To connect: ssh <your username>@graham.computecanada.ca

Graham and cedar are the two servers I use, graham has slightly more resources but cedar has a matlab license. To connect to cedar its just ssh [user@cedar.computecanada.ca](mailto:user@cedar.computecanada.ca)

To get it to remember you follow these instructions <https://serverfault.com/questions/241588/how-to-automate-ssh-login-with-password>

In short on your local terminal just go

ssh-keygen -t rsa

You can leave all the things it asks you for blank I think.

Then just

ssh-copy-id user@graham.blah.blah

Use ls+cd to navigate, all the things we’ll care about we’ll do in the projects folder (scratch is for temporary stuff, it has a lot of space and is fast, but gets deleted often). So I go to projects/def-jwerker/kjslakov/… to do things there, you guys will have similar paths to yours. From there clone the github repo onto the computecanada server

git clone <https://github.com/karlslakov/jwlab_eeg.git>

I then made a data folder alongside the repo, but not in the repo (cause obviously we don’t wanna upload data to github). So I have projects/.../kjslakov/jwlab and projects/.../kjslakov/data. You’ll have to change paths where necessary, it’ll be kind of annoying, but I don’t have a great solution for it now. Just change the paths as needed, it will need changes in the .sh job files and in the run/computecanada\_constants.py.

Next you need to set up a way to edit this stuff remotely. You should get a file transfer program (compute canada recommends Globus <https://docs.computecanada.ca/wiki/Globus>, but yeah use whatever), and also an editor that can have SSH filesystems/file editing, I use visual studio code with the Remote Development extension (<https://code.visualstudio.com/docs/remote/remote-overview>), but yeah there are lots of things that would work.

When you want to run things, you’ll have to use the file transfer program to upload the relevant data to the compute canada server first and make sure the paths are correct like I mentioned earlier. For example, if you want to run the all\_cleaning.m stuff, you need to upload the raw .set files; if you want to do classification, then upload ml\_df\_readys.pkl, etc. (Remember that to run Matlab you need to use cedar, or you could try to connect graham to ubc license server (<https://docs.computecanada.ca/wiki/MATLAB>) if you want.

Running Jobs - <https://docs.computecanada.ca/wiki/Running_jobs>

That doc has lots of good info you’ll want to read, but quickly to run one of the already existing jobs I’ve made:

1. Cd to the folder with the job you wanna run (classification/jobs probably)
2. sbatch <job .sh file>
3. Done! Type sq to check its status. I set most of the jobs to put their output in the output folder, so to see how it’s running you can type “cat output/<name of output file.out>” or just “cat output/\*” if its the only thing in there. After running your job I recommend moving it to saved\_outputs with a descriptive name if it’s important to keep the output log, otherwise just delete the output file, to keep things clean.

All these jobs just run scripts in the classification/code/jwlab/run folder, that do the actual work, look in those files to see what they are doing/change them. To make a new job, copy a .py file in the run folder (and edit it to do the thing you want), and then copy a .sh file in the jobs folder, specify the cpu/memory/time requirements, extra module installs, and change the target in the “python /$HOME/…” line to run your new file.

**Important Change to Imports**

Before importing things in a notebook, you now have to do

**import setup\_jwlab**

And instead of just saying

**from ml\_prep import load\_ml\_df**, you have to add a **jwlab.** So like

**from jwlab.ml\_prep import load\_ml\_df**

And there’s something similar you have to do in the /run/ .py files:

import run\_setup

**Running Jupyter Notebook**

**To Install**

Follow these instructions :

<https://docs.computecanada.ca/wiki/Jupyter#Installing_Jupyter_Notebook>

Afterwards, with the virtualenv activated, you’ll have to pip install all the modules,

pip install numpy

pip install pandas

pip install scikit-learn

**To Run**

<https://docs.computecanada.ca/wiki/Jupyter#Activating_the_environment>

module load python/3.6

source $HOME/jupyter\_py3/bin/activate

salloc --time=1:0:0 --ntasks=1 --cpus-per-task=2 --mem-per-cpu=1024M --account=def-<THIS IS PROBABLY JCAMPBELL?> srun $VIRTUAL\_ENV/bin/notebook.sh

You can increase memory or cpus if you need, as always. Once that’s running it’ll say something like

The Jupyter Notebook is running at:

http://cdr544.int.cedar.computecanada.ca:8888/?token=7ed7059fad64446f837567e32af8d20efa72e72476eb72ca

Copy the cdr544.int.cedar.computecanada.ca:8888 part out. You need to set up a SSH tunnel there. This will be different on mac than it was for me on windows, but computecanadas docs seem pretty good <https://docs.computecanada.ca/wiki/Jupyter#Connecting_to_Jupyter_Notebook>. Then you should be able to open up localhost:8888 on your browser and it’ll connect.

Hopefully, this is enough to get things running basically, I’m sure there will be some issues, but yeah computecanada has good documentation, and you can email too, maybe I can help.