed_summaries.txt
```

2.3. Alignment (17pts)

Align the paired trimmed fastq files using bwa mem and reference genome hg19 (edit your bwa mem step to include read group information in your BAM file) (9pts)

# A folder was created for the reference and its index files and then bwa was run to generate the index files

```
$ mkdir -p ~/ngs_course/dna_seq_assignment/data/reference
$ mv ~/ngs_course/dna_seq_assignment/data/hg19.fa.gz ~/ngs_course/dna_seq_assignment/data/reference/
$ bwa index ~/ngs_course/dna_seq_assignment/data/reference/hg19.fa.gz
$ ls ~/ngs_course/dna_seq_assignment/data/reference
```

##The following read group info will be used for the alignment: Read group identifier (ID): HWI-D0011.50.H7AP8ADXX.1.NGS0001 -Read group identifier (SM): NGS0001 -Read group identifier (PL): ILLUMINA -Read group identifier (LB): nextera-NGS001-blood -Read group identifier (PU): HWI-D00119 -Read group identifier (DT): 2022-03-23

```
# The directory aligned_data was made
$ mkdir ~/ngs_course/dna_seq_assignment/data/aligned_data
```

```
# BWA MEM was ran with the read group information
$ bwa mem -t 4 -v 1 -R '@RG\tID:HWI-
D0011.50.H7AP8ADXX.1.NGS0001\tSM:NGS0001\tPL:ILLUMINA\tLB:nextera-
NGS0001-blood\tDT:2022-03-23\tPU:HWI-D00119' -I 250,50
~/ngs_course/dna_seq_assignment/data/reference/hg19.fa.gz
~/ngs_course/dna_seq_assignment/data/trimmed_fastq/NGS0001_trimmed_R_1P
~/ngs_course/dna_seq_assignment/data/trimmed_fastq/NGS0001_trimmed_R_2P >
~/ngs_course/dna_seq_assignment/data/aligned_data/NGS0001.sam
```

```
# Change directories to aligned_data folder
$ cd ~/ngs_course/dna_seq_assignment/data/aligned_data
```

```
# Convert the sam file into bam format, sort it and generate an index using samtools
$ samtools view -h -b NGS0001.sam > NGS0001.bam
$ samtools sort NGS0001.bam > NGS0001_sorted.bam
```

```
# Note: Sam format is a text format that stores the sequence data in tab delimited
ASCII columns. BAM format stored the same data in a compressed, indexed, binary
form
```

```
# At this point I Ran out of space to move forward so I deleted NGS0001.sam file to
clear space. If I needed this file again at any point it can be regenerated by running
BWA MEME with the read group information.
$ rm NGS0001.sam
```

```
# Generate a .bai index file
$ samtools index NGS0001_sorted.bam
$ ls
```

Perform duplicate marking (2pts)

```
# Picard tools was used to mark duplicates. Picard tools examines aligned records in
the .bam dataset to locate duplicate molecules. All records are then written to the
output file with the duplicate records flagged.
# samtools was used to index the sort_marked bam file
```

```
$ picard MarkDuplicates I=NGS0001_sorted.bam O=NGS0001_sorted_marked.bam
M=marked_dup_metrics.txt
$ samtools index NGS0001_sorted_marked.bam
```

Quality Filter the duplicate marked BAM file (2pts)

# The NGS0001_sorted_marked.bam was filtered based on mapping quality and bitwise flags using samtools

# Reads were be filtered according to the following criteria: Minimum MAPQ quality score : 20 -Filter on bitwise flag: yes a. Skip alignments with any of these flag bits set i. The read is unmapped ii. The alignment or this read is not primary iii. The read fails platform/vendor quality checks iv. The read is a PCR or optical duplicate.

```
$ samtools view -F 1796  -q 20 -o NGS0001_sorted_filtered.bam
NGS0001_sorted_marked.bam
$ samtools index NGS0001_sorted_filtered.bam
```

# The BAM files were viewed using samtools view sample.bam | head

```
$ samtools view NGS0001_sorted_filtered.bam | head
```

Generate standard alignment statistics (i.e. flagstats, idxstats, depth of coverage, insert size) (4pts)

# The alignment statistics analysis

# **Task 1:** Samtools flagstat was used to calculate and print statistics if NGS001_sorted_filtered.bam

```
$ samtools flagstat NGS0001_sorted_filtered.bam > flagstat_output.txt
$ mv /home/ubuntu/ngs_course/dna_seq_assignment/data/aligned_data/flagstat_output.txt
/home/ubuntu/ngs_course/dna_seq_assignment/results
```

# **Task 2:** Samtools idxstats was used to generate alignment statistics per chromosome

