MEDICAL MICROBIOLOGY AND INFECTIOUS DISEASES CODING WORKSHOP

Presents

Introduction to BASH

INSTRUCTED BY

Grace E. Seo (MSc student)

Email: seog@myumanitoba.ca

INFORMATION FOR PARTICIPANTS

All workshops are being recorded and posted to the
MMID Coding Workshop - YouTube

Please hold your questions until Q & A session
*Question and Answer period will not be recorded.*

LEARNING OBJECTIVES

1. *Describe benefits of using command line interface (CLI) over graphical user interface (GUI)*

2. *Install BASH terminal on your operating system*

3. *Navigate, create, rename, move, and copy directory and files in terminal*

4. *Install, execute, and troubleshoot bioinformatics program*

5. *Write a BASH script, change permission, and execute the script*

LEARNING OBJECTIVES

1. **Describe benefits of using command line interface (CLI) over graphical user interface (GUI)**

2. *Install BASH terminal on your operating system*

3. *Navigate, create, rename, move, and copy directory and files in terminal*

4. *Install, execute, and troubleshoot bioinformatics program*

5. *Write a BASH script, change permission, and execute the script*
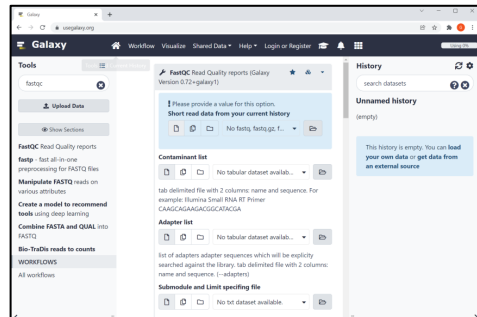
Graphical user interface (GUI) + Magic? = Result

https://usegalaxy.org/
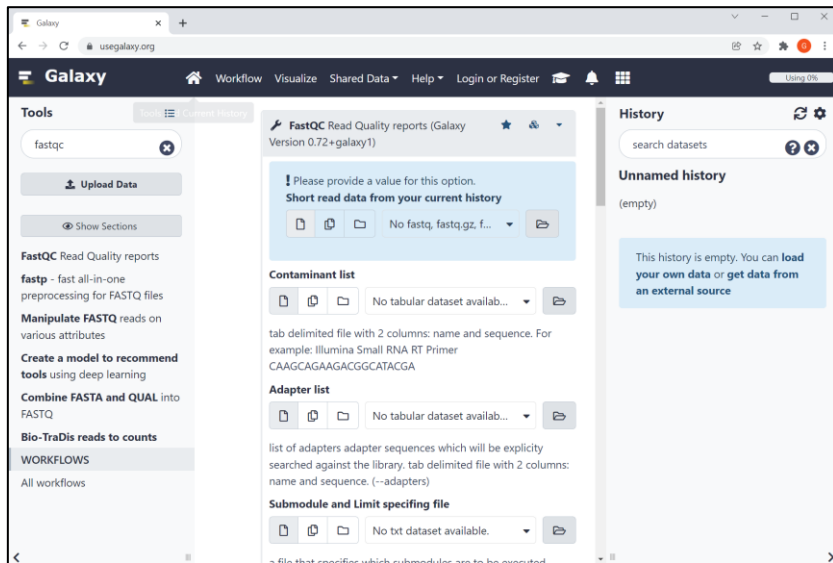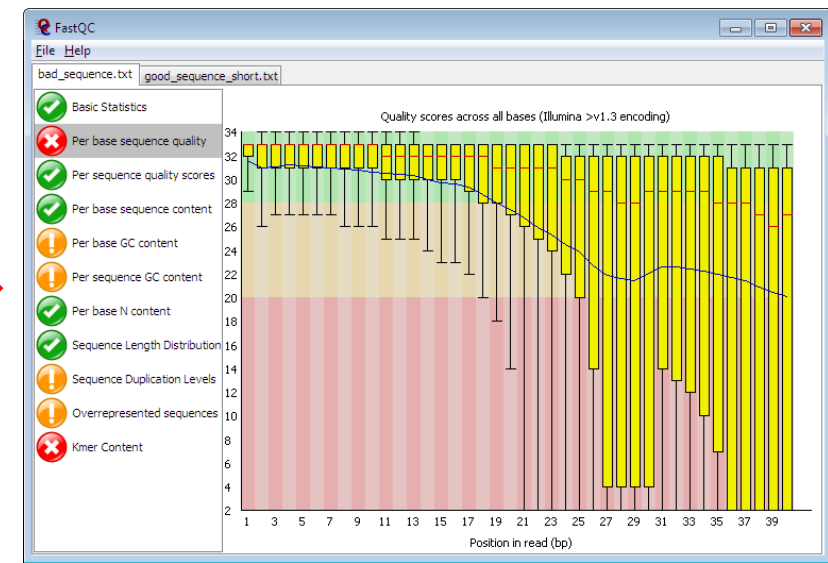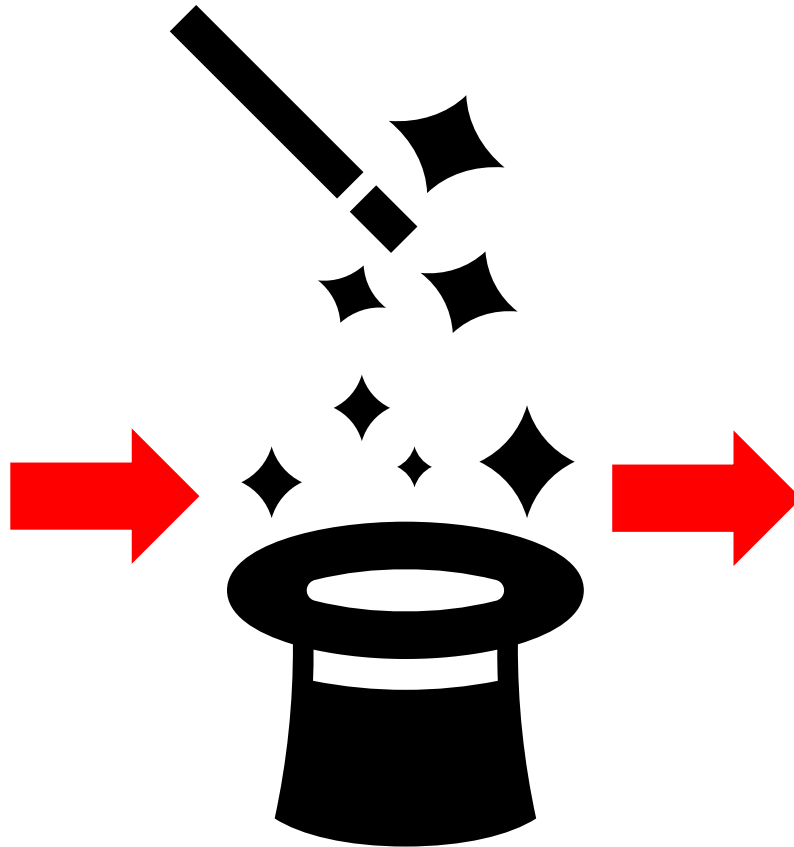
https://du-bii.github.io/module-5-Methodes-Outils/seance1_NGS/slides.html#1

Graphical user interface (GUI) →
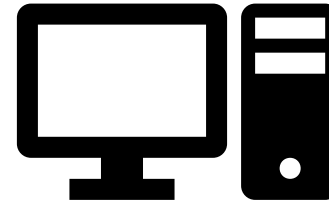# Command line interface (CLI) → Computer = Result

GUI

https://usegalaxy.org/

CLI

Computer

OR

https://en.wikipedia.org/wiki/Supercomputer

Result

https://du-bii.github.io/module-5-Methodes-Outils/seance1_NGS/slides.html#1

BASH (Bourne Again SHell) language

- **Command line interface aka console or terminal in Unix operating system**

- **BASH originally written by Stephen Bourne**

- **Unix shell is:**
  - **programming language**
  - **command interpreter**

- **BASH is a default shell in Linux and most Unix-like operating systems**

Why use command line interface (CLI)?

1. **Program is always available on CLI but might not be available on GUI**

2. Take control over your analysis (set your own parameters)

3. Analyze faster using less computer resource

4. Let computers do repetitive tasks for you

5. Automate your jobs and get them done while you are sleeping!

Why use command line interface (CLI)?

1. Program is always available on CLI but might not be available on GUI

2. **Take control over your analysis (set your own parameters)**

3. Analyze faster using less computer resource

4. Let computers do repetitive tasks for you

5. Automate your jobs and get them done while you are sleeping!

Why use command line interface (CLI)?

