

---

# Introduction to Compute Canada HPC

*Miistakis Institute Workshop*

*June 6, 2025*

*Sara Smith, MRU | [ssmith6@mtroyal.ca](mailto:ssmith6@mtroyal.ca)*

---

# Agenda

- Introductions
- Understanding command line basics
- Writing and running simple scripts
- Navigating and understanding the Compute Canada (Alliance) HPC environment





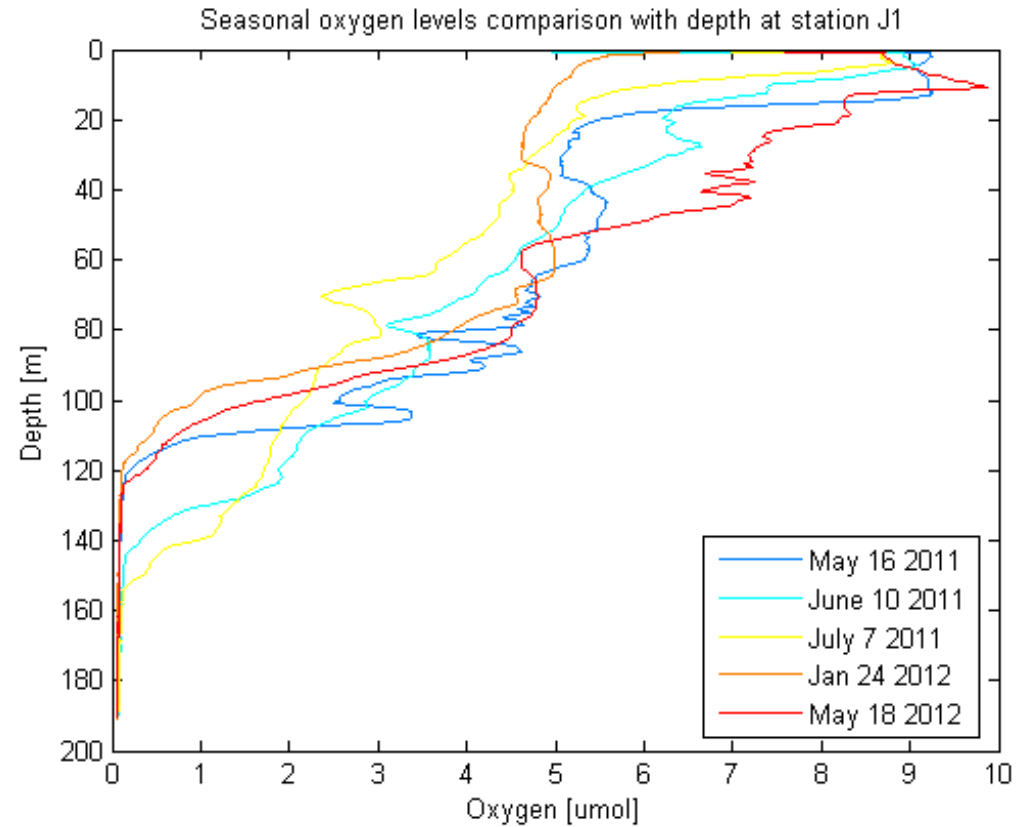
---

Assistant Professor, Ecology & Evolution  
Biology Research Coordinator

Consequences of climate change on species  
distributions & persistence

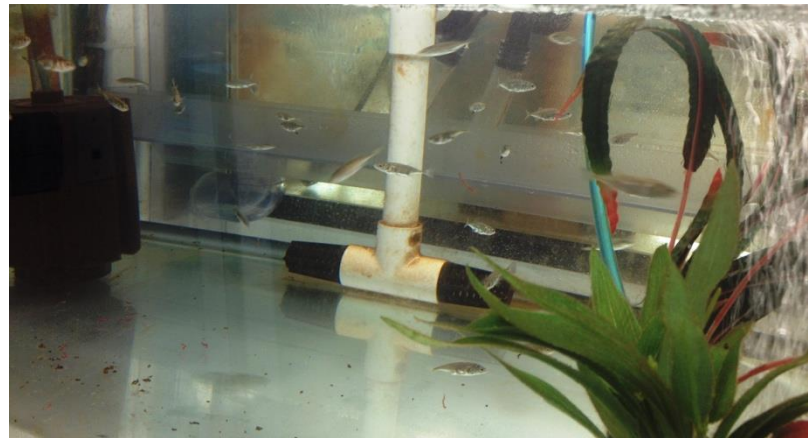
Bioinformatic analyses of population  
genomics