```
$ samtools idxstats NGS0001_sorted_filtered.bam > idxstats_output.txt
$ mv /home/ubuntu/ngs_course/dna_seq_assignment/data/aligned_data/idxstats_output.txt
/home/ubuntu/ngs_course/dna_seq_assignment/results
```

# **Task 3:** Picard insert_size_metrics was used to determine the distribution of insert sizes. To use picard Java is required.

# Check Java version

```
$ cd
$ java -version
```

# Downloading and using Picard insert_size_metrics

```
$ git clone https://github.com/broadinstitute/picard.git

$ ./gradlew shadowJar
$ java -jar build/libs/picard.jar

$ java -jar /home/ubuntu/picard/build/libs/picard.jar CollectInsertSizeMetrics \
     I= NGS0001_sorted_filtered.bam \
     O=insert_size_metrics.txt \
     H=insert_size_histogram.pdf \
     M=0.5
```

```
# Results were viewed and moved to results folder
$ less insert_size_metrics.txt
$ mv
/home/ubuntu/ngs_course/dna_seq_assignment/data/aligned_data/insert_size_metrics.txt
/home/ubuntu/ngs_course/dna_seq_assignment/results
```

# Task 4: Depth of Coverage was determined using bedtools

```
# Calculate depth of coverage for all regions in the .bam file using bedtools
$ bedtools genomecov -ibam NGS0001_sorted_filtered.bam -bga -split >
CoverageTotal.bedgraph.txt
```

```
# Move results to results folder and view data
mv
/home/ubuntu/ngs_course/dna_seq_assignment/data/aligned_data/CoverageTotal.bedgraph
/home/ubuntu/ngs_course/dna_seq_assignment/results
```

2.4. Variant Calling (4pts)

Call Variants using Freebayes restricting the analysis to the regions in the bed file provided (2pt)

```
# Freebayes is a Bayesian genetic variant detector that uses short-read alignments for
any number of individuaks from a population and a reference genome to determine
the most-likely combination of genotypes for a population at each position in the
reference. It produces a variant call file (VCF)
```

```
# hg19.fa.gz file was unzipped o it can be indexed
$ zcat ~/ngs_course/dna_seq_assignment/data/reference/hg19.fa.gz >
~/ngs_course/dna_seq_assignment/data/reference/hg19.fa
```

```
# A .fai index file for FASTA files was produced
$ samtools faidx ~/ngs_course/dna_seq_assignment/data/reference/hg19.fa
```

```
# Freebayes was used to produce the VCF file
$ freebayes --bam
/home/ubuntu/ngs_course/dna_seq_assignment/data/aligned_data/NGS0001_sorted_filtered.b
am --fasta-reference /home/ubuntu/lsngs_course/dna_seq_assignment/data/reference/hg19.fa -
-vcf /home/ubuntu/ngs_course/dna_seq_assignment/results/NGS0001.vcf
```

```
# The NGS0001.vcf file was zipped
$ bgzip /home/ubuntu/ngs_course/dna_seq_assignment/results/NGS0001.vcf
```

```
# The VCF was indexed with tabix
$ tabix -p vcf /home/ubuntu/ngs_course/dna_seq_assignment/results/NGS0001.vcf.gz
```

Quality Filter Variants using your choice of filters (2pt)

```
# The freebayes hard filter for human diploid sequencing was applied using vcffilter:
QUAL > 1: removes horrible sites QUAL / AO > 10 : additional contribution of each
obs should be 10 log units (~ Q10 per read) SAF > 0 & SAR > 0 : reads on both
strands RPR > 1 & RPL > 1 : at least two reads “balanced” to each side of the site
$ vcffilter -f "QUAL > 1 & QUAL / AO > 10 & SAF > 0 & SAR > 0 & RPR > 1 &
RPL > 1" /home/ubuntu/ngs_course/dna_seq_assignment/results/NGS0001.vcf.gz >
/home/ubuntu/ngs_course/dna_seq_assignment/results/NGS0001_filtered.vcf
```

```
# The vcf file was filtered using bedtools for the regions in annotation.bed file
provided in this assignment
$ bedtools intersect -header -wa -a
/home/ubuntu/ngs_course/dna_seq_assignment/results/NGS0001_filtered.vcf -b
/home/ubuntu/ngs_course/dna_seq_assignment/data/annotation.bed
          >
/home/ubuntu/ngs_course/dna_seq_assignment/results/NGS0001_filtered_annotation.
vcf
```

```
# The file was zipped
$ bgzip
/home/ubuntu/ngs_course/dna_seq_assignment/results/NGS0001_filtered_annotation.vcf
```

```
# The file was indexed with tabix
$ tabix -p vcf
/home/ubuntu/ngs_course/dna_seq_assignment/results/NGS0001_filtered_annotation.vcf.gz
```

2.5. Variant Annotation and Prioritization (10pts)

Annotate variants using ANNOVAR (4pt) and snpEFF (4pt)