3. Analyze faster using less computer resource

VS

https://decisionstats.com/2015/12/25/interview-damien-farrell-python-gui-dataexplore-python-rstats-pydata/

https://data36.com/command-line-data-science-introduction-to-bash/
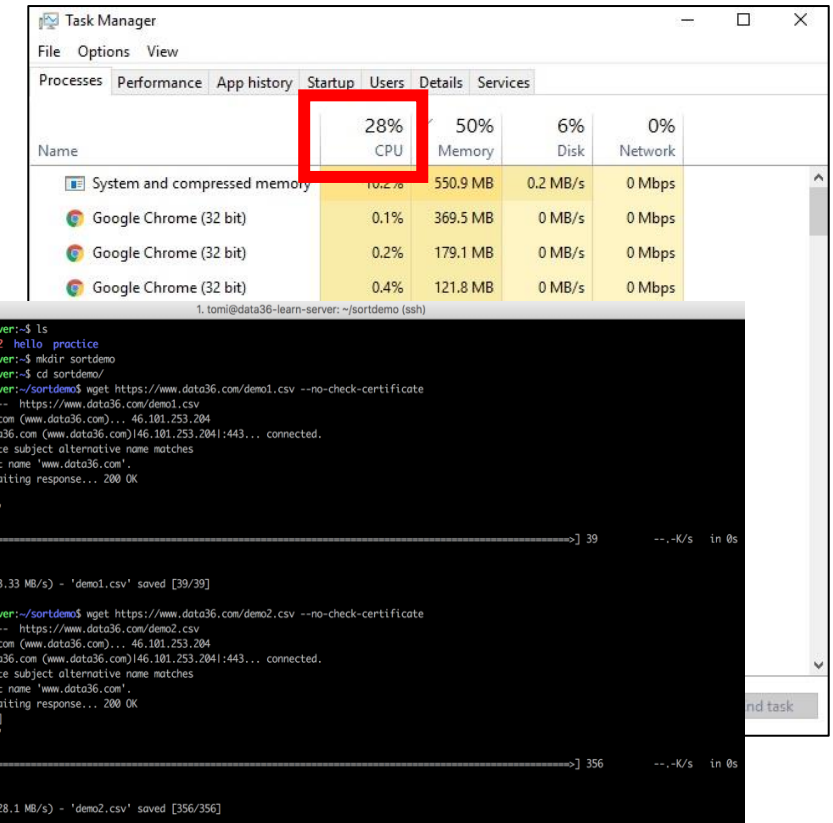
Why use command line interface (CLI)?

4. Let computers do repetitive tasks for you

VS

Why use command line interface (CLI)?
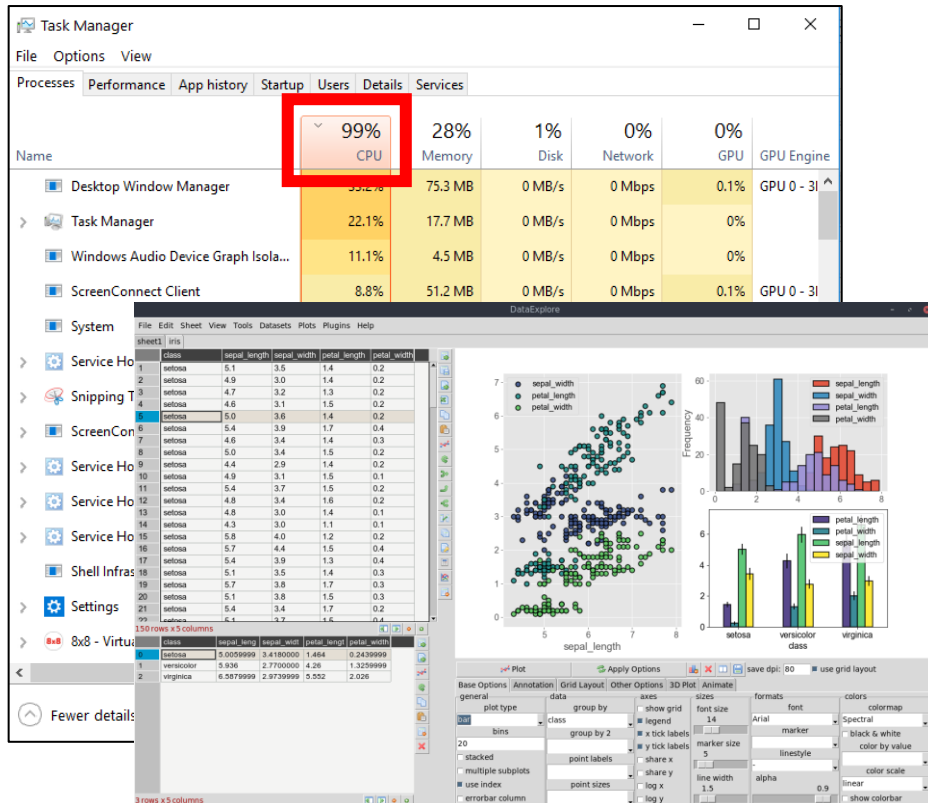
1. Program is always available on CLI but might not be available on GUI

2. Take control over your analysis (set your own parameters)

3. Analyze faster using less computer resource

4. Let computers do repetitive tasks for you

5. Automate your jobs and get them done while you are sleeping!

LEARNING OBJECTIVES

1. *Describe benefits of using command line interface (CLI) over graphical user interface (GUI)*

2. **Install BASH terminal on your operating system**

3. *Navigate, create, rename, move, and copy directory and files in terminal*

4. *Install, execute, and troubleshoot bioinformatics program*

5. *Write a BASH script, change permission, and execute the script*

Virtual Machine (VM)

Virtual machine is a computer within your actual computer

Advantages:

- **Explore and try BASH coding without affecting actual computer system**
- **If something happens, simply delete the virtual machine**

Disadvantages:

- **If you are doing data analysis, you need to transfer files between computer and VM using a "shared folder"**
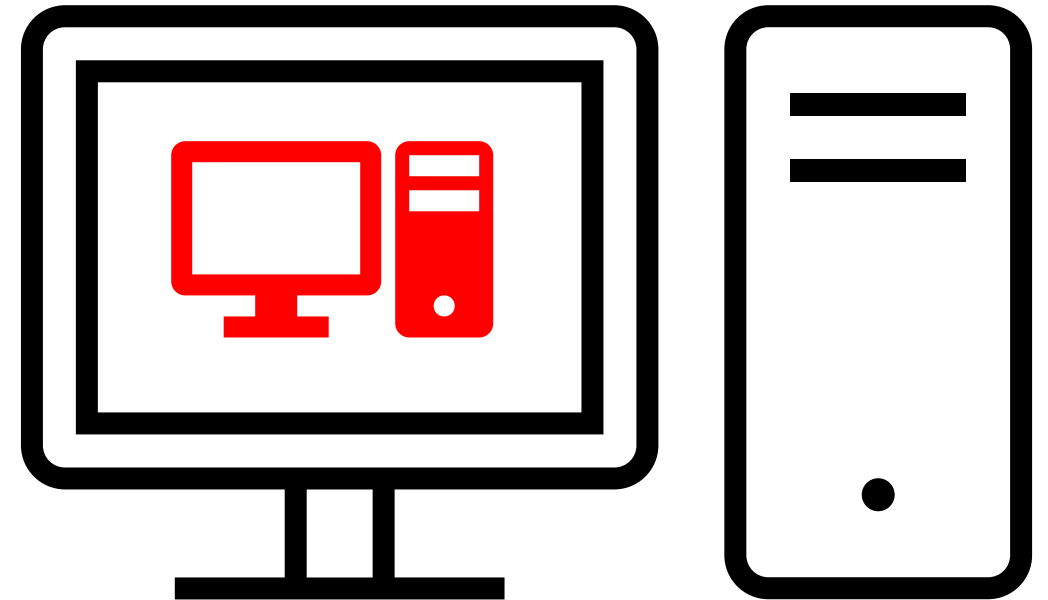
Virtual Machine (VM)

