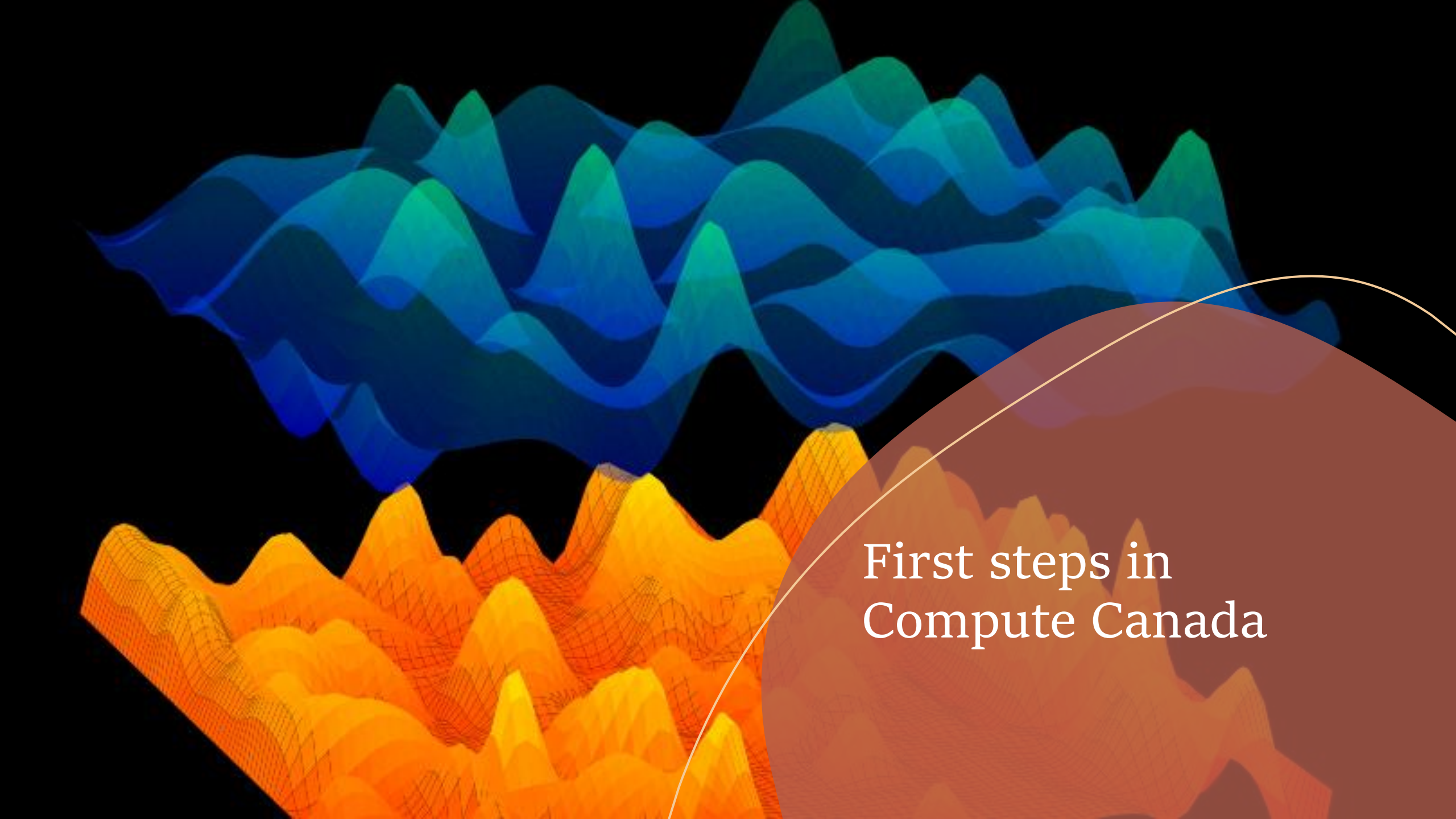


First steps in Compute Canada

INTRODUCTION

- What it is
- Why we need it
- How we use it

My_first_job.sl

- Log in
- Navigate folder
- Upload data
- Submit job
- Download results

Resources and example code are here: https://github.com/BIAPT/Tutorial_Compute_Canada.git

Preparation:

