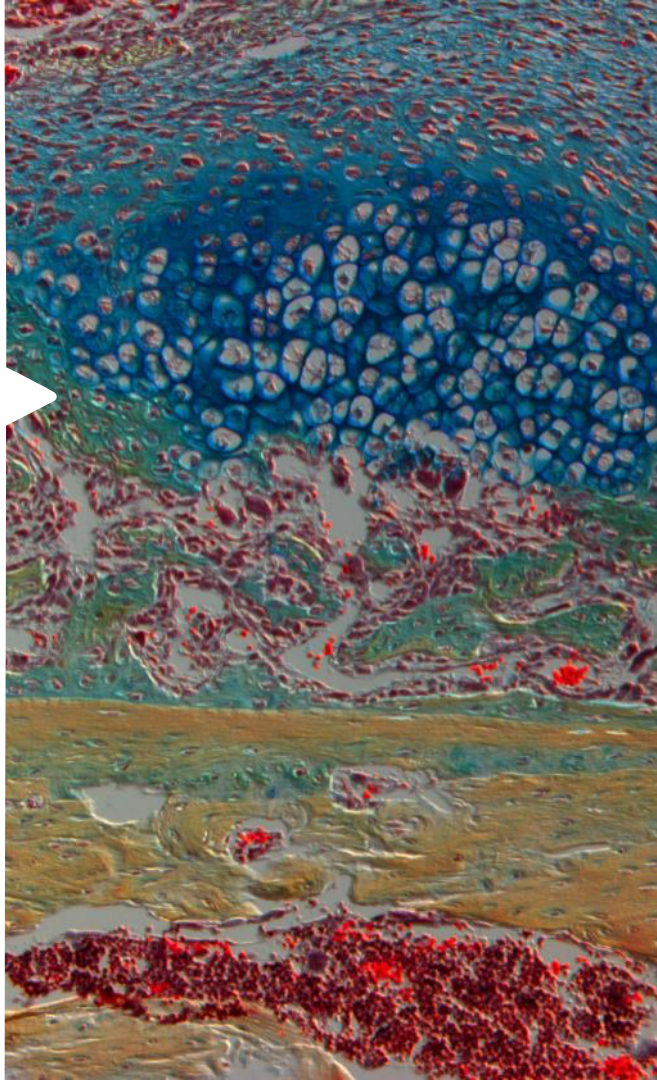
NYU Langone Health

Working on Big Purple

Mark Grivainis

08 October 2020



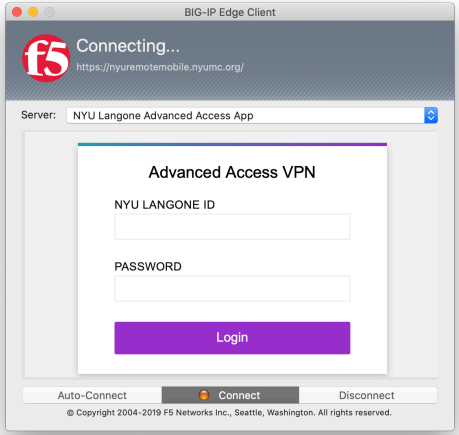
Content

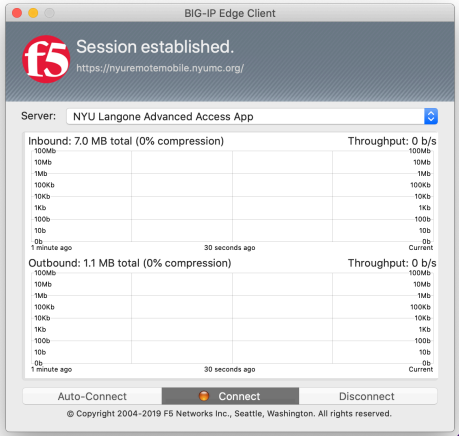
- Connecting to the VPN
- Connecting to BP
- Loading modules
- Running interactive jobs
  - Running a script in interactive mode
- Installing Python Packages
- Shell Scripts
- Running a Batch job