Biostatistics/Applied Data Analysis, Ecology,  
Field Research, Scientific Writing &  
Communication

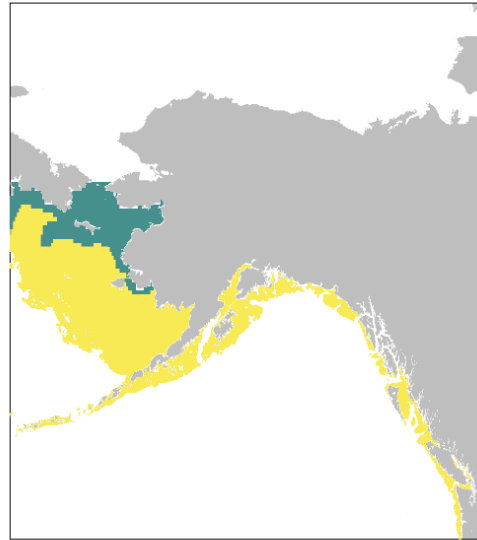


# How a Field Biologist Realized She Should Learn How to Code

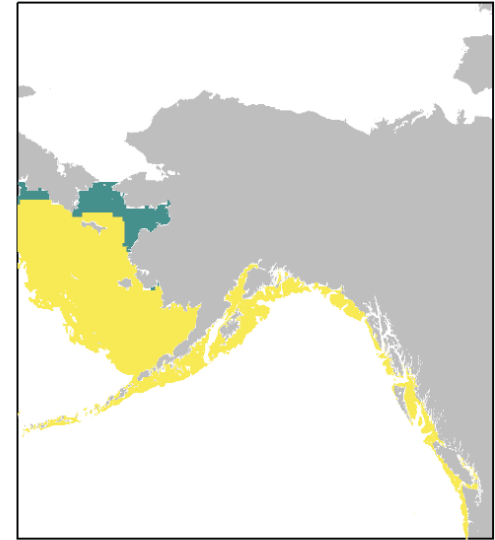




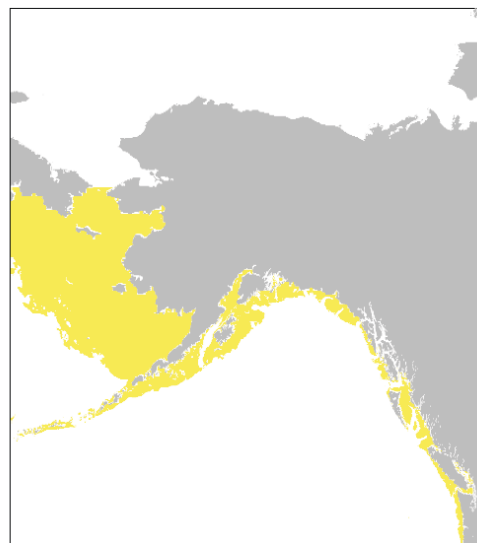