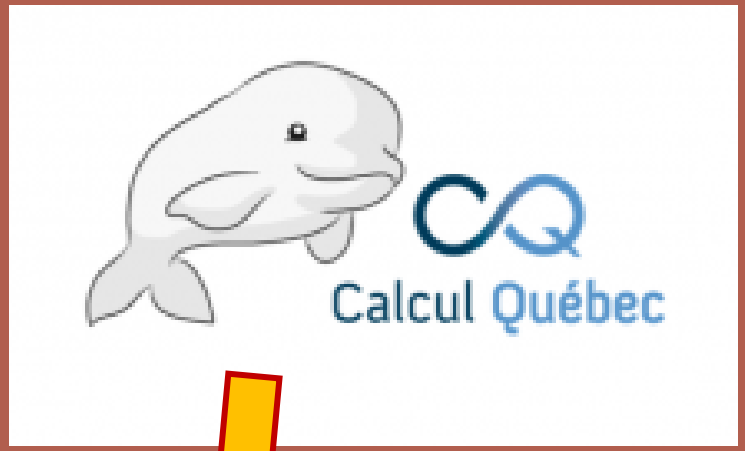
- **Windows users need to install this**:

  https://ubuntu.com/tutorials/ubuntu-on-windows#1-overview

  (follow the steps on the website and set up a password)

- Resources and example code are here: https://github.com/BIAPT/Tutorial_Compute_Canada.git
- For the detailed documentation, please refer to this: https://docs.computecanada.ca/wiki/Getting_started
- For further, more technical tutorials, have a look at this: https://www.youtube.com/watch?v=J9VCHe1ovBg