NYU Langone Health

Connecting to the VPN

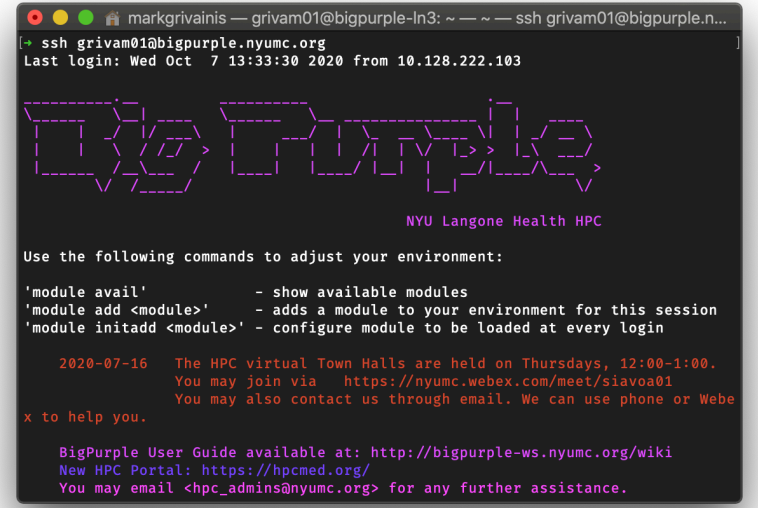
- Open the F5 App
- Enter your Kerberos ID and Password
- Complete the 2 – Factor authentication
- You are now logged in





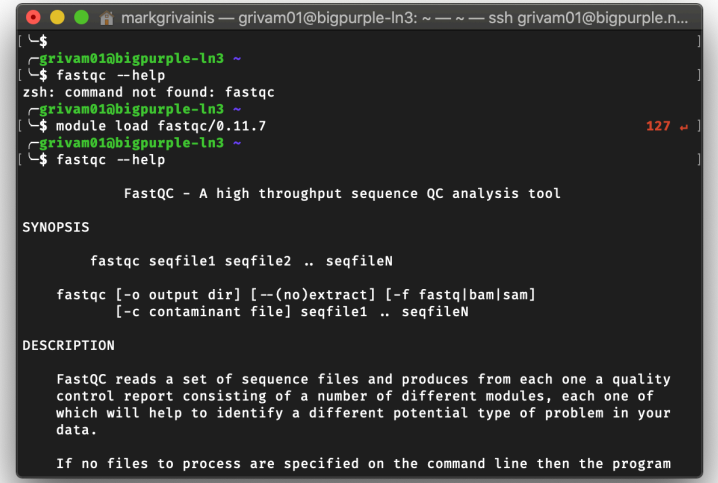
Log into Big Purple

- Open your teminal/powershell
- Enter:
  - ssh <kid>@bigpurple.nyumc.org
- You should now be in your home directory



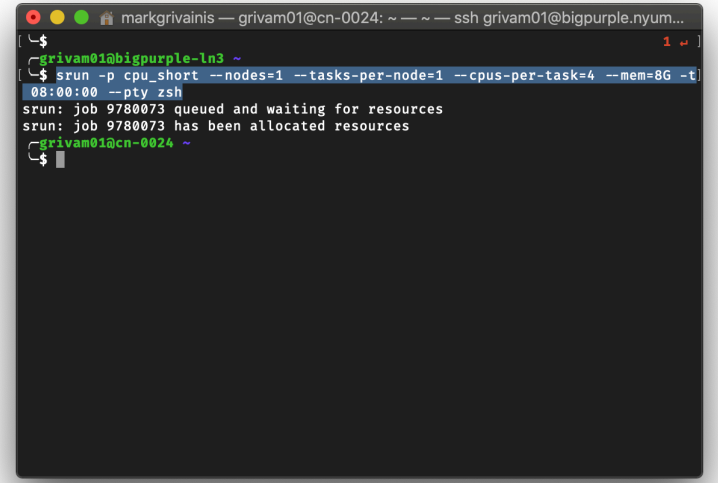
Loading Modules

- If you do not know exactly what module you need use:
  - module avail
- If a module is not loaded you will get an error:
  - Command not found: <module_name>
- Load a module using:
  - module load <module_name>/version/…



