Write a script that automate the analysis for you

To write a simple bash script you just need to open a text editor (Sublime, Notepad, Notepad++, Gedit, Vim, Textwrangler), write the commands that you want to execute and save the document as “the_name_of_your_script.sh” (ended in .sh). You can modify any of the scripts supplied as examples.

The first line of your script need to tell the interpreter coding language used (#!/bin/bash):

```
#!/bin/bash
```

Next initial lines will describe author, date, purpose of the script and usage instructions. To comment lines that will not be interpreted you can use # symbol.

```
#!/bin/bash

# @ author
# Date ---

# Script align BQ using students options.
# Inputs:
#             - Output name                             output_name
#             - Raw sequencing reads:                   reads.fq.gz
#             - Path for Bowtie2 index sequence
/home/manager/Desktop/course_materials/genomes/AFPN02.1/AFPN02.1_merge
#             - Bowtie2 parameters from the manual      "--end-to-end -D
int -R int..."      # It's better to quote this parameters
#               (http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml)
#   Usage:
#
#   bash align_BQ.sh reads.fq.gz
/home/manager/Desktop/course_materials/genomes/AFPN02.1/AFPN02.1_merge "--
end-to-end  -D int -R int..."
#
```

Then we capture each of the inputs with the special variables $1 (first input), $2 (second input), $3 (third input) and assign then to variables with meaningful names. Variables names should be a single word or split by underscores.

```
# Inputs
output_name=$1
reads=$2
path_index=$3
```

```
bowtie2_parameters=$4

# To use this varaibles you should use this format
${name_of_your_variable}
${output_name}
${reads}
${path_index}
${bowtie2_parameters}
```

To print a statement, info or variable you need to use command echo.

```
echo "Hello World"

# You can also print a instruction to see how you are supplying parameters:
echo "Output Name              "${output_name}
echo "Name of reads            "${reads}
echo "Path to Bowtie2 index    "${path_index}
echo "Bowtie2 parameters:      "${bowtie2_parameters}

# Print separator Input Output
printf "#####################\n\n"
```

Then save your program as (align_BQ.sh) and change the priorities to be able to run it. On the command line:

```
chmod -R 777 align_BQ.sh
```

To call your program on the command line just run:

```
bash align_BQ.sh reads.fq.gz
/home/manager/Desktop/course_materials/genomes/AFPN02.1/AFPN02.1_merge "--
end-to-end  -D int -R int..."
```

Example Script

I will supply more example scripts on ADI if students enroll on this topic. This will a final script that do some automation to our problem.

```
#!/bin/bash

# @ author
# Jan 2021

# Script align BQ using students options.
# Inputs:
#             - Output name                             output_name
#             - Raw sequencing reads:                   reads.fq.gz
#             - Path for Bowtie2 index sequence
/home/manager/Desktop/course_materials/genomes/AFPN02.1/AFPN02.1_merge
#             - Bowtie2 parameters from the manual      "--end-to-end -D
int -R int..."      # It's better to quote this parameters
#               (http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml)
#   Usage:
#
#   bash align_BQ.sh reads.fq.gz
/home/manager/Desktop/course_materials/genomes/AFPN02.1/AFPN02.1_merge "--
end-to-end  -D int -R int..."
#



# Inputs
output_name=$1
reads=$2
path_index=$3
bowtie2_parameters=$4



# Print inputs
echo "Output Name              "${output_name}
echo "Name of reads            "${reads}
echo "Path to Bowtie2 index    "${path_index}
echo "Bowtie2 parameters:      "${bowtie2_parameters}



# Cutadapt options
# cutadapt -q --trim-n .....

# Aligning BQ with Bowtie2
echo "Aligning BQ with Bowtie2"
time bowtie2 ${bowtie2_parameters} -x ${path_index} -q ${reads} -S
${output_name}.sam 2> ${output_name}_bowtie_stats.txt

# You can also print on screen the translated instruction you are trying to
run
echo "time bowtie2 ${bowtie2_parameters} -x ${path_index} -q ${reads} -S
${output_name}.sam 2> ${output_name}_bowtie_stats.txt"



# Samtools
# samtools ......
# samtools sort ..... > ....
# samtools index ......
# samtools stats ..... > .....
# samtools flagstat .......
# run multiQC
multiqc . -f

# Remove intermediate files and final clean up. Always remove intermediate files after you capture the stats with multiQC or you will end up with lots of files on your computer.
rm -r ${output_name}.sam ${output_name}_bowtie_stats.txt
# rm ........
```

Some tutorials and cheatsheet

```
https://devhints.io/bash

https://www.codecademy.com/learn/learn-the-command-line/modules/bash-scripting

https://linuxconfig.org/bash-scripting-tutorial-for-beginners

https://en.wikibooks.org/wiki/Bash_Shell_Scripting
```