What it is

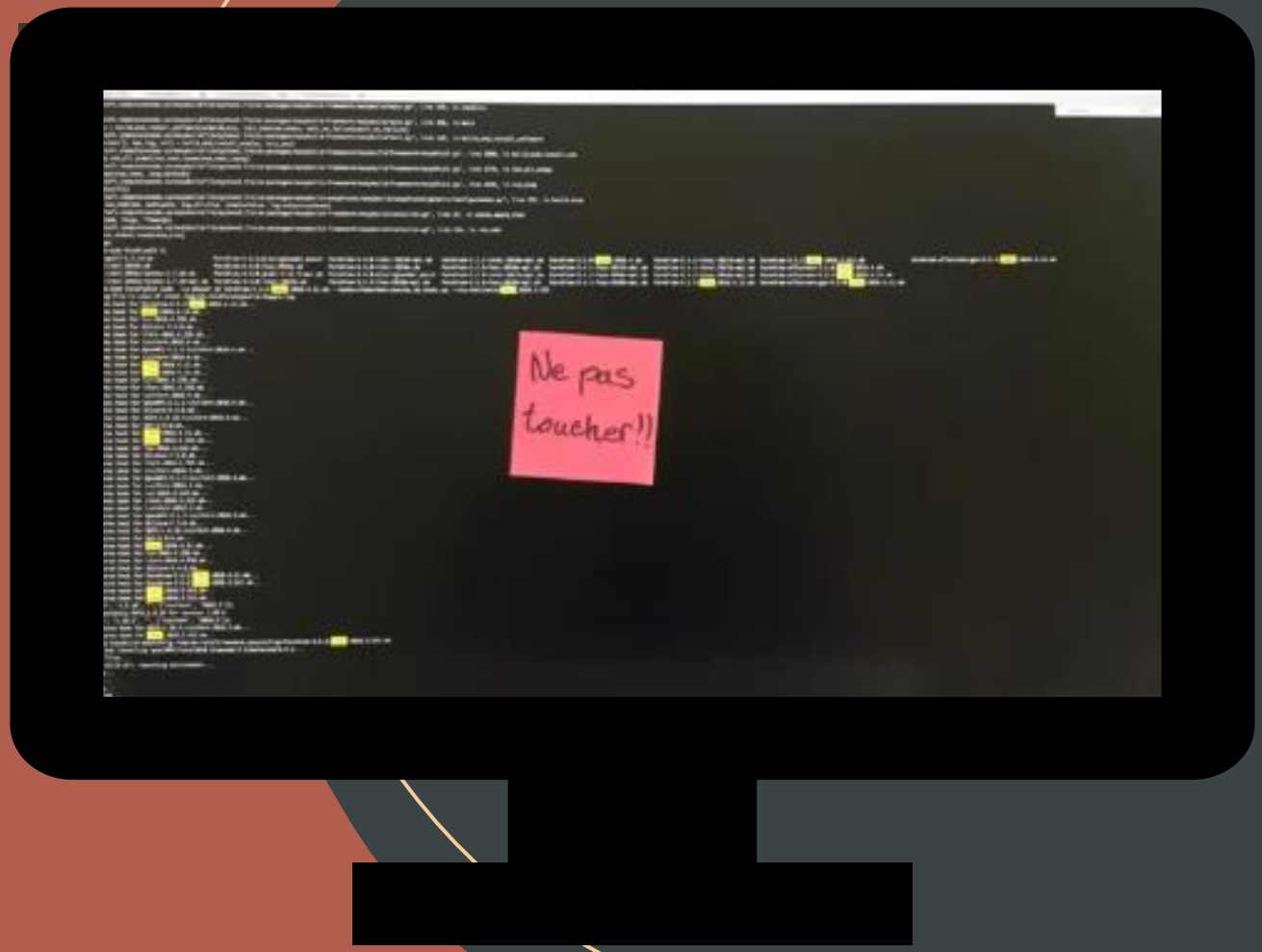

CQ
Calcul Québec


compute
canada
calcul
canada

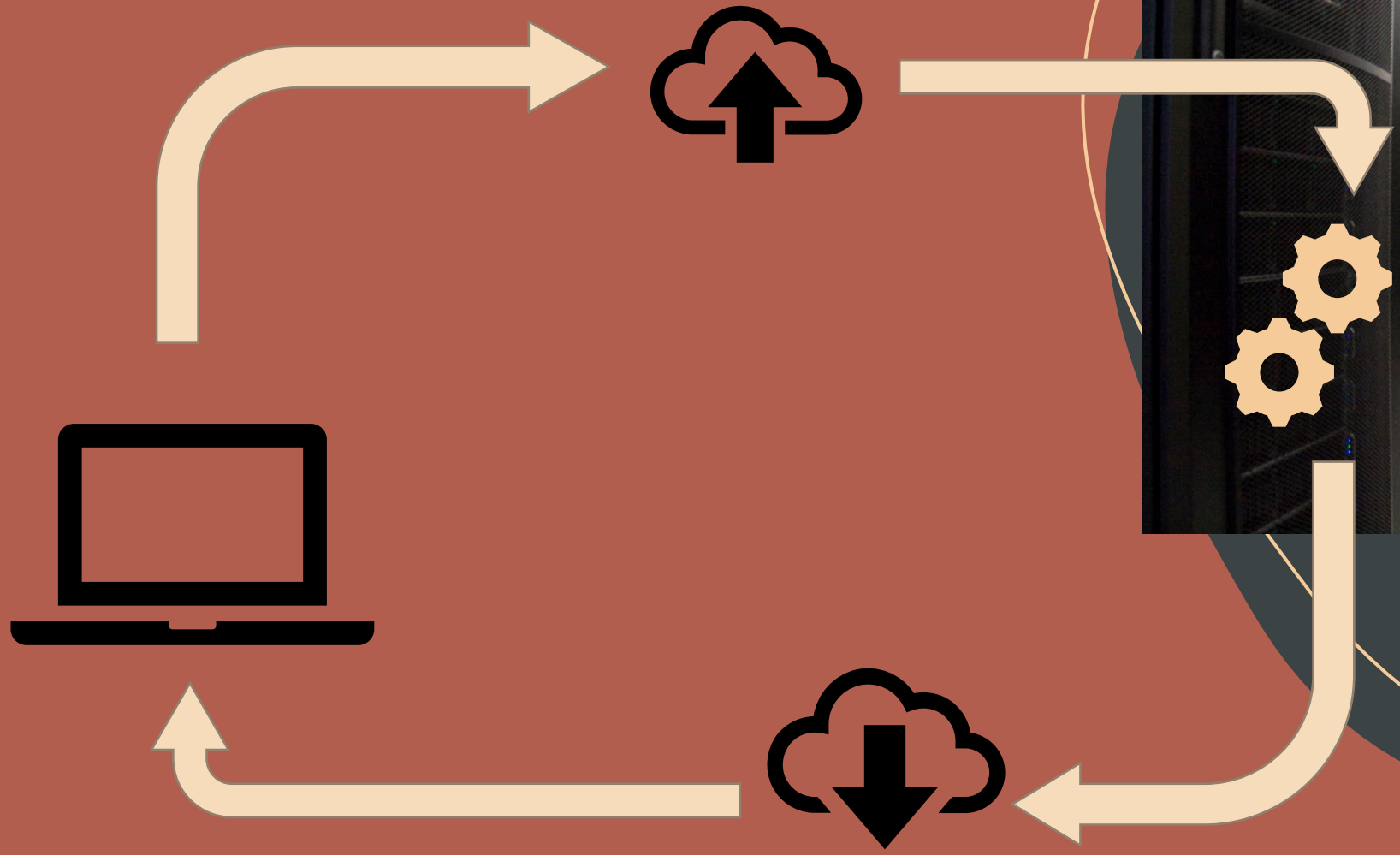

WestGrid
Compute Ontario
Calcul Québec
Acenet

Why we need it

- Large amount of data
- Large amount of computation
- Limited time
- Limited resources



How we use it



Log in:

WINDOWS:

- Open UBUNTU Terminal
- type:

```
lotte@Horst:~$ ssh lotte@beluga.computecanada.ca
```

Your account name *The cluster in Montréal*

