

# Introduction to Working in the Linux shell Environment

February 9, 2012

**Note: Do not attempt to copy and paste out of this document. Commands will likely not work!**

**Some software you will need** (Note: these are examples of free software, not endorsements):

<u>In order to</u>	<u>From Mac or Linux</u>	<u>From Windows</u>
Connect to submit (ssh)	Terminal	PuTTY
Move files to/from submit	Cyberduck	FileZilla
Edit text files	TextWrangler	Komodo Edit

## Getting around in Linux:

- File paths (directories or folders): /, /home/magitz/, /scratch/hpc/magitz/
- pwd, cd, ls (Where am I, change directory, list directory)
- cp, mv, rm (copy, move, delete)
- more, less, head, tail, cat (examine files)
- nano, vim (text editors in Linux)

## Making things easier:

- Tab completion
- history: redo something that you did before without retyping
- man: getting help, also -h or --help flag

## Learning by doing:

1. Connect to HPC Center: `ssh user@submit.hpc.ufl.edu`
  - a. Type your password and hit return (no characters display while you type).
2. Where are you when you login? `pwd`
3. What files are there? `ls`
4. Make a directory: `mkdir test_script`
5. Now what's there? `ls -l`
  - a. Linux commands usually have flags to change how they work
  - b. `man`, `-h` or `--help` often give you help
6. Change into test\_script directory: `cd test_script` or `cd tes<tab>`
7. Copy a sample file here (.): `cp /scratch/hpc/bio/training/09Feb12/simple.pbs .`
8. Check that the copy worked: `ls`
9. Delete that file: `rm simple.pbs`
  - a. That file is now GONE! Not in your recycle bin or trashcan, but gone!
10. Luckily we were working with a copy, let's copy it again: up-arrow, up-arrow, return
11. Look at the file: `more simple.pbs`
12. For now, ignore lines that start with a # or #PBS, what does this script do?

- a. Can we run this script here? Why not?
  - b. Connect to a test node: `ssh test[01,04, or 05]`
  - c. `cd test_script`
  - d. Now we can run it: `./simple.pbs`
  - e. Did it do what you expected?
  - f. Logout of test node: `logout`
13. Use the scheduler to run this job:
- a. Can't run from home, so...
  - b. `cd /scratch/hpc/$USER`
  - c. `qsub ~/test_script/simple.pbs` (~ means your home)
  - d. The scheduler pays attention to the #PBS lines to schedule and manage your job
14. How do the results differ?
- a. **Note** this script, and any that you write, should have: `cd $PBS_O_WORKDIR`
15. Time to get some data!
- a. Data used in analyses needs to be in `/scratch/hpc/$USER` **NOT** `/home/`
16. Go to your folder in `/scratch/hpc/` and make a directory:
- a. `cd /scratch/hpc/$USER`
  - b. `mkdir test_data`
17. Get some data: `cp ../bio/training/09Feb12/test_reads.fa test_data/`
18. `cd test_data`
19. Use `more`, `head` and `tail` to look at this file
20. Copy the example Velvet script here:
- a. `cp /scratch/hpc/bio/training/09Feb12/velvet.pbs .`
21. Edit this script to have your e-mail: `nano velvet.pbs`
22. Submit the Velvet script: `qsub velvet.pbs`
23. Check your e-mail
24. Change to assembly directory: `cd assembly`
25. Have a look at the contigs.fa
26. How many contigs? `cat contigs.fa | grep -c ^\>`
27. Now, let's run this on the test BioCluster
- a. Open a new window and `ssh user@bio.hpc.ufl.edu`
28. The BioCluster uses a different scheduler, GridEngine or GE.
- a. GE uses `#$` instead of `#PBS` for its directives, and some of the directives are a bit different. **We will cover these in more detail on Feb 23<sup>rd</sup>.**
29. Grab the GE version of the Velvet submission script:
- a. `cd /scratch/hpc/$USER/test_data/`
  - b. `cp /scratch/hpc/bio/training/09Feb12/velvet.ge .`
30. Compare the two scripts with `more`
31. Edit the GE script to have your e-mail: `nano velvet.ge`
32. Submit the GE script on the BioCluster: `qsub velvet.ge`