```
# ANNOVAR is used to annotate variants with respect to genes, databases of normal
variantion and pathogenicity predictors. Variant filters can be applied to the VCF file
in excel to generate a list of candidate genes.
```

```
# ANNOVAR was downloaded from Kai Wang (Wang K, Li M, Hakonarson H.
ANNOVAR: Functional annotation of genetic variants from next-generation
sequencing data, Nucleic Acids Research, 38:e164, 2010).
```

```
# Fillezilla was used to put ‘annovar.latest.tar.gz’ file onto openstack /home/ubuntu
$ tar -zxvf annovar.latest.tar.gz
```

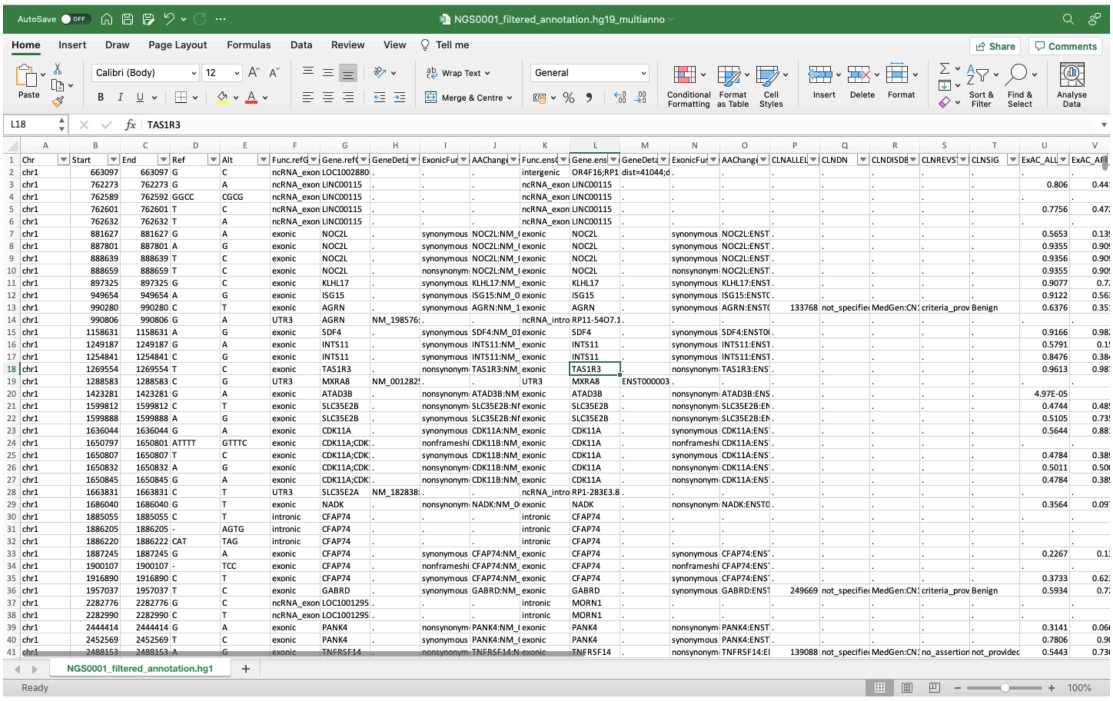
```
# Annovar databases that are used for annotation were downloaded
$ chmod +x annotate_variation.pl
$ cd annovar
$ ./annotate_variation.pl -buildver hg19 -downdb -webfrom annovar knownGene humandb/
$ ./annotate_variation.pl -buildver hg19 -downdb -webfrom annovar refGene humandb/
$ ./annotate_variation.pl -buildver hg19 -downdb -webfrom annovar ensGene humandb/
$ ./annotate_variation.pl -buildver hg19 -downdb -webfrom annovar clinvar_20180603 humandb/
$ ./annotate_variation.pl -buildver hg19 -downdb -webfrom annovar exac03 humandb/
$ ./annotate_variation.pl -buildver hg19 -downdb -webfrom annovar dbnsfp31a_interpro humandb/
```

```
# VCF was converted to Annovar input format
$ ./convert2annovar.pl -format vcf4 /home/ubuntu/ngs_course/dna_seq_assignment/results/NGS0001_filtered_annotation.vcf.gz > /home/ubuntu/ngs_course/dna_seq_assignment/results/NGS0001_filtered_annotation.avinput
```

```
# Annovar table function was run to produce an csv output
$ ./table_annovar.pl /home/ubuntu/ngs_course/dna_seq_assignment/results/NGS0001_filtered_annotation.avinput humandb/ -buildver hg19 -out /home/ubuntu/ngs_course/dna_seq_assignment/results/NGS0001_filtered_annotation -remove -protocol refGene,ensGene,clinvar_20180603,exac03,dbnsfp31a_interpro -operation g,g,f,f,f -otherinfo -nastring . -csvout
```

```
# The output will be in CSV format. A comma-separated values (CSV) file is a delimited text file that uses a comma to separate values.