Follow this guide to install and set-up Ubuntu VM: https://itsfoss.com/install-linux-in-virtualbox/
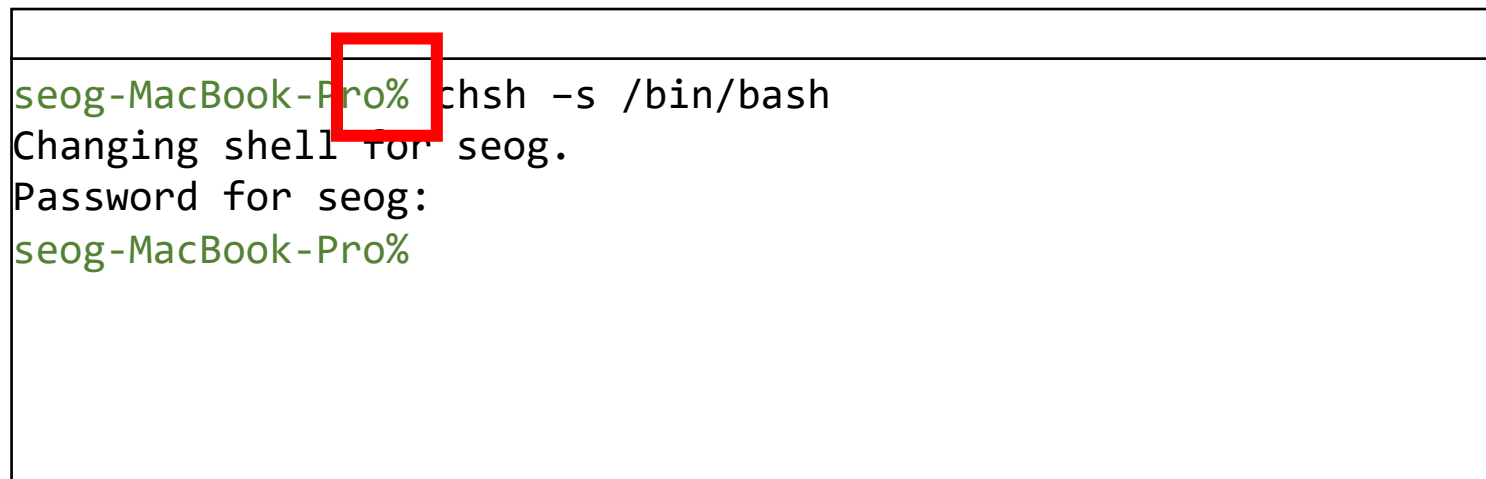
1. **Download VM by clicking the correct host**
   https://www.virtualbox.org/wiki/Downloads
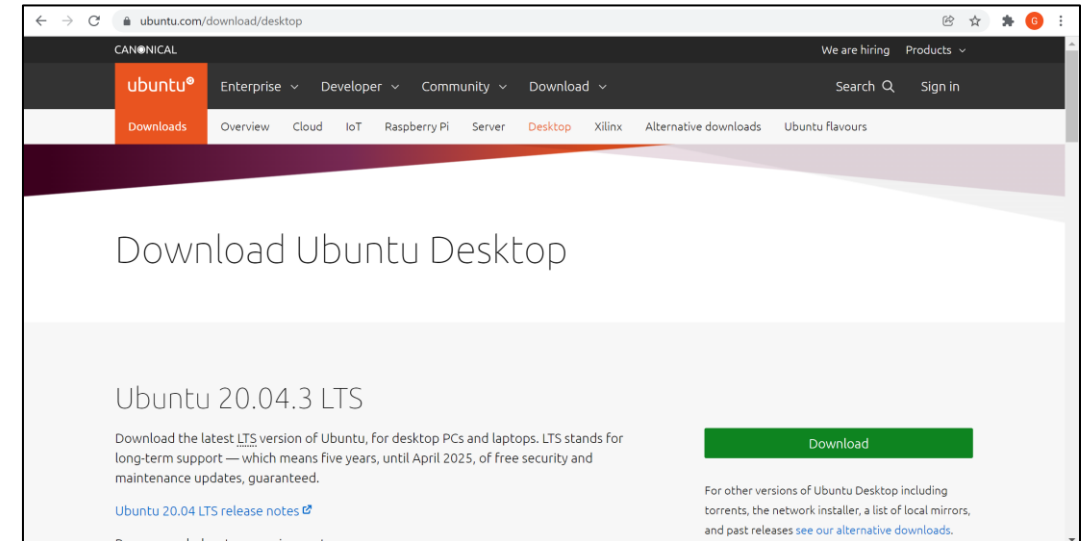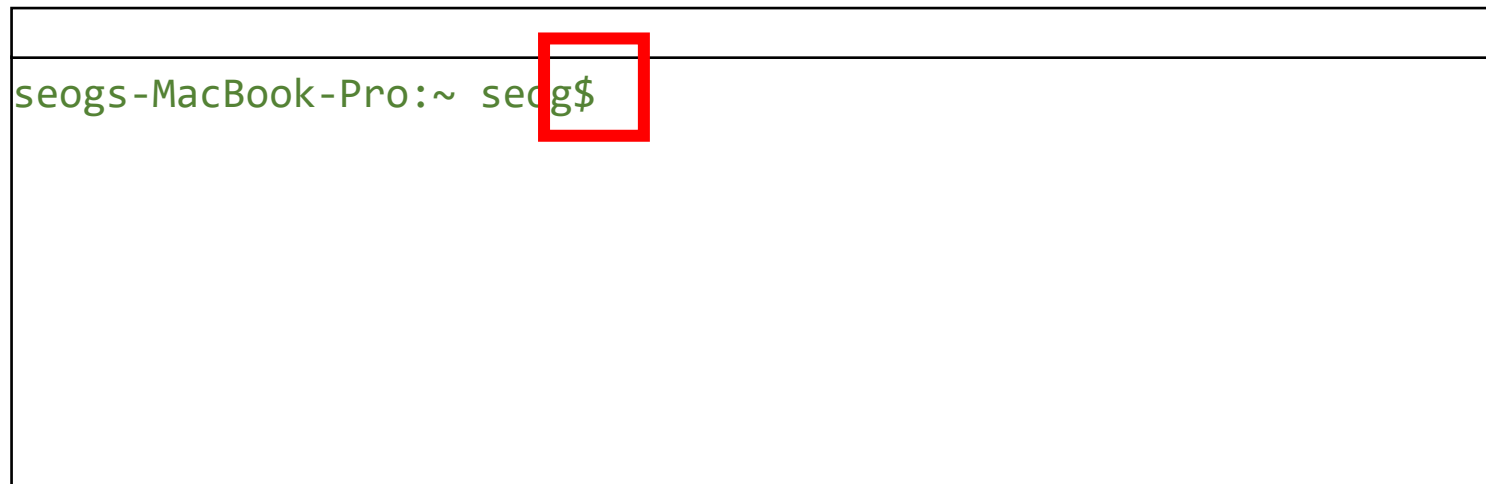
2. **Download Ubuntu iso**
   https://ubuntu.com/desktop

Linux and Mac OS – default terminal App

- **Open terminal app (Linux and Mac OS)**

- **BASH was the default shell in Mac OS until 2019 (replaced with Zsh)**

- **In Mac OS - To change BASH back to default shell:**
  - **Type the command: chsh –s /bin/bash then Restart the terminal**

```
seog-MacBook-Pro% chsh –s /bin/bash
Changing shell for seog.
Password for seog:
seog-MacBook-Pro%
```

Linux and Mac OS – default terminal App

- **Open terminal app (Linux and Mac OS)**

- **BASH was the default shell in Mac OS until 2019 (replaced with Zsh)**

- **In Mac OS - To change BASH back to default shell:**
  - Type the command: **chsh –s /bin/bash** then **Restart the terminal**

BASH prompt is $

```
seogs-MacBook-Pro:~ seog$
```

Windows Subsystem for Linux (WSL)

To use BASH on Windows, enable WSL

Advantages:

- Use BASH terminal on Windows as if you are on Linux machine.

- BASH terminal can be accessed using: PowerShell, command prompt or any other terminal programs.

- Access files and run bioinformatics program directly on your computer.

Disadvantages:

- You **can't undo** what you just did and there is **no trash can.**

Windows Subsystem for Linux (WSL)
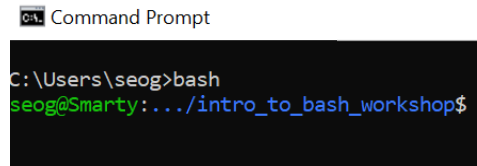
1. **Open control panel from Start menu.**

2. **Click "Programs" → "Turn Windows features on or off".**

3. **Scroll down and select "Windows Subsystem for Linux" then restart your computer.**

4. **In "Microsoft Store", search and install "Ubuntu" from Canonical Group Limited.**

5. **Open Ubuntu and setup your username and password.**
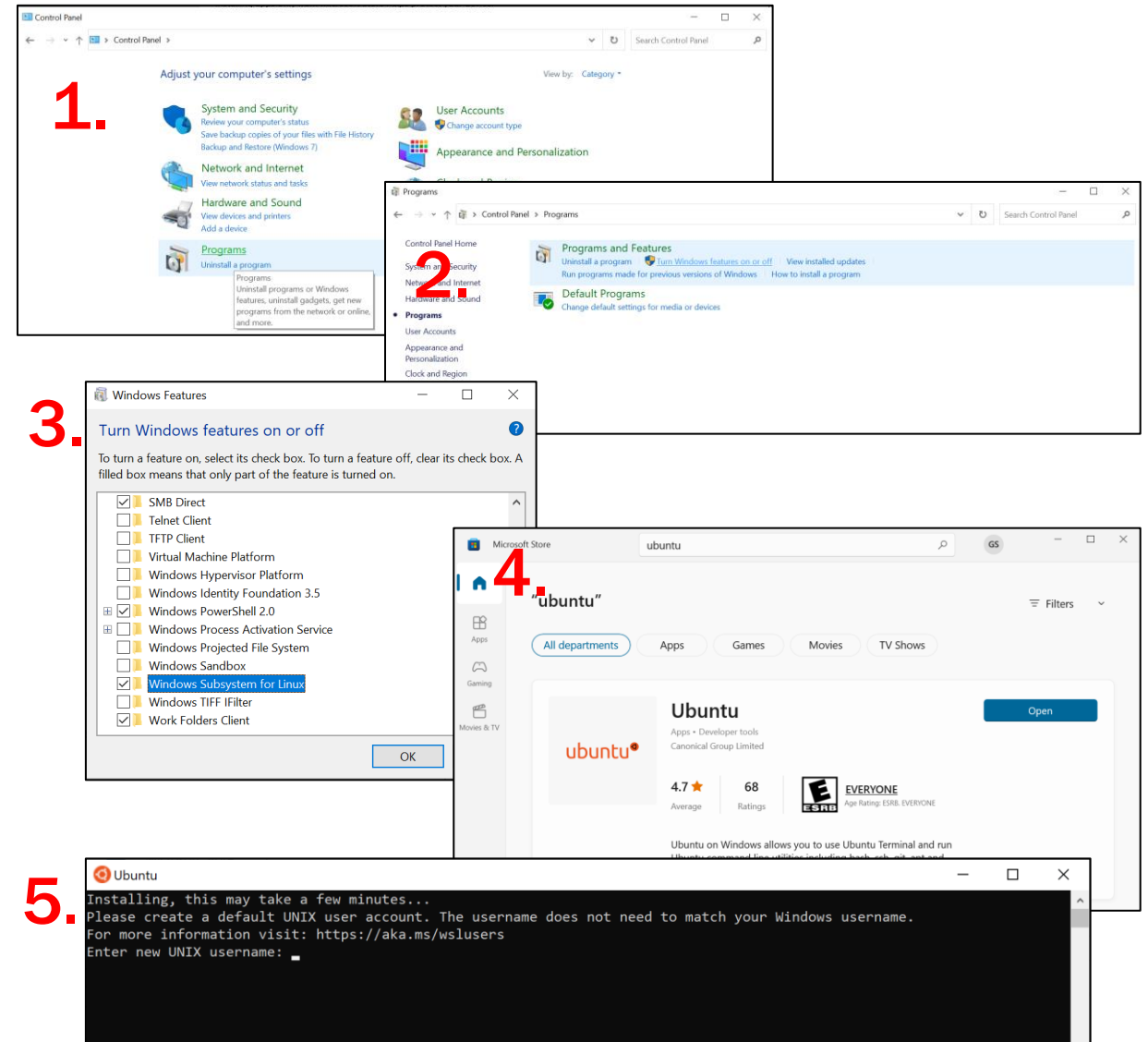
Examples of BASH terminal emulators