b) RCP 4.5



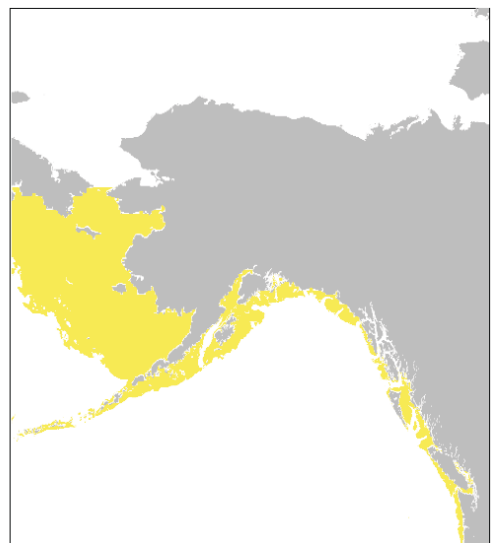
c) RCP 8.5

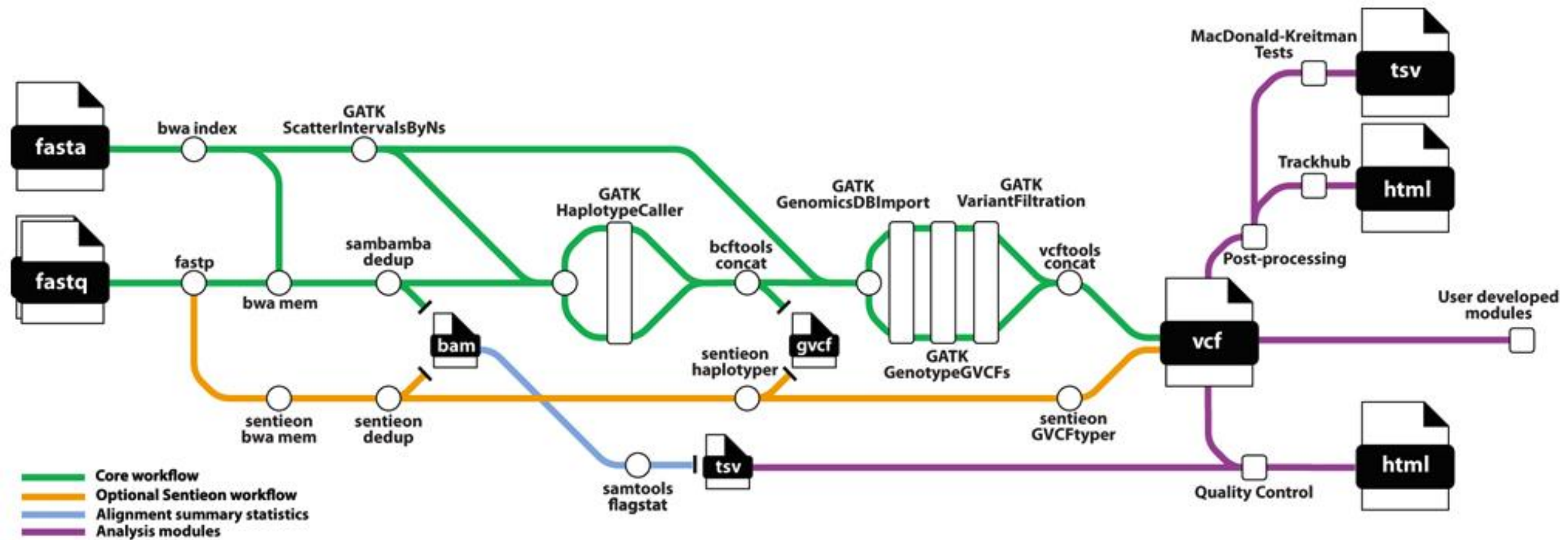


d) RCP 4.5



e) RCP 8.5





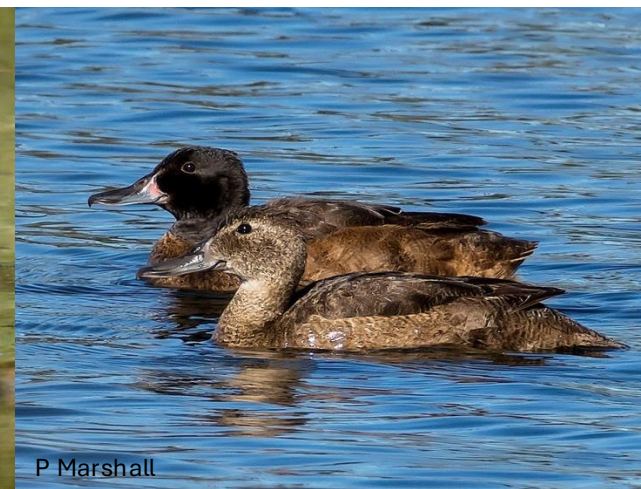
A Spencer



C Moring



J Clark



P Marshall



---

# Your Turn

(turn your camera on if possible/comfortable)