# This CSV file was downloaded via FileZilla and opened with Office Excel to view data as shown in next page.
```



L18 TAS1R3

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	Chr	Start	End	Ref	Alt	Func.refG	Gene.ref(	GeneDeta	ExonicFur	AAChange	Func.ens(	Gene.ens	GeneDeta	ExonicFur	AAChange	CLNALLEL	CLNDN	CLNDISDB	CLNREVS	CLNSIG	ExAC_ALL	ExAC_AF
2	chr1	663097	663097	G	C	ncRNA_exon	LOC1002880	.	.	.	intergenic	OR4F16;RP1	dist=41044;d	.	.	.	.	.	.	.	.	.
3	chr1	762273	762273	G	A	ncRNA_exon	LINC00115	.	.	.	ncRNA_exon	LINC00115	.	.	.	.	.	.	.	.	0.806	0.44
4	chr1	762589	762592	GGCC	CGCG	ncRNA_exon	LINC00115	.	.	.	ncRNA_exon	LINC00115	.	.	.	.	.	.	.	.	.	.
5	chr1	762601	762601	T	C	ncRNA_exon	LINC00115	.	.	.	ncRNA_exon	LINC00115	.	.	.	.	.	.	.	.	0.7756	0.47
6	chr1	762632	762632	T	A	ncRNA_exon	LINC00115	.	.	.	ncRNA_exon	LINC00115	.	.	.	.	.	.	.	.	.	.
7	chr1	881627	881627	G	A	exonic	NOC2L	.	synonymous	NOC2L:NM_(	exonic	NOC2L	.	synonymous	NOC2L:ENST	.	.	.	.	.	0.5653	0.13
8	chr1	887801	887801	A	G	exonic	NOC2L	.	synonymous	NOC2L:NM_(	exonic	NOC2L	.	synonymous	NOC2L:ENST	.	.	.	.	.	0.9355	0.90
9	chr1	888639	888639	T	C	exonic	NOC2L	.	synonymous	NOC2L:NM_(	exonic	NOC2L	.	synonymous	NOC2L:ENST	.	.	.	.	.	0.9356	0.90
10	chr1	888659	888659	T	C	exonic	NOC2L	.	nonsynonym	NOC2L:NM_(	exonic	NOC2L	.	nonsynonym	NOC2L:ENST	.	.	.	.	.	0.9355	0.90
11	chr1	897325	897325	G	C	exonic	KLHL17	.	synonymous	KLHL17:NM_	exonic	KLHL17	.	synonymous	KLHL17:ENST	.	.	.	.	.	0.9077	0.7
12	chr1	949654	949654	A	G	exonic	ISG15	.	synonymous	ISG15:NM_0	exonic	ISG15	.	synonymous	ISG15:ENST(	.	.	.	.	.	0.9122	0.56
13	chr1	990280	990280	C	T	exonic	AGRN	.	synonymous	AGRN:NM_1	exonic	AGRN	.	synonymous	AGRN:ENST(	133768	not_specifie	MedGen:CN1	criteria_prov	Benign	0.6376	0.35
14	chr1	990806	990806	G	A	UTR3	AGRN	NM_198576:	.	.	ncRNA_intro	RP11-54O7.1	.	.	.	.	.	.	.	.	.	.
15	chr1	1158631	1158631	A	G	exonic	SDF4	.	synonymous	SDF4:NM_01	exonic	SDF4	.	synonymous	SDF4:ENST0(	.	.	.	.	.	0.9166	0.98
16	chr1	1249187	1249187	G	A	exonic	INTS11	.	synonymous	INTS11:NM_	exonic	INTS11	.	synonymous	INTS11:ENST	.	.	.	.	.	0.5791	0.1
17	chr1	1254841	1254841	C	G	exonic	INTS11	.	synonymous	INTS11:NM_	exonic	INTS11	.	synonymous	INTS11:ENST	.	.	.	.	.	0.8476	0.38
18	chr1	1269554	1269554	T	C	exonic	TAS1R3	.	nonsynonym	TAS1R3:NM_	exonic	TAS1R3	.	nonsynonym	TAS1R3:ENS	.	.	.	.	.	0.9613	0.98
19	chr1	1288583	1288583	C	G	UTR3	MXRA8	NM_001282	.	.	UTR3	MXRA8	ENST000003	.	.	.	.	.	.	.	.	.
20	chr1	1423281	1423281	G	A	exonic	ATAD3B	.	nonsynonym	ATAD3B:NM_	exonic	ATAD3B	.	nonsynonym	ATAD3B:ENS	.	.	.	.	.	4.97E-05	