- ## Windows default apps
  - ### Command prompt
  - ### PowerShell
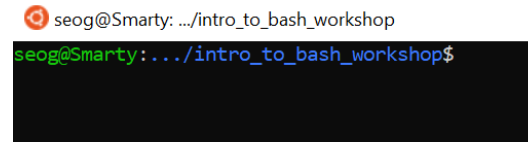
- ## Terminal emulator
  - ### Ubuntu terminal (emulator)
  - ### Visual Studio Code (emulator & text editor)
    - https://code.visualstudio.com/download
  - ### MobaXTerm (emulator & SSH client)
    - https://mobaxterm.mobatek.net/download-home-edition.html
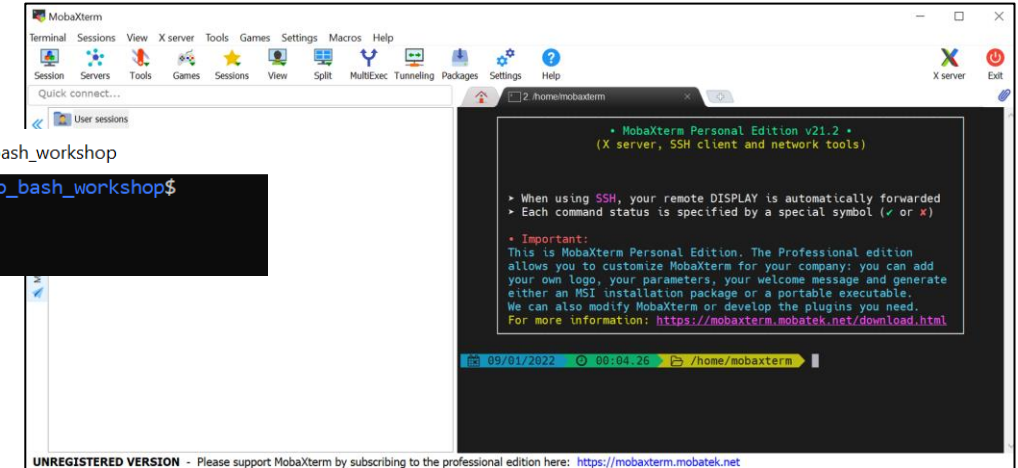
- ## SSH (secure shell) client only
  - ### Bitvise https://www.bitvise.com/
  - ### PuTTy
    https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html
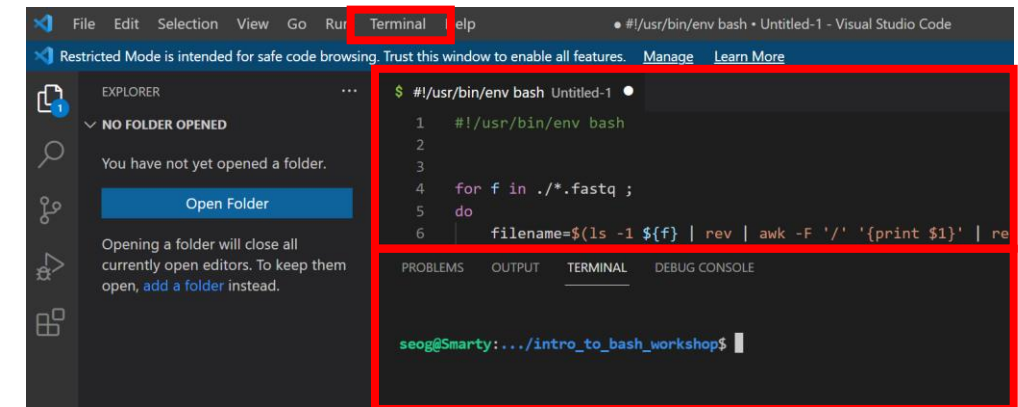
LEARNING OBJECTIVES

1. *Describe benefits of using command line interface (CLI) over graphical user interface (GUI)*

2. *Install BASH terminal on your operating system*

3. **Navigate, create, rename, move, and copy directory and files in terminal**

4. *Install, execute, and troubleshoot bioinformatics program*

5. *Write a BASH script, change permission, and execute the script*

BASH prompt

Users specify items written in ALL CAPS

Current directory

USERNAME@COMPUTER_NAME:/Desktop$

```
seog@Smarty:.../Desktop$
```

Prompt – Input command

If there is no prompt, either job is in progress or wrong command stalled your terminal

To get your prompt back
(Windows / Linux) Press ctrl-c
(Mac) Press command-c

Example Windows Subsystem for Linux BASH folder structure

Folder	Absolute path	Relative path
/ Root (BASH)	/	../../../../../
/bin /boot /dev /etc /home /mnt /usr	/mnt	../../../../
/c	/mnt/c	../../../
/Users	/mnt/c/Users	../../
/seog	/mnt/c/Users/seog	../
/Desktop /Documents /Downloads /Favorites		./

Example original folder path
(Windows) C:\Desktop
(Mac)   /Users/seog/Desktop
(Linux)   /home/seog/Desktop

Current directory: **/mnt/c/Users/seog/Desktop**

Example Windows Subsystem for Linux BASH folder structure

	Absolute path	Relative path
/ Root (BASH)	/	../../../../../
/bin /boot /dev /etc /home /mnt /usr	/mnt	../../../../
/c	/mnt/c	../../../
/Users	/mnt/c/Users	../../
/seog	/mnt/c/Users/seog	../
/Desktop /Documents /Downloads /Favorites		../Downloads

Example original folder path
(Windows) C:\Desktop
(Mac)   /Users/seog/Desktop
(Linux)   /home/seog/Desktop