Running an interactive job

- Use the ‘srun’ slurm command
  - `srun -p cpu_short --nodes=1 --tasks-per-node=1 --cpus-per-task=4 --mem=8G -t 08:00:00 --pty bash`
- -p: the type of node
- --nodes: How many nodes (use 1)
- --tasks-per-node: How many tasks (use 1)
- --cpus-per-task: How many CPUs to assign
  - This varies depending on what you are doing
  - For alignment use 8 or 16
  - For trimming / fastqc use 4 or 8 (remember to set the flag when calling the program)



BBMap CPU/Ntask comparisons

- 1 node; 1 task; 8 CPUs per task
  - 21978.373 seconds
- 1 node; 1 task; 16 CPUs per task
  - 10641.761 seconds.
- 1 node; 1 task; 1 CPUs per task
  - > 12 hours
- 1 node; 8 tasks; 1 CPUs per task
  - 24770.645 seconds.

NYU Langone Health

Running a Module in Interactive Mode

- Now that you are no longer on the head node you can run any command line applications you load
- Remember to read the documentation
  - Do this by adding the --help flag
- Once you know what the command is and what flags you need to set you can run it.
- The output that is sent to Stdout or Stderr will be printed in the terminal

```
markgrivainis — grivam01@cn-0024: ~/projects/dna_identification/dna_id — .....
[ └$                                                                          ]
 ┌grivam01@cn-0024 ~/projects/dna_identification/dna_id ‹master*›
[ └$ module load samtools/                                                    ]
 ┌grivam01@cn-0024 ~/projects/dna_identification/dna_id ‹master*›
[ └$ samtools flagstat ../data/tmp/fastq/0101_50_percent_and_0102_50_percent.fast]
q.gz.bwa.sam.bam
183855 + 0 in total (QC-passed reads + QC-failed reads)
0 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
92800 + 0 mapped (50.47% : N/A)
0 + 0 paired in sequencing
0 + 0 read1
0 + 0 read2
0 + 0 properly paired (N/A : N/A)
0 + 0 with itself and mate mapped
0 + 0 singletons (N/A : N/A)
0 + 0 with mate mapped to a different chr
0 + 0 with mate mapped to a different chr (mapQ≥5)
 ┌grivam01@cn-0024 ~/projects/dna_identification/dna_id ‹master*›
 └$
```