- Type your password (hidden text)
- Press ENTER

MAC:

- open terminal
- type:

```
lotte@Horst:~$ ssh lotte@beluga.computecanada.ca
```

Your account name *The cluster in Montréal*

- Type your password (hidden text)
- Press ENTER

Log out:

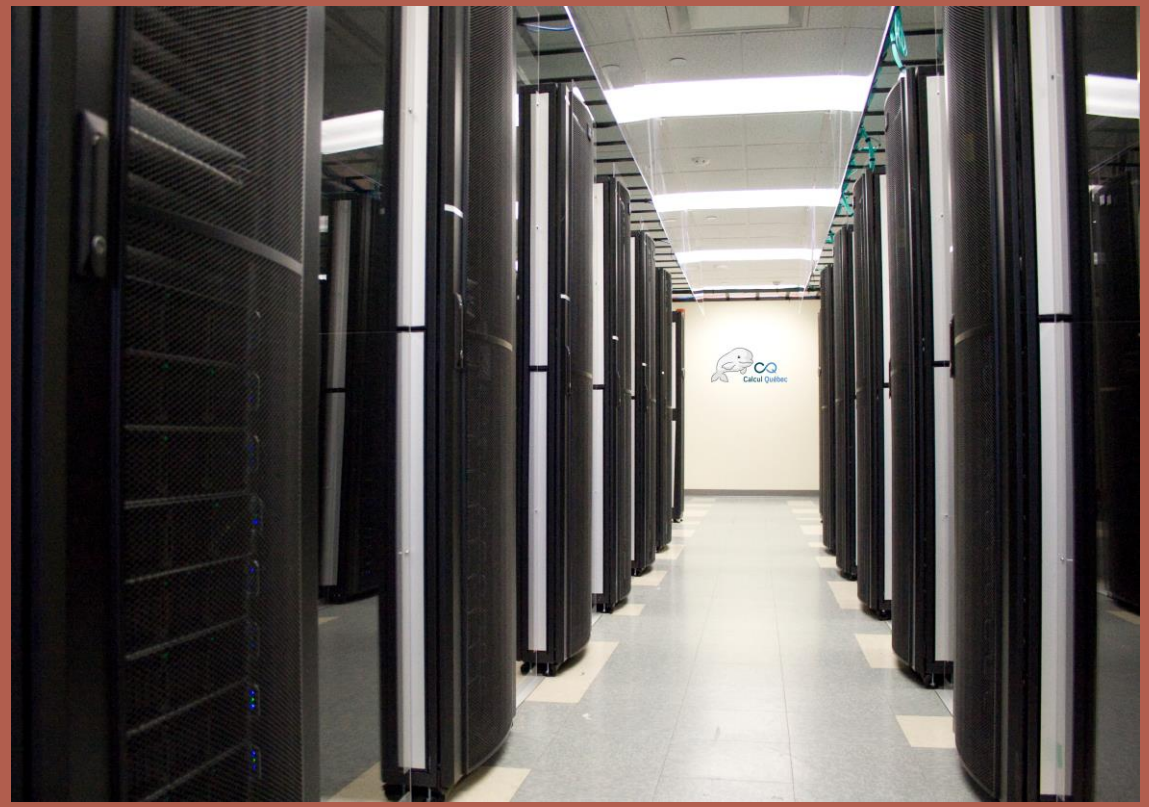
```
[lotte@beluga3 Dim_DOC]$ exit
logout
Connection to beluga.computecanada.ca closed.
lotte@Horst:/mnt/c/Users/User/Desktop$
```

Log in:

Welcome to Beluga!