Current directory: /mnt/c/Users/seog/Downloads
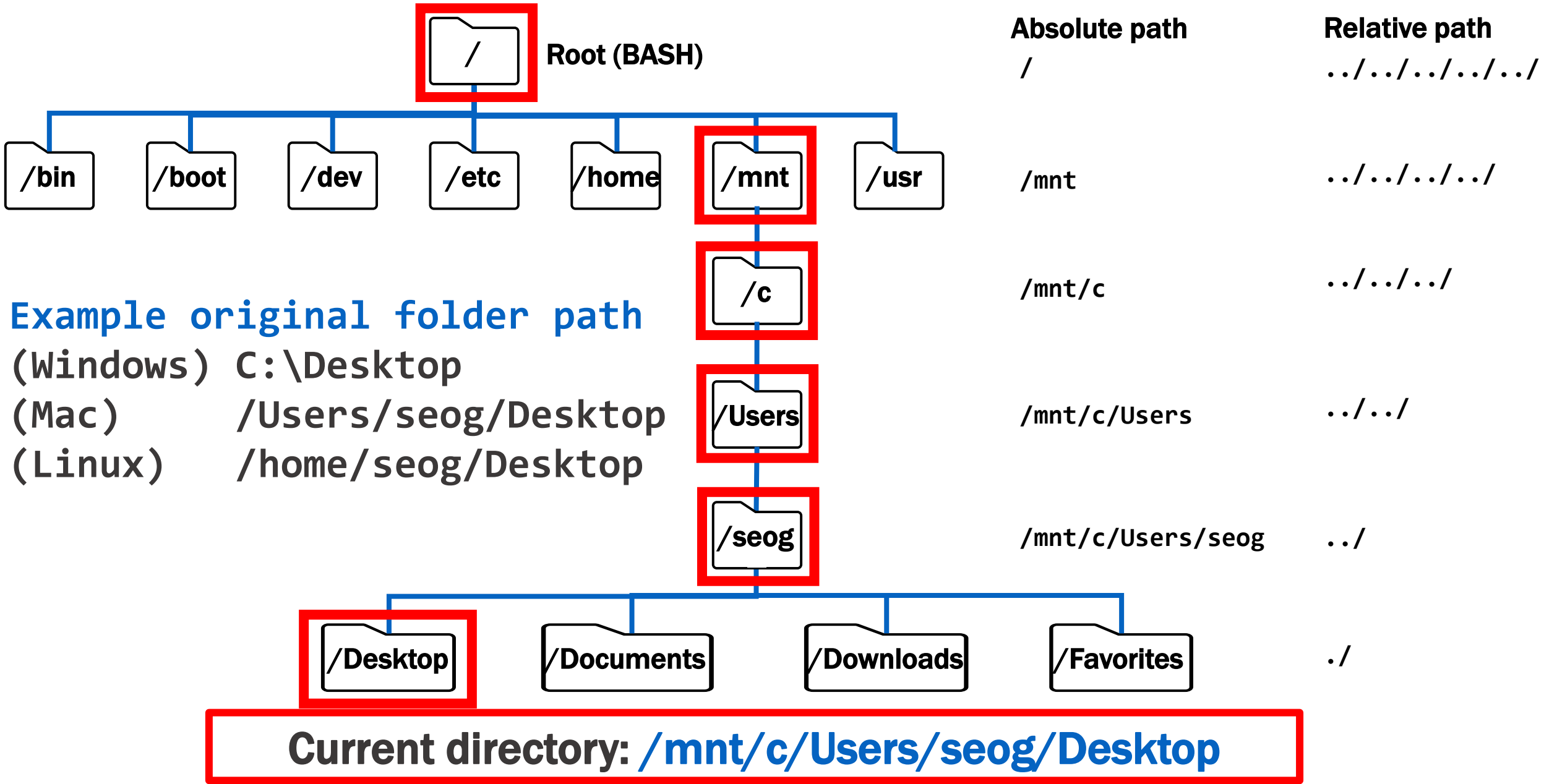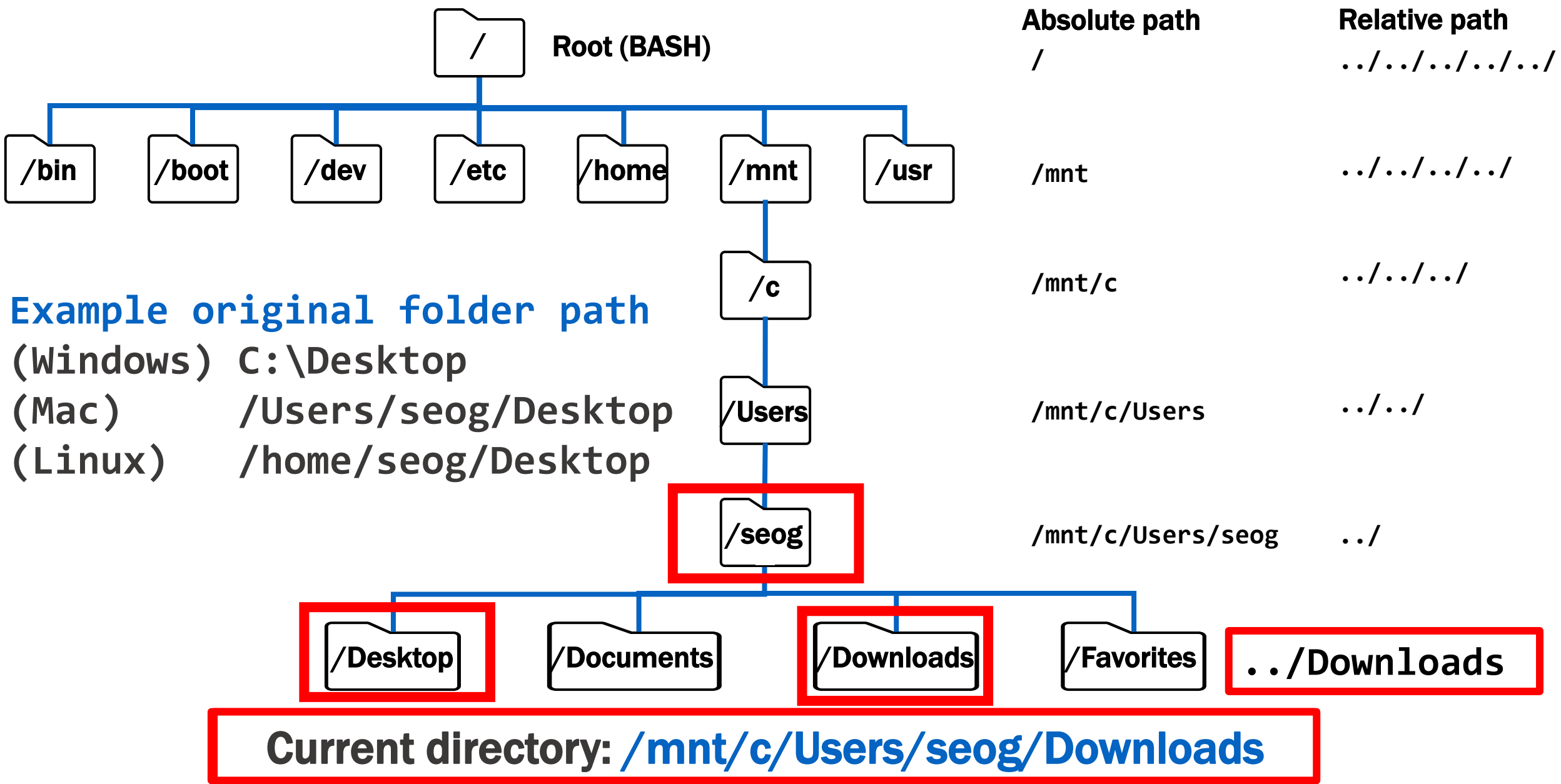
Navigate directories (folders)

Users specify items written in ALL CAPS

$ pwd   *(print current working directory)*

$ cd /PATH   *(change directory)*

```
seog@Smarty:~$ pwd
/home/seog
seog@Smarty:~$ cd /mnt/c/Users/seog/Desktop
seog@Smarty:.../Desktop$ pwd
/mnt/c/Users/seog/Desktop
seog@Smarty:.../Desktop$ cd /home
seog@Smarty:/home$ cd ../mnt/c/Users/seog/Desktop
seog@Smarty:.../Desktop$ cd ../../
seog@Smarty:.../Users$ cd seog/Desktop
seog@Smarty:.../Desktop$
```

Change highlighted part with your username
(WSL)   /mnt/c/Users/`seog`/Desktop
(Mac)   /Users/`seog`/Desktop
(Linux) /home/`seog`/Desktop

Create directory and files

Users specify items written in ALL CAPS

$ mkdir NAME   *(create a new folder)*

$ touch NAME   *(create a new file)*

$ ls   *(list files in the folder)*

$ ls -l   *(list files in the folder with long details)*

```
seog@Smarty:.../Desktop$ mkdir intro_to_bash
seog@Smarty:.../Desktop$ cd intro_to_bash
seog@Smarty:.../intro_to_bash$ touch test0001.txt test0002.txt
seog@Smarty:.../intro_to_bash$ ls
test0001.txt  test0002.txt
seog@Smarty:.../intro_to_bash$ ls -l
total 0
-rwxrwxrwx 1 seog seog 0 Jan  9 23:14 test0001.txt
-rwxrwxrwx 1 seog seog 0 Jan  9 23:14 test0002.txt
seog@Smarty:.../intro_to_bash$
```

File information and permission

```
                              File size
total 0           Owner  Group  Date & time created   File name
-rwxrwxrwx 1 seog seog 0 Jan  9 23:14 test0001.txt
-rwxrwxrwx 1 seog seog 0 Jan  9 23:14 test0002.txt
```

r: readable
w: writable
x: executable

Permission ($ chmod ###)

No permission	---	---	---
Symbol:	rwx	rwx	rwx
Bit:	421	421	421
Triad:	Owner	group	others

```
$ chmod 755 test0001.txt
Owner(rwx) group(r-x) others(r-x)
```

```
$ chmod a+wxr test0001.txt
Owner(rwx) group(rwx) others(rwx)
```

Rename, move & copy directory and files

Users specify items written in ALL CAPS

$ mv OLD NEW   *(move to a new location or rename)*
$ cp TARGET NEWCOPY   *(copy file or folder with new name)*

```
seog@Smarty:.../intro_to_bash$ ls
test0001.txt test0002.txt
seog@Smarty:.../intro_to_bash$ mv test0001.txt renamed0001.txt ; ls
renamed0001.txt  test0002.txt
seog@Smarty:.../intro_to_bash$ cp renamed0001.txt copied0001.txt
seog@Smarty:.../intro_to_bash$ mkdir testFolder
seog@Smarty:.../intro_to_bash$ mv renamed0001.txt testFolder/
seog@Smarty:.../intro_to_bash$ ls
copied0001.txt  test0002.txt  testFolder
seog@Smarty:.../intro_to_bash$ ls -l ./testFolder
total 0
-rwxrwxrwx 1 seog seog 0 Jan  09 23:14 renamed0001.txt
seog@Smarty:.../intro_to_bash$ mv testFolder renamedFolder ; ls
copied0001.txt renamedFolder  test0002.txt
```

Semicolon is a command separator

Remove files or folders

Users specify items written in ALL CAPS

$ rm FILE   *(delete file - **irreversible**)*

$ rm -r FOLDER   *(delete folder recursively - **irreversible***)