21	chr1	1599812	1599812	C	T	exonic	SLC35E2B	.	nonsynonym	SLC35E2B:NM	exonic	SLC35E2B	.	nonsynonym	SLC35E2B:EN	.	.	.	.	.	0.4744	0.48
22	chr1	1599888	1599888	A	G	exonic	SLC35E2B	.	synonymous	SLC35E2B:NM	exonic	SLC35E2B	.	nonsynonym	SLC35E2B:EN	.	.	.	.	.	0.5105	0.73
23	chr1	1636044	1636044	G	A	exonic	CDK11A	.	synonymous	CDK11A:NM_	exonic	CDK11A	.	synonymous	CDK11A:ENS	.	.	.	.	.	0.5644	0.88
24	chr1	1650797	1650801	ATTTT	GTTTC	exonic	CDK11A;CDK:	.	nonframeshi	CDK11B:NM_	exonic	CDK11A	.	nonframeshi	CDK11A:ENS	.	.	.	.	.	.	.
25	chr1	1650807	1650807	T	C	exonic	CDK11A;CDK:	.	synonymous	CDK11B:NM_	exonic	CDK11A	.	synonymous	CDK11A:ENS	.	.	.	.	.	0.4784	0.38
26	chr1	1650832	1650832	A	G	exonic	CDK11A;CDK:	.	nonsynonym	CDK11B:NM_	exonic	CDK11A	.	nonsynonym	CDK11A:ENS	.	.	.	.	.	0.5011	0.50
27	chr1	1650845	1650845	G	A	exonic	CDK11A;CDK:	.	nonsynonym	CDK11B:NM_	exonic	CDK11A	.	nonsynonym	CDK11A:ENS	.	.	.	.	.	0.4784	0.38
28	chr1	1663831	1663831	C	T	UTR3	SLC35E2A	NM_182838:	.	.	ncRNA_intro	RP1-283E3.8	.	.	.	.	.	.	.	.	.	.
29	chr1	1686040	1686040	G	T	exonic	NADK	.	nonsynonym	NADK:NM_0	exonic	NADK	.	nonsynonym	NADK:ENST0	.	.	.	.	.	0.3564	0.09
30	chr1	1885055	1885055	C	T	intronic	CFAP74	.	.	.	intronic	CFAP74	.	.	.	.	.	.	.	.	.	.
31	chr1	1886205	1886205	-	AGTG	intronic	CFAP74	.	.	.	intronic	CFAP74	.	.	.	.	.	.	.	.	.	.
32	chr1	1886220	1886222	CAT	TAG	intronic	CFAP74	.	.	.	intronic	CFAP74	.	.	.	.	.	.	.	.	.	.
33	chr1	1887245	1887245	G	A	exonic	CFAP74	.	synonymous	CFAP74:NM_	exonic	CFAP74	.	synonymous	CFAP74:ENS	.	.	.	.	.	0.2267	0.1
34	chr1	1900107	1900107	-	TCC	exonic	CFAP74	.	nonframeshi	CFAP74:NM_	exonic	CFAP74	.	nonframeshi	CFAP74:ENS	.	.	.	.	.	.	.
35	chr1	1916890	1916890	C	T	exonic	CFAP74	.	synonymous	CFAP74:NM_	exonic	CFAP74	.	synonymous	CFAP74:ENS	.	.	.	.	.	0.3733	0.62
36	chr1	1957037	1957037	T	C	exonic	GABRD	.	synonymous	GABRD:NM_	exonic	GABRD	.	synonymous	GABRD:ENST	249669	not_specifie	MedGen:CN1	criteria_prov	Benign	0.5934	0.7
37	chr1	2282776	2282776	G	C	ncRNA_exon	LOC1001295	.	.	.	intronic	MORN1	.	.	.	.	.	.	.	.	.	.
38	chr1	2282990	2282990	C	T	ncRNA_exon	LOC1001295	.	.	.	intronic	MORN1	.	.	.	.	.	.	.	.	.	.
39	chr1	2444414	2444414	G	A	exonic	PANK4	.	nonsynonym	PANK4:NM_(	exonic	PANK4	.	nonsynonym	PANK4:ENST	.	.	.	.	.	0.3141	0.06
40	chr1	2452569	2452569	T	C	exonic	PANK4	.	synonymous	PANK4:NM_(	exonic	PANK4	.	synonymous	PANK4:ENST	.	.	.	.	.	0.7806	0.9
41	chr1	2488153	2488153	A	G	exonic	TNFRSF14	.	nonsynonym	TNFRSF14:N	exonic	TNFRSF14	.	nonsynonym	TNFRSF14:EI	139088	not_specifie	MedGen:CN1	no_assertion	not_providec	0.5443	0.73

# snpEFF is annotation tool that annotates variants based on their genomic locations and predicts coding effects.

# snpEFF was downloaded