NYU Langone Health

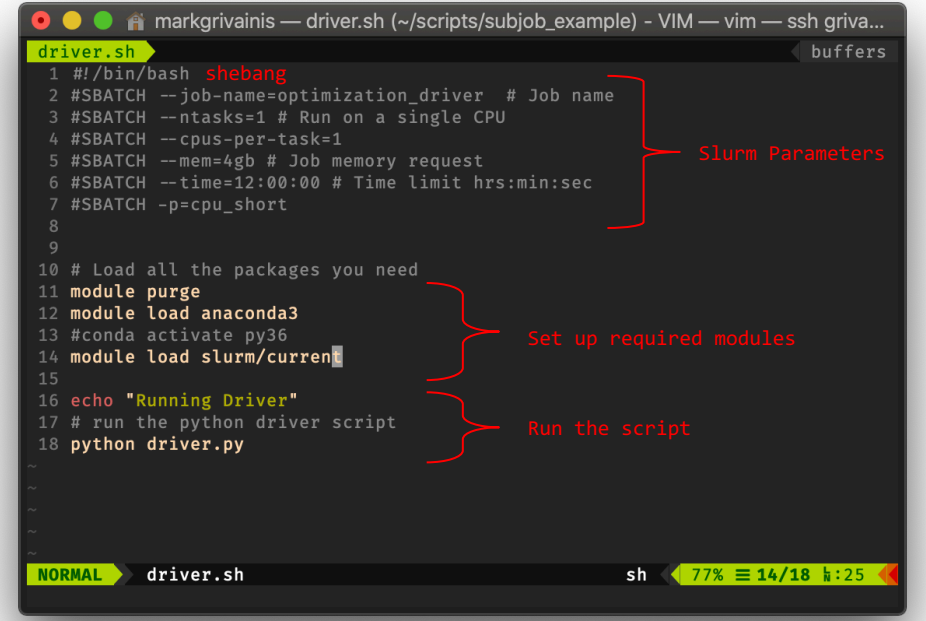
Installing Python Packages

- You can either use conda (anaconda / miniconda) or pip
- I prefer using pip as I have encountered less issues
- My go-to versions of python is:
  - module load python/cpu/3.7.2
  - module load python/gcc/3.6.5
- Pip should be installed with this version of Python, it is not with all the versions on BP
- If it is not installed, this command will install it after loading the python module
  - `python -m pip install --user pip`
- Pip will normally try to install packages into a folder that you do not have access to on BP
- To avoid any issues, install packages into your home directory using the --user flag
  - `pip install --user cutadapt`
- Remember to load the same version of python that was used to install the packages every time.

NYU Langone Health

Shell Scripts

- Shell scripts can range from a few lines to thousands of lines of code
- For this course, shell scripts will act as wrapper scripts when running sbatch.
  - A wrapper script will configure the environment for the batch job run by Slurm
  - The main components in the script are:
    - Shebang
    - Setting up the Slurm parameters
    - Purging and loading the correct modules
    - Loading a virtual environment / conda environment
    - Setting up any folders for output.
    - Running the program required for the analysis



NYU Langone Health

My Workflow

1. Start with an interactive job
    1. Load the modules you require (Keep track of the modules you use for writing .sh script)
    2. Use the command line application to run your analysis
        1. If your dataset is large, try to make a small version of it for testing
            1. For example, if you are working with FASTQ files, just include the first 100 reads from both files
        2. Read the documentation, focus on what flags are available to you, and what flags you require
    3. Once the script is running build your shell script using the commands you ran in the interactive session
2. Run the shell script using sbatch and the subset of the data, does it match the ijob
3. Run the same shell script using the entire dataset

NYU Langone Health

Adding Command Line Arguments to Shell

- Positional command line arguments are represented as variables in a shell script
- Variables start with a '$' symbol
  - $0 is the name of the script
  - Any subsequent numbers will be the argument in those positions
- Use these variables to pass on command line arguments to the wrapped script
  - In qc.sh

    ...

    ```
    fastqc <flags> $1 $2
    ```
  - `sbatch qc.sh test.R1 test.R2`

```
# shell script 'cmd_args.sh'
echo $1
echo $2
# ---- end of script ----
```

```
$ sh cmd_args.sh argument1 argument2
 argument1
 argument2
```


NYU Langone Health

Some Useful Commands

- `head <file>`
  - Output the first 10 lines of the file
  - "-n 100" flag will output the first 100 lines
- Piping
  - `<cmd> | <cmd>`
    - will feed the output of one command to the next
    - cat <filename> | head
  - `cmd > file`
    - output the result of the command to specified file file
  - `cmd >> file`
    - Append the result of the command to the file
- `grep "pattern" file`
  - Search for the specified pattern in the file
- `cat <file>, less <file>, more <file>`
  - Will all output the file to the terminal, all have different features.
  - `cat` can also be used to join files together

NYU Langone Health