$ rmdir FOLDER*   *(delete empty folders matching wildcard*

                                *pattern- **irreversible**)*

> **$ mkdir pattern{0000..0005}**
> **create 6 folders in order**

```
seog@Smarty:.../intro_to_bash$ mkdir testFolder{0000..0005}
seog@Smarty:.../intro_to_bash$ ls
copied0001.txt  test0002.txt    testFolder0001  testFolder0003  testFolder0005
renamedFolder   testFolder0000  testFolder0002  testFolder0004
seog@Smarty:.../intro_to_bash$ rmdir testFolder000*
seog@Smarty:.../intro_to_bash$ ls
copied0001.txt  renamedFolder   test0002.txt
seog@Smarty:.../intro_to_bash$ rm renamedFolder/
rm: cannot remove 'renamedFolder/': Is a directory
seog@Smarty:.../intro_to_bash$ rm -r renamedFolder/ ; ls
copied0001.txt  test0002.txt
```

> *** Wildcard is used to define pattern for search and matches**

Viewing files and editing text

Users specify items written in ALL CAPS

$ echo "TEXT" >> FILE   *(print function and >> store output in a file)*

$ cat FILE   *(concatenate files or printout file content)*

$ less FILE   *(view file interactively, press **q** (quit) viewer)*

$ nano FILE   *(create or edit file in terminal)*

```
seog@Smarty:.../intro_to_bash$ echo "Welcome to MMID Coding Workshop!"
Welcome to MMID Coding Workshop!
seog@Smarty:.../intro_to_bash$ echo "Welcome to MMID Coding Workshop!" >> welcome.txt
seog@Smarty:.../intro_to_bash$ cat welcome.txt
Welcome to MMID Coding Workshop!
seog@Smarty:.../intro_to_bash$ less welcome.txt
```

Viewing files and editing text

Users specify items written in ALL CAPS

$ echo "TEXT" >> FILE   *(print function and >> store output in a file)*

$ cat FILE   *(concatenate files or printout file content)*

$ less FILE   *(view file interactively, press **q** (quit) viewer)*

$ nano FILE   *(create or edit file in terminal)*

```
Welcome to MMID Coding Workshop!
Welcome.txt (END)
```

Viewing files and editing text

Users specify items written in ALL CAPS

$ echo "TEXT" >> FILE   *(print function and >> store output in a file)*

$ cat FILE   *(concatenate files or printout file content)*

$ less FILE   *(view file interactively, press **q** (quit) viewer)*

$ nano FILE   *(create or edit file in terminal)*

```
seog@Smarty:.../intro_to_bash$ echo "Welcome to MMID Coding Workshop!"
Welcome to MMID Coding Workshop!
seog@Smarty:.../intro_to_bash$ echo "Welcome to MMID Coding Workshop!" >> welcome.txt
seog@Smarty:.../intro_to_bash$ cat welcome.txt
Welcome to MMID Coding Workshop!
seog@Smarty:.../intro_to_bash$ less welcome.txt
seog@Smarty:.../intro_to_bash$ nano welcome.txt
```

Viewing files and editing text

Users specify items written in ALL CAPS

$ echo "TEXT" >> FILE   *(print function and >> store output in a file)*

$ cat FILE   *(concatenate files or printout file content)*

$ less FILE   *(view file interactively, press **q** (quit) viewer)*

$ nano FILE   *(create or edit file in terminal)*

```
  GNU nano 4.8
Welcome to MMID Coding Workshop!
Add more text here!




^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     M-U Undo       M-A Mark Text
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell    ^_ Go To Line  M-E Redo       M-6 Copy Text
```

Viewing files and editing text

Users specify items written in ALL CAPS

$ echo "TEXT" >> FILE *(print function and >> store output in a file)*

$ cat FILE *(concatenate files or printout file content)*

$ less FILE *(view file interactively, press q (quit) viewer)*

$ nano FILE *(create or edit file in terminal)*

```
seog@Smarty:.../intro_to_bash$ echo "Welcome to MMID Coding Workshop!"
Welcome to MMID Coding Workshop!
seog@Smarty:.../intro_to_bash$ echo "Welcome to MMID Coding Workshop!" >> welcome.txt
seog@Smarty:.../intro_to_bash$ cat welcome.txt
Welcome to MMID Coding Workshop!
seog@Smarty:.../intro_to_bash$ less welcome.txt
seog@Smarty:.../intro_to_bash$ nano welcome.txt
seog@Smarty:.../intro_to_bash$ cat welcome.txt
Welcome to MMID Coding Workshop!
Add more text here!
```

LEARNING OBJECTIVES

1. *Describe benefits of using command line interface (CLI) over graphical user interface (GUI)*

2. *Install BASH terminal on your operating system*

3. *Navigate, create, rename, move, and copy directory and files in terminal*

4. **Install, execute, and troubleshoot bioinformatics program**

5. *Write a BASH script, change permission, and execute the script*

Necessary updates and builds

Following installations are required to ensure proper program installation

```
seog@Smarty:.../intro_to_bash$ sudo apt-get update
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:2 http://archive.ubuntu.com/ubuntu focal InRelease
...
Fetched 21.2 MB in 34s (628 kB/s)
Reading package lists... Done
seog@Smarty:.../intro_to_bash$ sudo apt-get install build-essential
cc -c -g -Wall -O2 -Wc++-compat  -DHAVE_KALLOC  bseq.c -o bseq.o
cc -c -g -Wall -O2 -Wc++-compat  -DHAVE_KALLOC  sketch.c -o sketch.o
cc -c -g -Wall -O2 -Wc++-compat  -DHAVE_KALLOC  sdust.c -o sdust.o
cc -c -g -Wall -O2 -Wc++-compat  -DHAVE_KALLOC  options.c -o options.o
cc -c -g -Wall -O2 -Wc++-compat  -DHAVE_KALLOC  index.c -o index.o
index.c: In function 'mm_idx_load':
index.c:519:3: warning: ignoring return value of 'fread', declared with attribute warn_unused_result [-Wunused-result]
  519 |    fread(&l, 1, 1, fp);
      |    ^~~~~~~~~~~~~~~~~~~
...
z2_sse2.o ksw2_extd2_sse2.o ksw2_exts2_sse2.o ksw2_dispatch.o
ar: `u' modifier ignored since `D' is the default (see `U')
cc -g -Wall -O2 -Wc++-compat  main.o -o minimap2 -L. -lminimap2 -lm -lz -lpthread
seog@Smarty:.../intro_to_bash$
```

$ sudo apt-get update
$ sudo apt-get install build-essential

Command line program: (i.e.) minimap2

Git clone repository and make: minimap2

From README.md file find: $ git clone https://github.com/lh3/minimap2

This process clones repository to local computer for easy program installation.

```
seog@Smarty:.../intro_to_bash$ rm *
seog@Smarty:.../intro_to_bash$ git clone https://github.com/lh3/minimap2
Cloning into 'minimap2'...
remote: Enumerating objects: 5274, done.
remote: Counting objects: 100% (1359/1359), done.
remote: Compressing objects: 100% (209/209), done.
remote: Total 5274 (delta 1238), reused 1193 (delta 1149), pack-reused 3915
Receiving objects: 100% (5274/5274), 1.67 MiB | 5.71 MiB/s, done.
Resolving deltas: 100% (3806/3806), done.
Updating files: 100% (90/90), done.
seog@Smarty:.../intro_to_bash$ ls ; cd minimap2/ && make
minimap2
cc -c -g -Wall -O2 -Wc++-compat  -DHAVE_KALLOC  main.c -o main.o
cc -c -g -Wall -O2 -Wc++-compat  -DHAVE_KALLOC  kthread.c -o kthread.o
...
ar: `u' modifier ignored since `D' is the default (see `U')
cc -g -Wall -O2 -Wc++-compat  main.o -o minimap2 -L. -lminimap2 -lm -lz -lpthread
seog@Smarty:.../minimap2$
```

Execute minimap2 within the folder

You can now use absolute or relative path to run minimap2.

i.e. when you are in minimap2 folder, run the program: $ ./minimap2

```
seog@Smarty:.../minimap2$ ls
FAQ.md          bseq.o              index.o     ksw2_dispatch.c    ksw2_ll_sse.c    main.o          options.o    sketch.o
LICENSE.txt     code_of_conduct.md  kalloc.c    ksw2_dispatch.o    ksw2_ll_sse.o    map.c           pe.c         splitidx.c
MANIFEST.in     cookbook.md         kalloc.h    ksw2_extd2_sse.c   kthread.c        map.o           pe.o         splitidx.o
Makefile        esterr.c            kalloc.o    ksw2_extd2_sse2.o  kthread.h                         python       sse2neon
Makefile.simde  esterr.o            kdq.h       ksw2_extd2_sse41.o kthread.o        minimap2        sdust.c      test
NEWS.md         example.c           ketopt.h    ksw2_exts2_sse.c   kvec.h           minimap2.1      sdust.h      tex
README.md       format.c            khash.h     ksw2_exts2_sse2.o  lchain.c         misc            sdust.o
align.c         format.o            krmq.h      ksw2_exts2_sse41.o lchain.o         misc.c          seed.c
align.o         hit.c               kseq.h      ksw2_extz2_sse.c   lib              misc.o          seed.o
bseq.c          hit.o               ksort.h     ksw2_extz2_sse2.o  libminimap2.a    mmpriv.h        setup.py
bseq.h          index.c             ksw2.h      ksw2_extz2_sse41.o main.c           options.c       sketch.c
seog@Smarty:.../minimap2$ ./minimap2
Usage: minimap2 [options] <target.fa>|<target.idx> [query.fa] [...]
Options:
  Indexing:
    -H           use homopolymer-compressed k-mer (preferrable for PacBio)
    -k INT       k-mer size (no larger than 28) [15]
    -w INT       minimizer window size [10]
...
See `man ./minimap2.1' for detailed description of these and other advanced command-line options.
seog@Smarty:.../minimap2$
```