- Name
- City
- Current/past projects or research interests
- Why you are interested in using Compute Canada

---

# Before we start ...

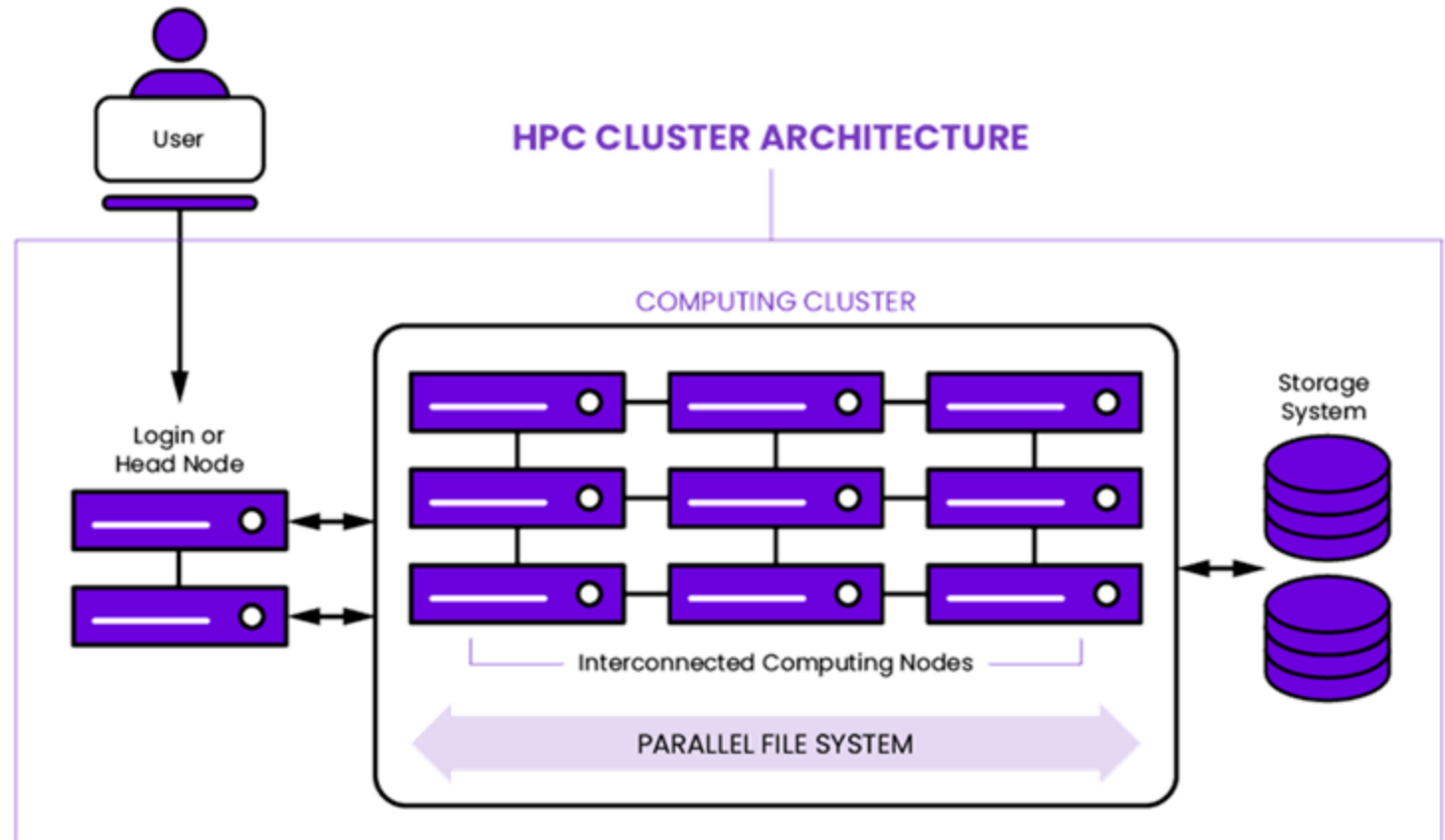
Was everyone able to complete the pre-work?