Log in:

Welcome to Beluga!



Navigate folders:

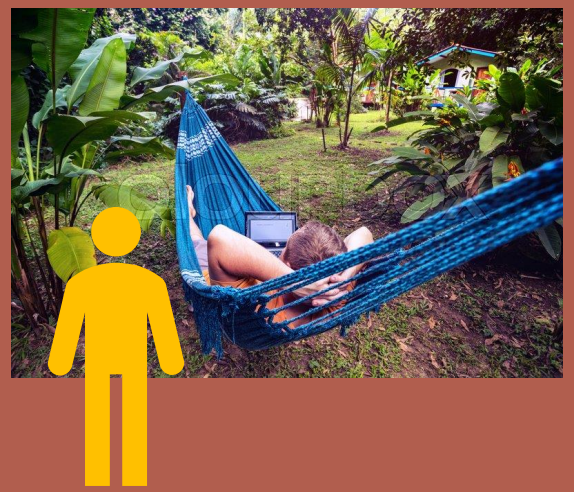
Navigation commands

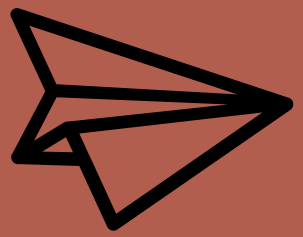
- pwd — *print working directory*
- cd — *change directory (go to)*
- *cd ..* — *Change directory (step outside)*
- ls — *list*
- mkdir — *make directory*
- rm — *remove (add -r if directory)*

→ Navigate to your personal project folder and create a directory called “Test_project” (don’t use spaces or / or . In a foldername)

Prepare code:

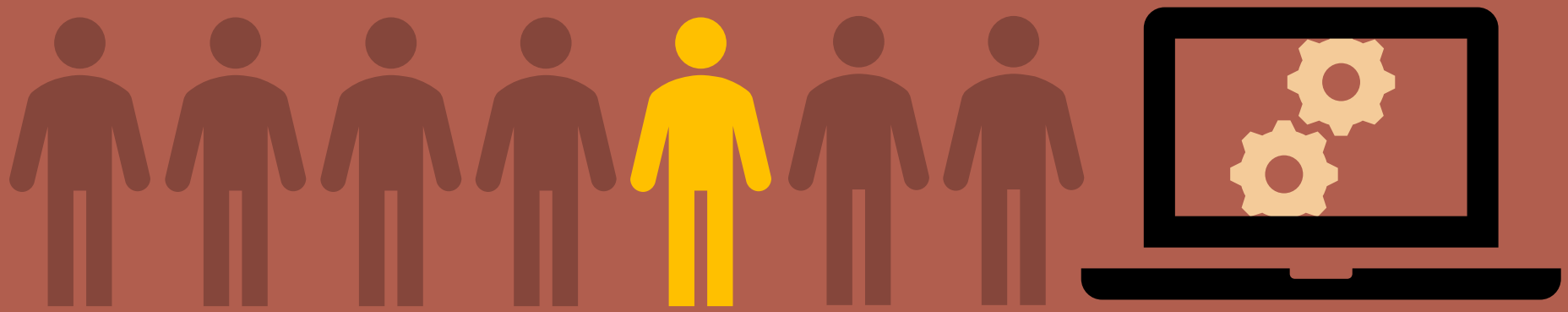
- Each job needs a corresponding **slurm** file.
- This specifies how much time and power your job needs and gives you a place in the waiting line
- Never try to run jobs on the login node (it will make everything really slow)





Prepare code:

- Each job needs a corresponding **slurm** file.
- This specifies how much time and power your job needs and gives you a place in the waiting line



→ Prepare your own slurm files

Prepare code:

README | README | CC_test.sl | CC_test2.sl

```
#!/bin/bash -l
#SBATCH --job-name=testComputeCanada
#SBATCH --account=def-sblain # adjust this to match the accounting group you are using to submit jobs
#SBATCH --time=0-00:10:00      # adjust this to match the walltime of your job (D-HH:MM:SS)
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=40      # adjust this if you are using parallel commands
#SBATCH --mem=90000          # adjust this according to the memory requirement per node you need (this is MegaByte)
#SBATCH --mail-user=q2h3s6p4k0e9o7a5@biaptlab.slack.com
#SBATCH --mail-type=ALL

# Choose a version of MATLAB by loading a module:
module load matlab/2018a

# Create temporary job info location
mkdir -p /scratch/$USER/$SLURM_JOB_ID

srun matlab -nodisplay -r "CC_test"

# Cleanup
rm -rf /scratch/$USER/$SLURM_JOB_ID
```

