Introduction to bash & SLURM

C. Ryan Campbell

Duke University

c.ryan.campbell@duke.edu

14 Sept 2017

Overview

Today's Goals

- Log into the cluster

Today's Goals

- Log into the cluster
- Learn some basic commands

Today's Goals

- Log into the cluster
- Learn some basic commands
    - cd, mkdir, ls, scp

Today's Goals

- Log into the cluster
- Learn some basic commands
  - cd, mkdir, ls, scp
  - sort, uniq, cut, nano

Today's Goals

- Log into the cluster
- Learn some basic commands
    - cd, mkdir, ls, scp
    - sort, uniq, cut, nano
- Learn some advanced commands

Today's Goals

- Log into the cluster
- Learn some basic commands
    - cd, mkdir, ls, scp
    - sort, uniq, cut, nano
- Learn some advanced commands
    - grep, sed

Today's Goals

- Log into the cluster
- Learn some basic commands
  - cd, mkdir, ls, scp
  - sort, uniq, cut, nano
- Learn some advanced commands
  - grep, sed
- Submit a cluster job

Today's Goals

- Log into the cluster
- Learn some basic commands
  - cd, mkdir, ls, scp
  - sort, uniq, cut, nano
- Learn some advanced commands
  - grep, sed
- Submit a cluster job
- See how far we get

Logging On & Aliases

- Use command “ssh”
- Logs on to one of two compute nodes:

```
ssh <netid>@dscr-slogin-01.oit.duke.edu
ssh <netid>@dscr-slogin-02.oit.duke.edu
```

- Your terminal window is now on the computer “dscr-slogin-01.oit.duke.edu”
- Note the command prompt

Logging On & Aliases

- To log in from off campus use a VPN
- vpn.duke.edu

Logging On & Aliases

- To log in from off campus use a VPN
- vpn.duke.edu

- Once you install the software you can access the slogin computer from off campus

Logging On & Aliases

- Aliases are used to save common commands
- On my computer the ssh login is saved as “DSCR”

```
alias DSCR="ssh cc216@dscr-slogin-01.oit.duke.edu"
```

- To set up an alias on your local machine edit your bash profile

```
nano ˜/.bash_profile
```

nano

- nano is a text-based text editor
- Just running "`nano`" opens an unsaved text document

- (O for "write Out")
- ctrl+X exits
- (X for "eXit")

nano

- nano is a text-based text editor
- Just running "`nano`" opens an unsaved text document
- Adding a filename "`nano <file>`" opens that file

- (O for "write Out")
- ctrl+X exits
- (X for "eXit")

nano

- nano is a text-based text editor
- Just running “`nano`” opens an unsaved text document
- Adding a filename “`nano <file>`” opens that file

- ctrl+O saves the file
- (O for “write Out”)
- ctrl+X exits
- (X for “eXit”)

Logging On & Aliases

- Now that you know more about nano
- On you local computer run:

```
nano ~/.bash_profile
```

- Then copy and paste this into the file

```
alias DSCR="ssh <netid>@dscr-slogin-01.oit.duke.edu"
```

- And save the file (ctrl+O)
- Once you close and re-open the terminal the alias will work

Getting Around

- You're on the cluster, now what?
- Where are you?

```
pwd
```

`pwd` = print working directory

- Change into our course directory
- Absolute vs. Relative location

```
cd /work/cc216/490S/
```

`cd` = change directory

- Is this a relative or absolute address?
- How can you tell?

Looking Around

- You're in our directory, now what?
- What's in here?

```
ls
```

`ls` = list

- Also works with an absolute address

```
ls /work/cc216/490S/
```

`ls -l` = list with long format

- -l is a "flag"

Flags and Help

- -l is a “flag”
- General grammar rules:

  `<command> <flags> <input>`
- Spaces and order matter

```
ls -l /work/cc216/490S
```

- If you’re unsure what the flags are “-h” or “–help” often works

```
ls -h
ls --help
```

Changing Files

- How do you add a folder?

```
mkdir <name of directory>
mkdir /work/cc216/490S/test_directory
```

  `mkdir` = make directory

- Try making a directory:

```
mkdir <your netid>
```

  `ls` list to confirm it worked

Changing Files

- There are a couple of options for file and folder manipulation
- To create a file:

```
touch <name of file>
```

- To copy a file:

```
cp <name of file> <new name of file>
```

- To move a file:

```
mv <current name of file> <new name of file>
```

- Try these inside of your recently created directory

Running Jobs on the Cluster

- The main reason to use the cluster is for additional computing ability
- This means either longer runs or higher computing power
- To take advantage of this you have to submit a “job”
- The cluster will then place your job in a queue and run it when the time comes
- The language of these jobs is bash

Running Jobs on the Cluster

```
#!/bin/bash
#
#SBATCH --job-name=L2_L001_bwa
#SBATCH --output=/work/cc216/microcebus_gatk/align/L2_L001.align.out
#SBATCH --error=/work/cc216/microcebus_gatk/align/L2_L001.align.err
#
#SBATCH -p yoderlab
#SBATCH --mem=16G
#SBATCH --nodes=1


cd /work/cc216/microcebus_gatk/align/
bwa mem -t 4 -M mmur3
/work/cc216/microcebus_gatk/input_fastq/L2/L2_L001.r1.trimm.5.20.fastq.gz
/work/cc216/microcebus_gatk/input_fastq/L2/L2_L001.r2.trimm.5.20.fastq.gz >
L2_L001.sam
samtools view -bt ../chr_group_contigs/mouse_lemur_sex_chr.fa L2_L001.sam >
L2_L001.bam
```