Any issues/concerns?



# What is a High-Performance Computing (HPC) cluster?

- A series of parallel, interconnected compute nodes
- Accessed through login nodes
- **Architecture** (the way the nodes are connected) varies between clusters



---

# Why work on HPC clusters?

- Your computer takes a task that you give it and divides that task into several steps. Each step has to be completed before the next step can start
- HPC clusters can support **parallel computing**, where a task can be split into multiple steps that can run at the same time
- Even if you are not taking advantage of parallel computing, HPC clusters tend to have more memory & storage and can run your tasks more efficiently

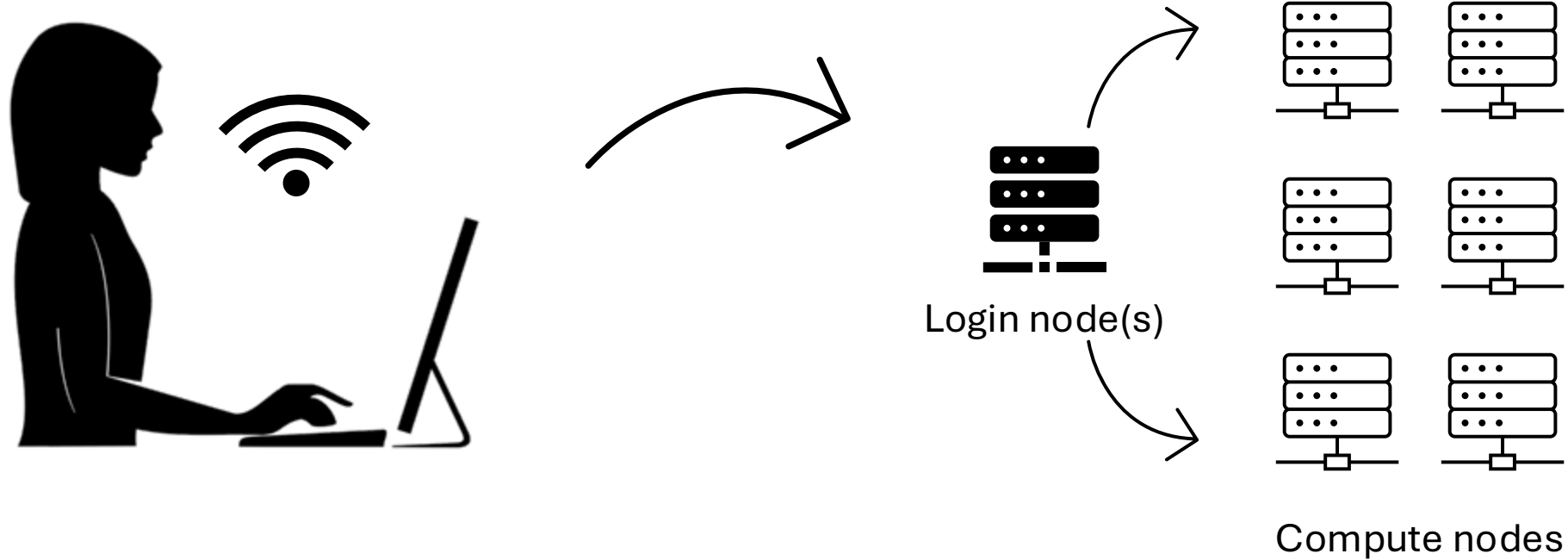
---

# ... How do we work on an HPC cluster?

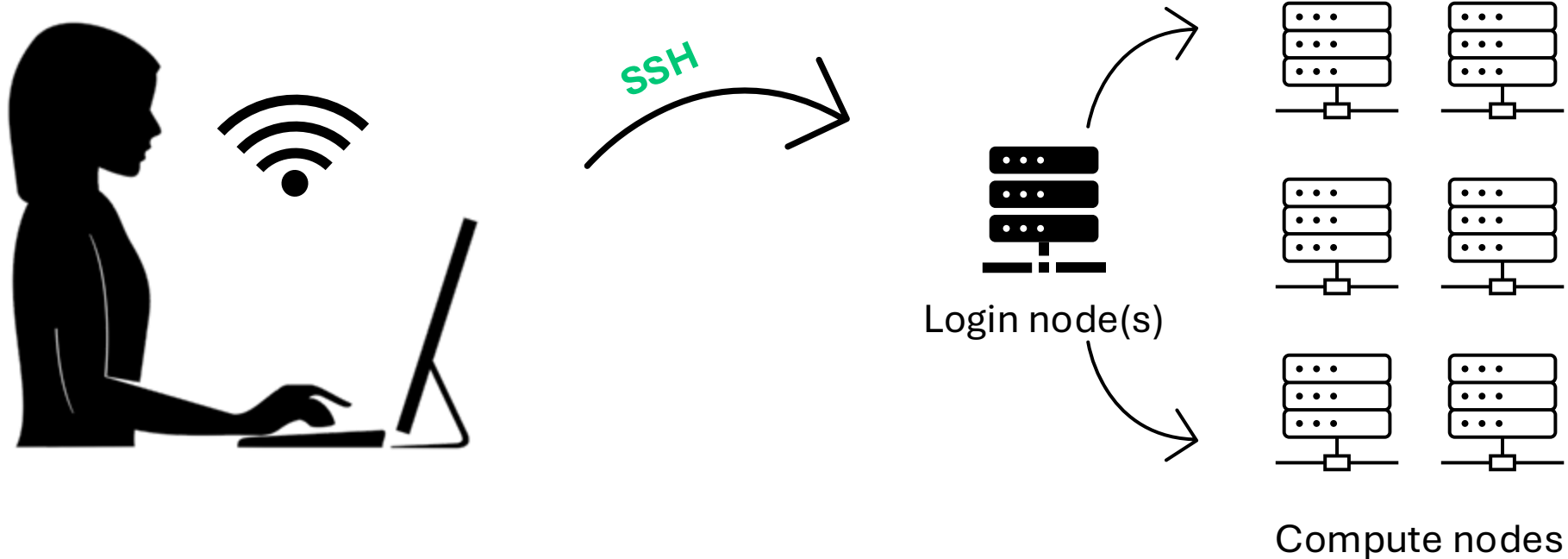
- We can work on an HPC cluster *interactively* or *non-interactively*
- We access HPC clusters predominantly through a **command line interface (CLI)**
  - Text-based, fast, replicable/reproducible/repeatable
- This differs from a **graphical user interface (GUI)**
  - Visual, intuitive, manual