Any name to recognize your job

Time Limit

Nr of Nodes (stays at 1 for now)

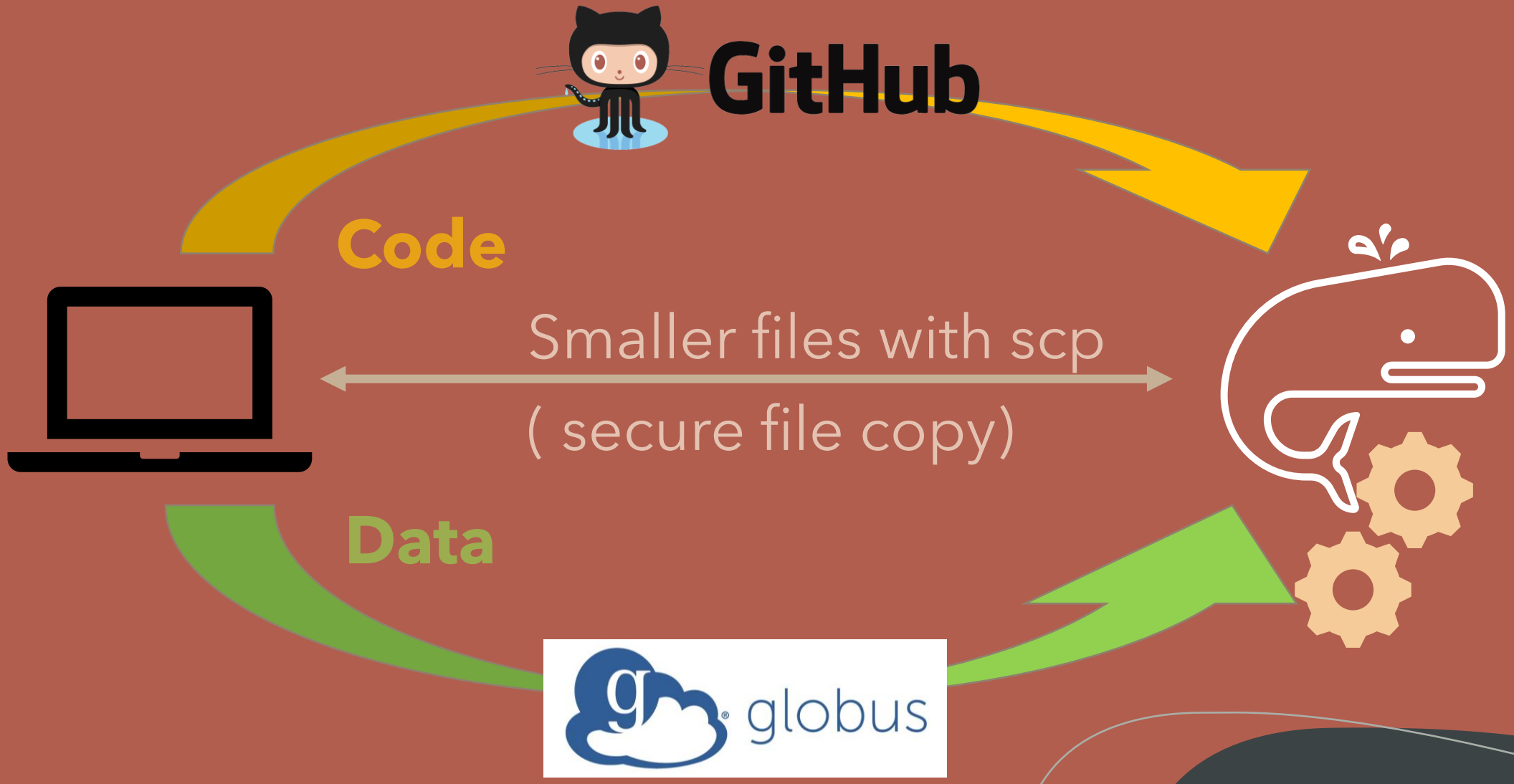
Your E-Mail

Matlab verison

How your .m is called

→ Prepare your own slurm files (open the .sl with a text editor)