```
$ cd ~/
$ wget http://sourceforge.net/projects/snpeff/files/snpEff_latest_core.zip
```

# The file was unzipped

```
$ unzip snpEff_latest_core.zip
```

# Download database of interest – hg19

```
$ cd /home/ubuntu/snpEff/
$ java -jar snpEff.jar build -refSeq -v hg19
```

# Download the pre-built human database (GRCh37.75) used to annotate data

```
$ cd /home/ubuntu/
$ java -jar snpEff.jar download -v GRCh37.75
```

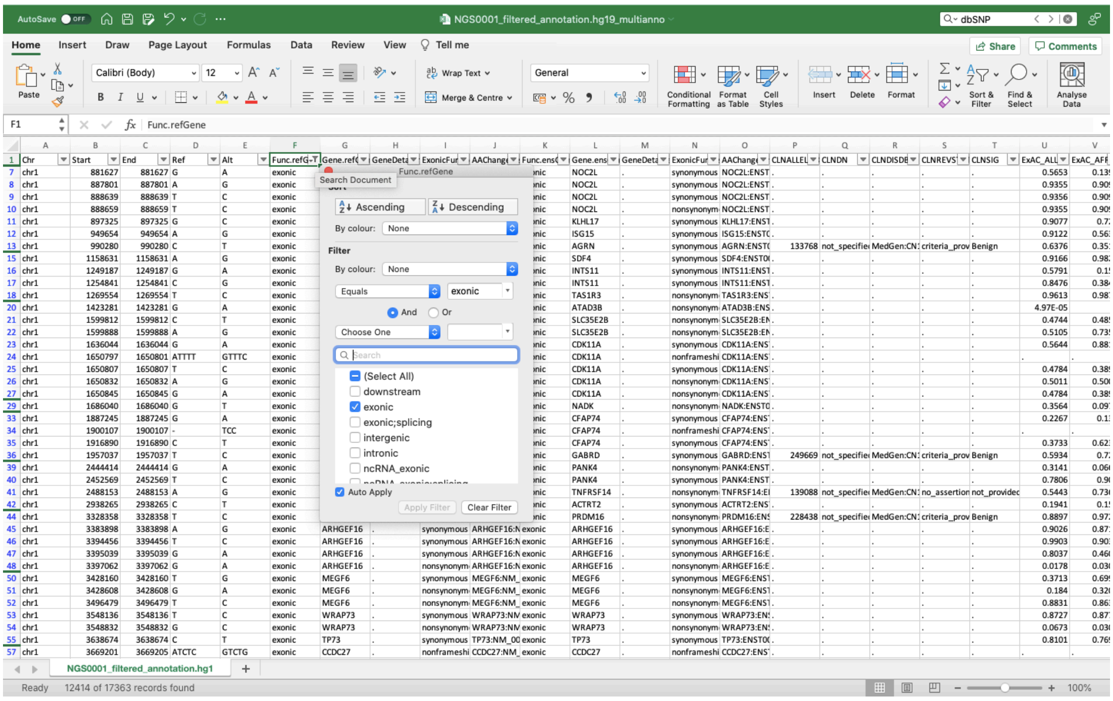
# I couldn't figure out how to download the latest version of this database and kept getting errors but the next steps to be taken after this would be to annotate the VCF file produced previously by running the following command.