Execute minimap2 within the folder

If you want to run minimap2 anywhere without having to assign absolute path to minimap2, copy minimap2 into one of your $PATH directory

```
seog@Smarty:.../minimap2$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
seog@Smarty:.../minimap2$ sudo cp minimap2 /usr/local/bin/
seog@Smarty:.../minimap2$ ls -l /usr/local/bin/
total 1920
drwxr-xr-x 1 root root    4096 Jan 10 18:54 ./
drwxr-xr-x 1 root root    4096 Aug 19 16:40 ../
-rwxr-xr-x 1 root root 1403440 Jan 10 18:54 minimap2*
seog@Smarty:.../minimap2$ cd ../
seog@Smarty:.../intro_to_bash$ minimap2
Usage: minimap2 [options] <target.fa>|<target.idx> [query.fa] [...]
Options:
  Indexing:
    -H           use homopolymer-compressed k-mer (preferrable for PacBio)
    -k INT       k-mer size (no larger than 28) [15]
    -w INT       minimizer window size [10]
...
See `man ./minimap2.1' for detailed description of these and other advanced command-line options.
seog@Smarty:.../intro_to_bash$ rm -r minimap2
```

Download fastq files and reference

Download fastq files and reference file from

https://github.com/MMID-coding-workshop/2022-01-12-Introduction-to-BASH.git

```
seog@Smarty:.../intro_to_bash$ git clone https://github.com/MMID-coding-workshop/2022-01-12-Introduction-to-BASH.git
Cloning into '2022-01-12-Introduction-to-BASH'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 9 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (9/9), 7.68 MiB | 3.55 MiB/s, done.
seog@Smarty:.../intro_to_bash$ ls 2022-01-12-Introduction-to-BASH/data/
1_control_18S_2019_minq7.fastq   1_control_ITS2_2019_minq7.fastq   reference_MN385595_1.fasta
seog@Smarty:.../intro_to_bash$ cd 2022-01-12-Introduction-to-BASH/
seog@Smarty:.../2022-01-12-Introduction-to-BASH$ mkdir aligned ; cd aligned
seog@Smarty:.../aligned$
```

Map reads using minimap2

Press TAB to auto-complete unique file/folder names to speed-up typing

1. Create an index
2. Map reads against reference genome

```
seog@Smarty:.../aligned$ minimap2 -d ../data/reference_MN385595_1.mmi ../data/reference_MN385595_1.fasta
[M::mm_idx_gen::0.002*0.00] collected minimizers
[M::mm_idx_gen::0.004*0.00] sorted minimizers
[M::main::0.007*0.00] loaded/built the index for 1 target sequence(s)
[M::mm_idx_stat] kmer size: 15; skip: 10; is_hpc: 0; #seq: 1
[M::mm_idx_stat::0.008*0.00] distinct minimizers: 216 (100.00% are singletons); average occurrences:
1.000; average spacing: 5.671; total length: 1225
[M::main] Version: 2.24-r1122
[M::main] CMD: minimap2 -d ./data/reference_MN385595_1.mmi ./data/reference_MN385595_1.fasta
[M::main] Real time: 0.016 sec; CPU: 0.000 sec; Peak RSS: 0.002 GB

seog@Smarty:.../aligned$ ls ../data/
1_control_18S_2019_minq7.fastq  1_control_ITS2_2019_minq7.fastq  reference_MN385595_1.fasta
reference_MN385595_1.mmi

seog@Smarty:.../aligned$
```

Map reads using minimap2

Press TAB to auto-complete unique file/folder names to speed-up typing

1. Create an index
2. Map reads against reference genome

```
seog@Smarty:.../aligned$ minimap2 -a ../data/reference_MN385595_1.mmi
../data/1_control_18S_2019_minq7.fastq > alignment.sam
[M::main::0.004*0.00] loaded/built the index for 1 target sequence(s)
[M::mm_mapopt_update::0.005*0.00] mid_occ = 10
[M::mm_idx_stat] kmer size: 15; skip: 10; is_hpc: 0; #seq: 1
[M::mm_idx_stat::0.006*0.00] distinct minimizers: 216 (100.00% are singletons); average occurrences:
1.000; average spacing: 5.671; total length: 1225
[M::worker_pipeline::0.218*2.15] mapped 18357 sequences
[M::main] Version: 2.24-r1122
[M::main] CMD: minimap2 -a ../data/reference_MN385595_1.mmi ../data/1_control_18S_2019_minq7.fastq
[M::main] Real time: 0.219 sec; CPU: 0.469 sec; Peak RSS: 0.019 GB
seog@Smarty:.../aligned$ ls -l
total 9232
drwxrwxrwx 1 gseo gseo     4096 Jan 10 04:02 ./
drwxrwxrwx 1 gseo gseo     4096 Jan 10 03:58 ../
-rwxrwxrwx 1 gseo gseo  9452653 Jan 10 04:02 alignment.sam*
seog@Smarty:.../aligned$
```

Troubleshooting

Troubleshooting - installation

Often when you get an error during installation, it could be due to missing dependency. Run following commands to install all required packages

```
seog@Smarty:.../intro_to_bash$ cd minimap2
seog@Smarty:.../minimap2$ make
cc -c -g -Wall -O2 -Wc++-compat  -DHAVE_KALLOC  main.c -o main.o
cc -c -g -Wall -O2 -Wc++-compat  -DHAVE_KALLOC  kthread.c -o kthread.o
cc -c -g -Wall -O2 -Wc++-compat  -DHAVE_KALLOC  kalloc.c -o kalloc.o
cc -c -g -Wall -O2 -Wc++-compat  -DHAVE_KALLOC  misc.c -o misc.o
cc -c -g -Wall -O2 -Wc++-compat  -DHAVE_KALLOC  bseq.c -o bseq.o
bseq.c:1:10: fatal error: zlib.h: No such file or directory
    1 | #include <zlib.h>
      |          ^~~~~~~~
compilation terminated.
make: *** [Makefile:41: bseq.o] Error 1
seog@Smarty:.../minimap2$
```

$ sudo apt-get update

$ sudo apt-get install build-essential

Troubleshooting – missing file

When you are unsure of the error message, search it online!
Someone probably had the same problem and found the solution.

```
seog@Smarty:.../aligned$ minimap2 -a ./reference_MN385595_1.mmi
../data/1_control_18S_2019_minq7.fastq > alignment.sam
[ERROR] failed to open file './reference_MN385595_1.mmi': No such file or directory
```

Check your path for each files

```
seog@Smarty:.../aligned$ minimap2 -v ./reference_MN385595_1.mmi
../data/1_control_18S_2019_minq7.fastq > alignment.sam
[ERROR] missing input: please specify a query file to map or option -d to keep the index
```

Check if proper options were used

LEARNING OBJECTIVES

1. *Describe benefits of using command line interface (CLI) over graphical user interface (GUI)*

2. *Install BASH terminal on your operating system*

3. *Navigate, create, rename, move, and copy directory and files in terminal*

4. *Install, execute, and troubleshoot bioinformatics program*

5. **Write a BASH script, change permission, and execute the script**

BASH scripting

Create a file with .sh extension inside intro_to_bash folder

You can create it using a notepad, VS Code or use text editors on terminal

i.e. one of the following:

$ touch SCRIPT.sh

$ nano SCRIPT.sh

```
seog@Smarty:.../intro_to_bash$ touch script.sh
```

BASH scripting: add shebang

#!/usr/bin/env bash   *(shebang, for recognizing which shell was used to write the script)*

```
C: > Users > seog > Desktop > intro_to_bash > script.sh
#!/usr/bin/env bash

# Anything after the pound sign become comments
# shebang can also be written as: #!/bin/bash

echo "Welcome to MMID Coding Workshop!" # echo function prints message inside ""
```

```
seog@Smarty:.../intro_to_bash$ chmod 755 script.sh
seog@Smarty:.../intro_to_bash$ ./script.sh
Welcome to MMID Coding Workshop!
seog@Smarty:.../intro_to_bash$
```

BASH scripting: adding variables

Add elements to the script (scroll down) + save script & run

```
C: > Users > seog > Desktop > intro_to_bash > script.sh
echo "Welcome to MMID Coding Workshop!" # echo function prints message inside ""

# set variables – use absolute path for directories!
BASE_DIR="/mnt/c/Users/seog/Desktop/intro_to_bash/2022-01-12-Introduction-to-BASH"
DATA_DIR="/mnt/c/Users/seog/Desktop/intro_to_bash/2022-01-12-Introduction-to-BASH/data"
OUTPUT_DIR="$BASE_DIR/script_output"
INPUT="1_control_18S_2019_minq7.fastq"
REFERENCE="reference_MN385595_1.fasta"

echo $BASE_DIR # print out variable
```

> **Variables: abstract storage location where information can be saved and accessed later**

```
seog@Smarty:.../intro_to_bash$ ./script.sh
Welcome to MMID Coding Workshop!
/mnt/c/Users/seog/Desktop/intro_to_bash/2022-01-12-Introduction-to-BASH
seog@Smarty:.../intro_to_bash$ echo $BASE_DIR

seog@Smarty:.../intro_to_bash$
```