# Talking to Compute Canada



# Talking to Compute Canada (via **Secure SHell**)



---

# Over to the Compute Canada (Alliance) cluster

[https://docs.alliancecan.ca/wiki/Technical\\_documentation](https://docs.alliancecan.ca/wiki/Technical_documentation)

\$ssh <username>@cedar.computecanada.ca

```
Last login: Wed Jun  4 18:32:52 on ttys005
|>ssh sjsmith@cedar.computecanada.ca
|(sjsmith@cedar.computecanada.ca) Password:
|(sjsmith@cedar.computecanada.ca) Duo two-factor login for sjsmith

Enter a passcode or select one of the following options:

1. Duo Push to Sara's iPhone (iOS)
2. Duo Push to Sara's iPhone 16 Pro (iOS)

Passcode or option (1-2): |
```



---

```
cd projects/def-<nilosusername>
```

```
mkdir <user>
```

```
cd <newdir>
```

```
# walk through GitHub repo quickly
```

```
$git clone https://github.com/sjswuitchik/miistakis_workshop.git
```

```
cd miistakis_workshop
```

```
touch testfile.txt
```

```
ls
```

```
nano testfile.txt
```

```
This is a line from the terminal
```

```
cntrl-o / return / cntrl-x
```

---

ls

cat testfile.txt

cat test <tab>

rm testfile.txt

cd data/

pwd

ls

ls -lh

cd ..

