Introduction to Server and Workstation Use in the Barker & Dlugosch Labs
KM Dlugosch – March 2017

1) Server organization

For most tasks you can use the Dlugosch lab workstation **RADthistle**

- **128.196.193.184** - general purpose workstation with 8 cores and 64GB RAM

As needed, you can also use the Barker lab servers. These include:

- **ratel.arl.arizona.edu** - high memory cycle server with 48 total cores and 512 GB RAM for next-gen assembly and other high memory data processing
- **hyrax.arl.arizona.edu** – cycle server with 64 total cores and 128 GB RAM for general purpose computing (ie, non-memory intensive)
- **packrat.arl.arizona.edu** - lab file server for long-term data storage with 16TB of RAID 10 storage. Your data and any files not in active use should be stored here.

Use SSH to log onto a machine with the username that Mike and/or Katrina gave you:
for example RADthistle (you must be on campus or on the campus VPN)

```
~$ ssh kdlugosch@128.196.193.184
```

for example Ratel:

```
~$ ssh kdlugosch@ratel.arl.arizona.edu
```

you can use sftp from one machine to connect to another to 'put' and 'get' files

2) Barker lab server use etiquette

Note that these are lab-wide resources and we use informal load-balancing to keep everything running smoothly. This is achieved either by asking around the lab to see if anybody will be using lots of cores soon or using the load-balancing script (ratel_cpu_loader.pl) that will automatically check the current load against your requested load. As a group we try to manage load so that ratel stays under 48 and hyrax under 64 cores used.

To check the current activity, type 'top' to see the current processor use and memory load. Type 'q' to get out of top. You can also type 'free -m' to get a short summary of the memory load. You want to see plenty of available memory and essentially no swap activity. There are 48 cores on Ratel, so 480% use of processors shows that they are fully maxed out. Don't fully max them out.

If the machine looks like it has some memory available for your work, start your run(s), and then check back on 'top' periodically to make sure that you haven't overloaded the processors or the RAM on the machine.

3) Running things on the server

When you log on, you are obviously accessing the server (e.g. Ratel) through a shell terminal on your own computer. When you want to start a run, you'll want to create another shell terminal on Ratel itself, that you can leave it running when you have set things up and want to disconnect from Ratel. This is accomplished though a type of software creatively called 'Screen'. Screen's latest incarnation is 'Byobu' (Japanese for 'screen').

To launch a Byobu terminal on Ratel, simply type 'byobu' at the command line in your shell session. You will see a new screen pop up with various info. More info about byobu is here: http://manpages.ubuntu.com/manpages/precise/en/man1/byobu.1.html (note especially the 'Keybindings' section, and https://help.ubuntu.com/10.04/serverguide/C/byobu.html). The most important commands are F6 to detach from your byobu screen – this leaves it running while you go back to your regular shell – and 'screen -r' or 'byobu' from the regular shell, which resumes the byobu screen that you left running in the background. Practice launching, leaving, and returning to byobu before you use it for your real analyses.

Something that you will often do is launch a process in byobu, detach it, and then check 'top' to see the resources that you are consuming.

4) Managing your data files

Next-gen data files are HUGE! You MUST manage them responsibly.

- Keep the compressed (e.g. *.gz file) version of your original data on either RADthistle or packrat ONLY. Don't keep multiple copies around.
- To work with the data, move the compressed version if you need to move the file (e.g. if you need to work with it on Ratel), open it and delete the copy (not original) of the compressed version immediately.
- As soon as you process your raw data file in any way (e.g. cleaning it), delete the uncompressed raw data file immediately.
- Do this with every file, at every step. Incorporate immediate deletion of files that you don't need into your pipelines. For example, if your Illumina data came back as 40 compressed files, create a pipeline to work through them, deleting each uncompressed raw file after you clean it. Several such pipelines are available in the lab.