Running Jobs on the Cluster

- A simple one to try out:

```
#!/bin/bash
#
#SBATCH --job-name=test_script
#SBATCH --output=/work/cc216/490S/<your netid>/test.out
#SBATCH --error=/work/cc216/490S/<your netid>/test.err
#
#SBATCH --mem=2G
#SBATCH --nodes=1

cd /work/cc216/490S/<your netid>/
date
sleep 60
date
touch script_is_done.finish
```

Making a Cluster Script

- What SHOULDN'T you use?

Making a Cluster Script

- What SHOULDN'T you use?
- What SHOULD you use?

Making a Cluster Script

- What SHOULDN'T you use?
- What SHOULD you use?
    - nano on the cluster

Making a Cluster Script

- What SHOULDN'T you use?
- What SHOULD you use?
  - nano on the cluster
  - TextWrangler/Sublime/etc locally

Making a Cluster Script

- What SHOULDN'T you use?
- What SHOULD you use?
  - nano on the cluster
  - TextWrangler/Sublime/etc locally
- but how do you get it to the cluster?

Secure Copy

- Used to copy a file down from the cluster or up to the cluster

```
scp <name of file to copy> <file location desired>
scp <name of file to copy> <location and name if renaming>
```

- To copy a file from a different computer, the location has to contain the computer name:

```
scp -v
cc216@dcc-slogin-02.oit.duke.edu:/dscrhome/cc216/490S/hsap_hypoxia_gene.diff
 /Documents/git_repos/duke-bio490s/labs/20170831_lab_intro/
```

- use “slogin-02” for file transfers
- Also useful - cyberduck

Making a Cluster Script

- Now, make the file on your computer or on the cluster
- Then use the `cat` command to check the contents

  `cat` = concatenate, writes the contents of a file to the screen

Running a Cluster Script

- Once you have a script ready to run, (doublechecked with `cat`) we'll use `sbatch` to submit the script

```
sbatch <name of script>
```

- Then to track the progress, use `squeue`:

```
squeue -u <your netid>
```

- This returns the jobs you have in the queue
- What do you think happens if you run just `squeue`?

You Ran a Cluster Script!

- Ta-Da!! Now you can run a job on the cluster!

You Ran a Cluster Script!

- Ta-Da!! Now you can run a job on the cluster!
- You now have the minimum knowledge to get onto the cluster and use the resource to run software that is unsuitable for your laptop

You Ran a Cluster Script!

- Ta-Da!! Now you can run a job on the cluster!
- You now have the minimum knowledge to get onto the cluster and use the resource to run software that is unsuitable for your laptop
- Now, there are a lot more things that the cluster and bash can do, mostly involving data manipulation

You Ran a Cluster Script!

- Ta-Da!! Now you can run a job on the cluster!
- You now have the minimum knowledge to get onto the cluster and use the resource to run software that is unsuitable for your laptop
- Now, there are a lot more things that the cluster and bash can do, mostly involving data manipulation
- Let's delve into some of those!

Tools

- Input and output from commandline
    - Using > and |
- `sort` - sorting
- `uniq` - sort and eliminate duplicates
- `cut` - split a file into columns

Input and Output

- Unless otherwise noted:
  - Input comes after the command
  - Output “prints to screen”
- Using > and | changes that
- > takes output and (over)writes to a file

```
ls > list.txt
```

- If you use double >> it appends to a file

```
ls -lt >> list.txt
```

Input and Output

- Using | changes input
- | takes output and passes it to the next command

```
ls | head -n1
```

- You can string together many |'s to perform complicated actions

sort

- `sort` takes input and sorts it! (simple, right?)

```
sort <name of file>
cat <name of file> | sort
```

- Common flags:
    - -n sorts numerically
    - -u sorts and only presents unique hits
    - -r sorts reverse

uniq

- `uniq` takes input and removes adjacent duplicates! (still simple, right?)

```
uniq <name of file>
cat <name of file> | uniq
```

- Common flags:
  - -c counts each unique line
  - -d reverses the meaning (prints only duplicates)

cut

- `cut` takes input and divides it into “columns”

```
cut -d<what to divide by> -f<which columns
you want> <name of file>
```

- Common flags:
  - -d what to divide by (“,” “ ” “tab”)
  - -c take n characters (-c1-10 takes first 10 characters in each line)
  - -f which columns you want:
    - -f1, first only
    - -f1-5 one through five
    - -f1,5 first and fifth only

Advanced Tools

- The following tools are more advanced and complicated
- You will often see them in online forums
- You don't have to be a wizard
- It is good to be familiar with them

  (Once you are, there isn't a dataset you won't be able to manage)

- `grep` - regular expression search
- `sed` - search and replace patterns
- `awk` - counting as well as search

The End