Upload data/code:



Upload data/code:

scp from to

scp oldfile newfile

```
scp test.txt lotte@beluga.computecanada.ca:projects/
def-sblain/lotte/test.txt
```

→ Move your slurm and test matlab script into your folder

Upload data/code:

helpful commands

- unzip *name*.zip *unzip folder*
- zip -r name.zip name zip folder

→ Move your slurm and test matlab script into your folder

Submit job:

sacct --format JobID,State -j 29231822 | grep TIMEOUT

- Log in your CC account and move to the directory of your code

→ submit your job

```
sbatch yourjobname.sl
```

→ Check the status of your job

```
lotte@beluga1 ~]$ sq
             JOBID     USER      ACCOUNT           NAME  ST  TIME_LEFT NODES CPUS TRES_PER_N MIN_MEM NODELIST (REASON)
          15383554    lotte def-sblain_c testComputeCan  PD       10:00     1   40        N/A  90000M  (Priority)
          15383558    lotte def-sblain_c test2ComputeCa  PD       10:00     1   40        N/A  90000M  (Priority)
```

→ Check the progress of your job / Check why it crashed

```
cat yourjobid.out
```

Do some other stuff

Do some other stuff

Slackbot 4

Do some other stuff

Slackbot 4

Slackbot 3:26 PM
Charlotte Maschke | Email

slurm@calculquebec.ca (SLURM workload manager)
Slurm Job_id=15350790 Name=make-dPLI Began, Queued time 00:00:02
(No content)

Slackbot 5:29 PM
Charlotte Maschke | Email

slurm@calculquebec.ca (SLURM workload manager)
Slurm Job_id=15350790 Name=make-dPLI Ended, Run time 02:03:41, COMPLETED, ExitCode 0
(No content)

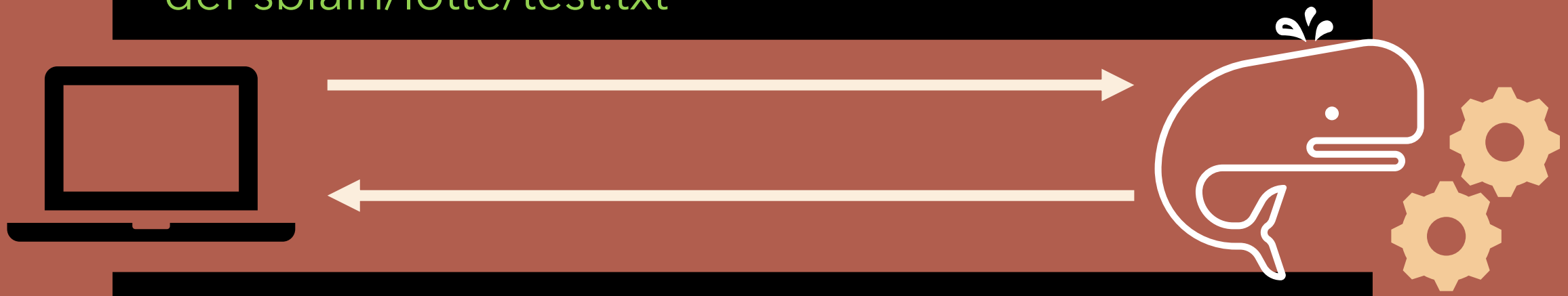
Download results :

Log out from Beluga and use the scp command on your computer

scp from to

scp oldfile newfile

```
scp test.txt lotte@beluga.computecanada.ca:projects/
def-sblain/lotte/test.txt
```



```
scp lotte@beluga.computecanada.ca:projects/
def-sblain/lotte/test.txt test.txt
```

→ Download the output.txt to your computer

When to use it?

- *Always try on your own Computer first !*

  (use a reduced dataset …)
- *Debug on your own Machine*

<u>ONCE you have a finished pipeline</u>

- *Parallelize the code (search the bottleneck)*
- *When piloting a job on CC always start with a small number and time (proof of concept for additional debugging)*

How to specify your needs?

"seff JOB_ID

```
[lotte@beluga4 4_Neuroimage_submission_2021]$ seff 19464051
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
        LANGUAGE = (unset),
        LC_ALL = (unset),
        LANG = "C.UTF-8"
    are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").
Job ID: 19464051
Cluster: beluga
User/Group: lotte/lotte
State: COMPLETED (exit code 0)
Nodes: 1
Cores per node: 40
CPU Utilized: 00:02:28
CPU Efficiency: 2.00% of 02:03:20 core-walltime
Job Wall-clock time: 00:03:05
Memory Utilized: 214.71 MB
Memory Efficiency: 0.24% of 87.89 GB
[lotte@beluga4 4_Neuroimage_submission_2021]$
```

Check the efficiency of your job and adapt it for next time !