```
$ java -Xmx8g -jar snpEff.jar GRCh37.75
/home/ubuntu/ngs_course/dna_seq_assignment/results/NGS0001_filtered_annotation.
vcf >
/home/ubuntu/ngs_course/dna_seq_assignment/results//home/ubuntu/ngs_course/dna
_seq_assignment/results/NGS0001_filtered_annotation_snpEFF.vcf
```

Perform basic variant prioritization: filter to exonic variants not seen in dbSNP (2pts)

# ANNOVAR produced VCF file can be manually filtered in excel to show exonic variants not seen in dbSNP. To do this turn on filtering in excel and choose what you want to include/exclude.

# For example to look at only exonic variants.



F1 Func.refGene

Func.refGene

Sort

Ascending | Descending

By colour: None

Filter

By colour: None

Equals exonic

● And ○ Or

Choose One

Search

- ▬ (Select All)
- □ downstream
- ☑ exonic
- □ exonic;splicing
- □ intergenic
- □ intronic
- □ ncRNA_exonic

☑ Auto Apply

Apply Filter | Clear Filter

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	Chr	Start	End	Ref	Alt	Func.refG	Gene.ref(	GeneDeta	ExonicFur	AAChange	Func.ens(	Gene.ens	GeneDeta	ExonicFur	AAChange	CLNALLEL	CLNDN	CLNDISDE	CLNREVS	CLNSIG	ExAC_ALL	ExAC_AFF
7	chr1	881627	881627	G	A	exonic					nic	NOC2L	.	synonymous	NOC2L:ENST	.	.	.	.	.	0.5653	0.13
8	chr1	887801	887801	A	G	exonic					nic	NOC2L	.	synonymous	NOC2L:ENST	.	.	.	.	.	0.9355	0.90
9	chr1	888639	888639	T	C	exonic					nic	NOC2L	.	synonymous	NOC2L:ENST	.	.	.	.	.	0.9356	0.90
10	chr1	888659	888659	T	C	exonic					nic	NOC2L	.	nonsynonym	NOC2L:ENST	.	.	.	.	.	0.9355	0.90
11	chr1	897325	897325	G	C	exonic					nic	KLHL17	.	synonymous	KLHL17:ENST	.	.	.	.	.	0.9077	0.7
12	chr1	949654	949654	A	G	exonic					nic	ISG15	.	synonymous	ISG15:ENST0	.	.	.	.	.	0.9122	0.56
13	chr1	990280	990280	C	T	exonic					nic	AGRN	.	synonymous	AGRN:ENST0	133768	not_specifie	MedGen:CN1	criteria_prov	Benign	0.6376	0.35
15	chr1	1158631	1158631	A	G	exonic					nic	SDF4	.	synonymous	SDF4:ENST0	.	.	.	.	.	0.9166	0.98
16	chr1	1249187	1249187	G	A	exonic					nic	INTS11	.	synonymous	INTS11:ENST	.	.	.	.	.	0.5791	0.1
17	chr1	1254841	1254841	C	G	exonic					nic	INTS11	.	synonymous	INTS11:ENST	.	.	.	.	.	0.8476	0.38
18	chr1	1269554	1269554	T	C	exonic					nic	TAS1R3	.	nonsynonym	TAS1R3:ENS	.	.	.	.	.	0.9613	0.98
20	chr1	1423281	1423281	G	A	exonic					nic	ATAD3B	.	nonsynonym	ATAD3B:ENS	.	.	.	.	.	4.97E-05	
21	chr1	1599812	1599812	C	T	exonic					nic	SLC35E2B	.	nonsynonym	SLC35E2B:EN	.	.	.	.	.	0.4744	0.48
22	chr1	1599888	1599888	A	G	exonic					nic	SLC35E2B	.	nonsynonym	SLC35E2B:EN	.	.	.	.	.	0.5105	0.73
23	chr1	1636044	1636044	G	A	exonic					nic	CDK11A	.	synonymous	CDK11A:ENS	.	.	.	.	.	0.5644	0.88
24	chr1	1650797	1650801	ATTTT	GTTTC	exonic					nic	CDK11A	.	nonframeshi	CDK11A:ENS	.	.	.	.	.	.	.
25	chr1	1650807	1650807	T	C	exonic					nic	CDK11A	.	synonymous	CDK11A:ENS	.	.	.	.	.	0.4784	0.38
26	chr1	1650832	1650832	A	G	exonic					nic	CDK11A	.	nonsynonym	CDK11A:ENS	.	.	.	.	.	0.5011	0.50
27	chr1	1650845	1650845	G	A	exonic					nic	CDK11A	.	nonsynonym	CDK11A:ENS	.	.	.	.	.	0.4784	0.38
29	chr1	1686040	1686040	G	T	exonic					nic	NADK	.	nonsynonym	NADK:ENST0	.	.	.	.	.	0.3564	0.09
33	chr1	1887245	1887245	G	A	exonic					nic	CFAP74	.	synonymous	CFAP74:ENS	.	.	.	.	.	0.2267	0.1
34	chr1	1900107	1900107	-	TCC	exonic					nic	CFAP74	.	nonframeshi	CFAP74:ENS	.	.	.	.	.	.	.
35	chr1	1916890	1916890	C	T	exonic					nic	CFAP74	.	synonymous	CFAP74:ENS	.	.	.	.	.	0.3733	0.62
36	chr1	1957037	1957037	T	C	exonic					nic	GABRD	.	synonymous	GABRD:ENST	249669	not_specifie	MedGen:CN1	criteria_prov	Benign	0.5934	0.7
39	chr1	2444414	2444414	G	A	exonic					nic	PANK4	.	nonsynonym	PANK4:ENST	.	.	.	.	.	0.3141	0.06
40	chr1	2452569	2452569	T	C	exonic					nic	PANK4	.	synonymous	PANK4:ENST	.	.	.	.	.	0.7806	0.9
41	chr1	2488153	2488153	A	G	exonic					nic	TNFRSF14	.	nonsynonym	TNFRSF14:E	139088	not_specifie	MedGen:CN1	no_assertion	not_providec	0.5443	0.73
42	chr1	2938265	2938265	C	T	exonic					nic	ACTRT2	.	synonymous	ACTRT2:ENS	.	.	.	.	.	0.1941	0.1
44	chr1	3328358	3328358	T	C	exonic					nic	PRDM16	.	nonsynonym	PRDM16:ENS	228438	not_specifie	MedGen:CN1	criteria_prov	Benign	0.8897	0.97