BASH scripting: cd and create directory

Add elements to the script (scroll down) + save script & run

```
C: > Users > seog > Desktop > intro_to_bash > script.sh
BASE_DIR="/mnt/c/Users/seog/Desktop/intro_to_bash/2022-01-12-Introduction-to-BASH"
DATA_DIR="/mnt/c/Users/seog/Desktop/intro_to_bash/2022-01-12-Introduction-to-BASH/data"
OUTPUT_DIR="$BASE_DIR/script_output"
INPUT="1_control_18S_2019_minq7.fastq"
REFERENCE="reference_MN385595_1.fasta"

# echo $BASE_DIR # print out variable
cd $BASE_DIR # you can also use ${BASE_DIR}
mkdir -p "script_output" # create folder if it doesn't exist -p
echo "hi" # just random word to verify script output
pwd # print script's current working directory
```

```
seog@Smarty:.../intro_to_bash$ ./script.sh
Welcome to MMID Coding Workshop!
hi
/mnt/c/Users/seog/Desktop/intro_to_bash/2022-01-12-Introduction-to-BASH
seog@Smarty:.../intro_to_bash$
```

BASH scripting: add minimap2 jobs

Add elements to the script (scroll down) + save script & run

C: > Users > seog > Desktop > intro_to_bash > script.sh

```
# Run minimap2 and save output result into $OUTPUT_DIR
# Copy and paste previous command used but use variables for input files
# minimap2 -d ../data/reference_MN385595_1.mmi ../data/reference_MN385595_1.fasta
minimap2 -d $DATA_DIR/reference_MN385595_1.mmi $DATA_DIR/$REFERENCE


# minimap2 -a ../data/reference_MN385595_1.mmi ../data/1_control_18S_2019_minq7.fastq >
alignment.sam
minimap2 -a $DATA_DIR/reference_MN385595_1.mmi $DATA_DIR/$INPUT > $OUTPUT_DIR/alignment.sam


echo "the end!"
exit
```

```
seog@Smarty:.../intro_to_bash$ ./script.sh
Welcome to MMID Coding Workshop!
...
[M::main] Real time: 0.495 sec; CPU: 0.875 sec; Peak RSS: 0.019 GB
the end!
seog@Smarty:.../intro_to_bash$
```

BASH scripting: minimap2 output

Final BASH script output printed on command line interface

```
[M::mm_idx_gen::0.002*0.00] collected minimizers
[M::mm_idx_gen::0.004*0.00] sorted minimizers
[M::main::0.0
[M::mm_idx_st
[M::mm_idx_st                                                    1.000; average
spacing: 5.67
[M::main] Ver
[M::main] CMD
[M::main] Real time: 0.016 sec; CPU: 0.000 sec; Peak RSS: 0.002 GB

[M::main::0.004*0.00] loaded/built the index for 1 target
[M::mm_mapopt_update::0.005*0.00] mid_occ = 10
[M::mm_idx_stat] kmer size: 15; skip: 10; is_hpc: 0; #seq
[M::mm_idx_stat::0.006*0.00] distinct minimizers: 216 (100
spacing: 5.671; total length: 1225
[M::worker_pipeline::0.218*2.15] mapped 18357 sequences
[M::main] Version: 2.24-r1122
[M::main] CMD: minimap2 -a ../data/reference_MN385595_1.mm
[M::main] Real time: 0.219 sec; CPU: 0.469 sec; Peak RSS:
```

Manual minimap2 analysis result "mapped 18357 sequences"

```
Welcome to MMID Coding Workshop!
hi
/mnt/c/Users/gseo/Desktop/intro_to_bash/2022-01-12-Introduction-to-BASH
[M::mm_idx_gen::0.011*1.44] co
[M::mm_idx_gen::0.014*1.14] so
[M::main::0.019*0.85] loaded/b
[M::mm_idx_stat] kmer size: 15
[M::mm_idx_stat::0.020*0.79] d
ing: 5.671; total length: 1225
[M::main] Version: 2.24-r1122
[M::main] CMD: minimap2 -d /mn
95_1.mmi /mnt/c/Users/gseo/Des
[M::main] Real time: 0.044 sec
[M::main::0.008*1.92] loaded/b
[M::mm_mapopt_update::0.009*1.68] mid_occ = 10
[M::mm_idx_stat] kmer size: 15; skip: 10; is_hpc: 0; #seq: 1
[M::mm_idx_stat::0.010*1.58] distinct minimizers: 216 (100.00% are singletons); average occurrences: 1.000; average spac
ing: 5.671; total length: 1225
[M::worker_pipeline::0.490*1.7 ] mapped 18357 sequences
[M::main] Version: 2.24-r1122
[M::main] CMD: minimap2 -a /mnt/c/Users/gseo/Desktop/intro_to_bash/2022-01-12-Introduction-to-BASH/data/reference_MN3855
95_1.mmi /mnt/c/Users/gseo/Desktop/intro_to_bash/2022-01-12-Introduction-to-BASH/data/1_control_18S_2019_minq7.fastq
[M::main] Real time: 0.495 sec; CPU: 0.875 sec; Peak RSS: 0.019 GB
the end!
```

Script automated minimap2 analysis result "mapped 18357 sequences"

Complete BASH script

```bash
C: > Users > seog > Desktop > intro_to_bash > script.sh
#!/usr/bin/env bash

# Anything after the pound sign become comments
# shebang can also be written as: #!/bin/bash

echo "Welcome to MMID Coding Workshop!" # echo function prints message inside ""

# set variables – use absolute path for directories!
BASE_DIR="/mnt/c/Users/seog/Desktop/intro_to_bash/2022-01-12-Introduction-to-BASH"
DATA_DIR="/mnt/c/Users/seog/Desktop/intro_to_bash/2022-01-12-Introduction-to-BASH/data"
OUTPUT_DIR="$BASE_DIR/script_output"
INPUT="1_control_18S_2019_minq7.fastq"
REFERENCE="reference_MN385595_1.fasta"

# echo $BASE_DIR # print out variable
cd $BASE_DIR # you can also use ${BASE_DIR}
mkdir -p "script_output" # create folder if it doesn't exist -p
echo "hi" # just random word to verify script output
pwd # print script's current working directory

# Run minimap2 and save output result into $OUTPUT_DIR
# Copy and paste previous command used but use variables for input files
# minimap2 -d ../data/reference_MN385595_1.mmi ../data/reference_MN385595_1.fasta
minimap2 -d $DATA_DIR/reference_MN385595_1.mmi $DATA_DIR/$REFERENCE

# minimap2 -a ../data/reference_MN385595_1.mmi ../data/1_control_18S_2019_minq7.fastq > alignment.sam
minimap2 -a $DATA_DIR/reference_MN385595_1.mmi $DATA_DIR/$INPUT > $OUTPUT_DIR/alignment.sam

echo "the end!"
exit
```
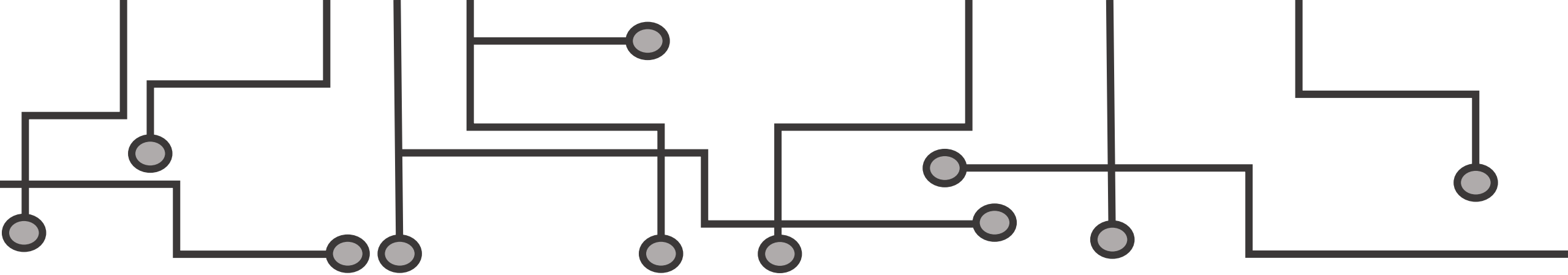
LEARNING OBJECTIVES

1. **Describe benefits of using command line interface over GUI**

2. **Install BASH terminal on your operating system**

3. **Navigate, create, rename, move, and copy directory and files in terminal**

4. **Install, execute, and troubleshoot bioinformatics program**

5. **Write a BASH script, change permission, and execute the script**

HELPFUL RESOURCES

1. *What is CLI* https://www.hostinger.com/tutorials/what-is-cli
2. *BASH manual:* https://www.gnu.org/software/bash/manual/bash.html
3. *Information on Linux folder structure* https://www.howtogeek.com/117435/htg-explains-the-linux-directory-structure-explained/
4. **The BASH Guide:* https://guide.bash.academy/
5. **Learn Enough Command-Line to be dangerous (free first few chapters):* https://www.learnenough.com/command-line-tutorial

YouTube Videos
1. **Joe Collins - Beginner's Guide to the Bash Terminal:* https://www.youtube.com/watch?v=oxuRxtrO2Ag
2. **Traversy Media – Shell Scripting Crash Course – Beginner Level:* https://www.youtube.com/watch?v=v-F3YLd6oMw

THANK YOU FOR ATTENDING!
The Q&A Session will now begin.

Please make sure to fill out the Exit Survey
We value your feedback!

More questions? Please email us at
mmid.coding.workshop@gmail.com or post them to the workshop slack channel