(it is ok to overestimate time a little)

How to use packages?

- *Look whether the package you want to use is available on CC (pre-installed)*

  https://docs.computecanada.ca/wiki/Available_software

- *If yes, you can use it like this:*

```
#SBATCH --mail-type=FAIL


module load python/3.9
module load scipy-stack
```

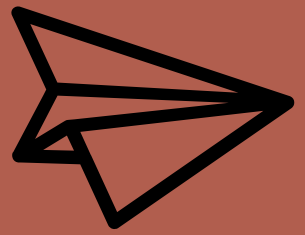
Inside your slurm (.sl) file

How to use packages?

- *If No, need to install it yourself*

PIP install

Login NODE



SLURM file

PIP install

How to use packages?

- If No, need to install it yourself

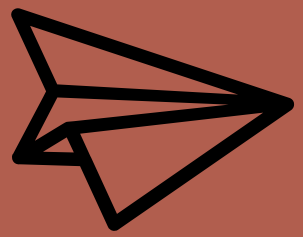
1. Create a venv on your login node

2. Send it in your slurm file to the compute node

3. Use it on compute node

PIP install

Login NODE



SLURM file

How to use packages?

- If No, need to install it yourself

1. Create a venv on your login node

https://docs.computecanada.ca/wiki/Python#Creating_and_using_a_virtual_environment

Load the python version you need

Create your venv

```
[lotte@beluga4 Tutorial_Compute_Canada]$ module load python/3.9
[lotte@beluga4 Tutorial_Compute_Canada]$ virtualenv --no-download test_env
created virtual environment CPython3.9.6.final.0-64 in 43659ms
  creator CPython3Posix(dest=/lustre03/project/6010672/lotte/Tutorial_Compute_Canada/test_env,
)
  seeder FromAppData(download=False, pip=latest, setuptools=latest, wheel=latest, via=copy, app
cal/share/virtualenv/seed-app-data/v1.0.1)
  activators BashActivator,CShellActivator,FishActivator,PowerShellActivator,PythonActivator,Xo
[lotte@beluga4 Tutorial_Compute_Canada]$
[lotte@beluga4 Tutorial_Compute_Canada]$
```

How to use packages?

```
[lotte@beluga4 Tutorial_Compute_Canada]$ source test_env/bin/activate
(test_env) [lotte@beluga4 Tutorial_Compute_Canada]$
```

Activate your venv

You are now in your venv

… and you can start installing your packages

```
(test_env) [lotte@beluga4 Tutorial_Compute_Canada]$ pip install mne
Ignoring pip: markers 'python_version < "3"' don't match your environment
Looking in links: /cvmfs/soft.computecanada.ca/custom/python/wheelhouse/gentoo/avx512, /cvmfs/soft.computecanada.ca/cus
om/python/wheelhouse/gentoo/avx2, /cvmfs/soft.computecanada.ca/custom/python/wheelhouse/gentoo/generic, /cvmfs/soft.com
utecanada.ca/custom/python/wheelhouse/generic
Collecting mne
  Downloading mne-0.24.1-py3-none-any.whl (7.4 MB)
     |████████████████████████████████| 7.4 MB 58.7 MB/s
Processing /cvmfs/soft.computecanada.ca/custom/python/wheelhouse/gentoo/generic/scipy-1.7.1+computecanada-cp39-cp39-lin
x_x86_64.whl
Processing /cvmfs/soft.computecanada.ca/custom/python/wheelhouse/gentoo/generic/numpy-1.21.2+computecanada-cp39-cp39-li
ux_x86_64.whl
Installing collected packages: numpy, scipy, mne
```

How to use packages?

- *If No, need to install it yourself*

```
#SBATCH --mail-user=q2h3s6p4k0e9o7a5@biaptlab.slack.com
#SBATCH --mail-type=FAIL


module load python/3.9
module load scipy-stack
source aperiodic_env/bin/activate

python _1_spectral_decomposition.py data_aperiodic results_aperiodic data_aperiodic/data_Baseline.txt Base
```

2. Send it in your slurm file to the compute node

END