45	chr1	3383898	3383898	A	G	exonic	ARHGEF16	.	synonymous	ARHGEF16:N	exonic	ARHGEF16	.	synonymous	ARHGEF16:E	.	.	.	.	.	0.9026	0.87
46	chr1	3394456	3394456	T	C	exonic	ARHGEF16	.	synonymous	ARHGEF16:N	exonic	ARHGEF16	.	synonymous	ARHGEF16:E	.	.	.	.	.	0.9903	0.90
47	chr1	3395039	3395039	G	A	exonic	ARHGEF16	.	synonymous	ARHGEF16:N	exonic	ARHGEF16	.	synonymous	ARHGEF16:E	.	.	.	.	.	0.8037	0.46
48	chr1	3397062	3397062	G	A	exonic	ARHGEF16	.	nonsynonym	ARHGEF16:N	exonic	ARHGEF16	.	nonsynonym	ARHGEF16:E	.	.	.	.	.	0.0178	0.03
50	chr1	3428160	3428160	T	G	exonic	MEGF6	.	synonymous	MEGF6:NM_	exonic	MEGF6	.	synonymous	MEGF6:ENST	.	.	.	.	.	0.3713	0.69
51	chr1	3428608	3428608	G	A	exonic	MEGF6	.	nonsynonym	MEGF6:NM_	exonic	MEGF6	.	nonsynonym	MEGF6:ENST	.	.	.	.	.	0.184	0.32
52	chr1	3496479	3496479	T	C	exonic	MEGF6	.	nonsynonym	MEGF6:NM_	exonic	MEGF6	.	nonsynonym	MEGF6:ENST	.	.	.	.	.	0.8831	0.86
53	chr1	3548136	3548136	T	C	exonic	WRAP73	.	synonymous	WRAP73:NM	exonic	WRAP73	.	synonymous	WRAP73:ENS	.	.	.	.	.	0.8727	0.87
54	chr1	3548832	3548832	G	C	exonic	WRAP73	.	nonsynonym	WRAP73:NM	exonic	WRAP73	.	nonsynonym	WRAP73:ENS	.	.	.	.	.	0.0673	0.03
55	chr1	3638674	3638674	C	T	exonic	TP73	.	synonymous	TP73:NM_00	exonic	TP73	.	synonymous	TP73:ENST0	.	.	.	.	.	0.8101	0.76
57	chr1	3669201	3669205	ATCTC	GTCTG	exonic	CCDC27	.	nonframeshi	CCDC27:NM_	exonic	CCDC27	.	nonframeshi	CCDC27:ENS	.	.	.	.	.	.	.

Ready 12414 of 17363 records found

3. R/RStudio assessment (40pts – 40% of final mark)

In this assessment you will be asked to perform a number of tasks in R/RStudio and report them in your own markdown document.

Initial task: Create a new markdown document in *RStudio*, set the title to "Advanced Bioinformatics 2019 assessment", and insert an "author:" tag below the title, followed by your student id. Share your markdown document and html via your github account.

In the following, for each task, create a new heading called "Task X" for task X, and insert a new R code chunk that holds any code required. Make sure to evaluate the expression before saving to include the output in the html file. If you have multiple lines that produce outputs, you can split them into separate code chunks for increase clarity (but it is not necessary to pass the assessment). Please also explain your steps.

3.1. Using the *sum*() function and : operator, write an expression in the code snippet to evaluate the sum of all integers between 5 and 55. (5pt)

3.2. Write a function called *sumfun* with one input parameter, called *n*, that calculates the sum of all integers between 5 and *n*. Use the function to do the calculation for *n* = 10, *n* = 20, and *n* = 100 and present the results. (5pt)

3.3. The famous Fibonacci series is calculated as the sum of the two preceding members of the sequence, where the first two steps in the sequence are 1, 1. Write an R script using a for loop to calculate and print out the first 12 entries of the Fibonacci series. (5pt)

3.4. With the *mtcars* dataset bundled with R, use *ggplot* to generate a box of miles per gallon (in the variable *mpg*) as a function of the number of gears (in the variable *gear*). Use the fill aesthetic to colour bars by number of gears. (5pt)

3.5. Using the *cars* dataset and the function *lm*, fit a linear relationship between *speed* and breaking distance in the variable *distance*. What are the fitted slope and intercept of the line, and their standard errors? What are the units used for the variables in the dataset? (5pt)

3.6. Use *ggplot* to plot the data points from Task 6 and the linear fit. (5pt)

3.7. Again using the cars dataset, now use linear regression (*lm*) to estimate the average reaction time for the driver to start breaking (in seconds). To simplify matters you may assume that once breaking commences, breaking distance is proportional to the square of the speed. Explain the steps in your analysis. Do you get reasonable results? Finally, use *ggplot* to plot the data points and the fitted relationship. (10pt)