pwd

---

```
cd scripts/
```

```
chmod +x 01_wordcount.sh
```

```
less 01_wordcount.sh
```

```
## explain the shebang line - Shebang tells the shell to execute it via bash shell; it's  
simply an absolute path to the bash interpreter
```

```
## walk through the script and what it does but leave out what `wc -l` means
```

```
q # to exit a file viewed by `less`
```

```
man wc
```

```
## edit 01_wordcount.sh to be `for i in ../data/*.txt`
```

```
./01_wordcount.sh (or `bash 01_wordcount.sh` or `sh 01_wordcount.sh`)
```



---

# Basic commands

*pwd* – print working directory

*ls* – list files

*cd* – change directory

*mkdir* – make new directory

*touch* – create new file

*cp, mv, rm* – copy, move, delete files **\*\*\*\* be careful with rm, it's forever \*\*\*\***

**Tab completion** for efficiency

**Arrow keys** (↑/↓) for command history

*man* for manuals

---

# BioBreak

*Take 5 and we'll come back to more scripts*

---

```
less 02_slurm_test.sh
```

```
# SLURM resources https://slurm.schedmd.com/documentation.html
```

```
## SLURM is the workload manager/job scheduler that allocates resources on CC
```

```
# walk through each line of 02_slurm_test.sh
```

```
## edit the necessary lines for the 02_slurm_test.sh script to make sure it runs on  
your (Nilo's) allocation
```

```
sbatch 02_slurm_test.sh
```

```
queue --user=<yourusername>
```

---

# How to use software on Compute Canada

[https://docs.alliancecan.ca/wiki/Available\\_software](https://docs.alliancecan.ca/wiki/Available_software)

---

```
module load <modulename>
```

```
module load r/4.5.0
```

```
R
```

```
## works exactly like your R on your computer
```

```
getwd()
```

```
x <- c(1,2,3)
```

```
x
```

```
quit()
```

```
n
```



---

```
less 03_pkg_install.R
```

```
# walk through repo designation
```

```
Rscript 03_pkg_install.R
```

```
## while installing, open a new tab, sign into CC again, navigate to GH dir, and  
explore 04_addCol.R and 05_run_r_test.sh, edit 05_run_r_test.sh to run on Nilo's  
allocation
```

```
# once tidyverse has installed, run script
```

```
sbatch 05_run_r_test.sh
```

```
# explore output
```

---

```
# open 04_addCol.R in other tab
```

```
R
```

```
library(tidyverse)
```

```
df <- read_delim("../data/minData.csv", delim = ',') %>% mutate(spp = "G.  
aculeatus")
```

```
write_delim(df, "minData.clean.inter.csv")
```

```
quit()
```

```
n
```

```
## look at new output
```

---

# transferring files with Secure File Transfer Protocol

# open new tab

sftp <username>@cedar.computecanada.ca

# password and 2FA

# 'l' in front of everything is your **local** computer e.g., `lcd` is changing your directory on your computer, `cd` is changing your directory on the cluster

lpwd

pwd

lls

ls

---

lcd Desktop

# switch to other tab

pwd

# copy path, switch to `sftp` tab

cd <pastedpath>

get minData.clean.inter.csv

# check Desktop for the new file, open it, change it, save it with a new name, close it

put minData.clean.inter.changed.csv

# check out and open new file on cluster

---

# so why do bash scripting of your work?

# explore the example script to showcase the combination of command line programs and R, and consider how useful it is to not have to switch back and forth constantly

---

# Other important things to explore on your own

Interactive jobs: ``salloc``

Key elements of SLURM scripts: time, memory, CPUs required

- `--account:` your allocation
- `--time:` runtime limit
- `--mem:` memory requested
- `--cpus-per-task:` CPU threads
- `--output:` log file



---

# Other important things to explore on your own

Interactive jobs: `salloc`

Key elements of SLURM scripts: time, memory, CPUs required

`sbatch job.sh` – submit job

`squeue -u <youruser>` – view queue

`sacct` – check job history

`seff <jobid>` – job efficiency

`scancel <jobid>` – cancel job

---

# Other important things to explore on your own

Interactive jobs: ``salloc``

Key elements of SLURM scripts: time, memory, CPUs required

Storage & compute nodes: scratch, projects, home

Anaconda environments: reproducible version control for your packages and tools

<https://swcarpentry.github.io/shell-novice/>



**Questions?**