Analyzing ChIP-Seq Data at the Command Line

Quick UNIX Introduction:
UNIX is an operating system like OSX or Windows. The interface between you and the UNIX OS is called “the shell”. There are a few flavors of shell but the MSI standard is bash. The shell is what you see, your environment, when you open PuTTY or Terminal.

UNIX acts like any other operating system and allows you to store and create files then use them during processes (e.g. running a program). Unlike OSX or Windows UNIX is not visual and has to be navigated using simple commands through the shell.

A Note about remote computing:
The goal of this tutorial is to introduce you to using MSI computational resources. This means you are going to use your computer to talk to other computers and tell those computers what to do. This also means that you have to make sure you know how to navigate a different computational environment (UNIX) and the data you need is somewhere MSI systems can access it (i.e. not on your personal computer).

Brief Outline of MSI:

Login Node: When you access MSI systems the login node is the first system that you connect to. The login node is only there to direct you to other MSI systems.
Lab System: A group of computers that can be accessed using the isub command. This is an older system with less computational power behind it. This group of computers is transitioning towards an interactive only system and soon you will not be able to submit batch jobs to Lab.

Itasca System: This is one of the faster, fancier computer systems. It is designed with multiple node computational jobs in mind.

**Mesabi System:** This is the system where you should be doing all of your work. Mesabi is 7 times larger than Itasca and has many queues designed for the type of computational work often done by biologists.

SUs: Service Units (SUs) are MSI method of tracking usage on Itasca and Mesabi. Starting with the 2016 yearly allocation each group will automatically be granted 70,000 SUs and access to Mesabi. A vast majority of groups will not need more than 70,000 SUs but MSI is happy to grant your group more SUs

PBS queuing: PBS is a queuing program that takes care of reserving and ordering jobs to be run on the different systems. PBS takes special commands that allow you to ask for a specific amount of time and computational power. Once you submit your job (i.e. program) it will get in line (in the queue) and will run once room opens up.

Storage: Each group is allotted some amount of storage the default amount is 100GB but your group can request as much storage as you can justify.

Software that you downloaded:


This editor will allow you to write and edit scripts and save them to your MSI space. By connecting Komodo Edit to MSI you can write, saved and edit documents and scripts that are saved in MSI space. Komodo Edit is an alternative to learning a true editor such as VI, Nano or Emacs. You should still learn how to use an editor because you might not always have Komodo and the editors can do much fancier tricks.

Preferences->Servers-> + button
Server type: SFTP
Name: MSI
Hostname: login.msi.umn.edu
Username: <your MSI username>
Password: <your MSI password>

**FileZilla Client**: [https://filezilla-project.org/](https://filezilla-project.org/)
Setup: [https://www.msi.umn.edu/support/faq/how-do-i-use-filezilla-transfer-](https://www.msi.umn.edu/support/faq/how-do-i-use-filezilla-transfer-)
This SFTP/FTP client will help you move files into and out of your MSI space.


This editor will allow you to write and edit scripts and save them to your MSI space. We will set this up in class.

**Windows Users Only:**

**PuTTY:** [http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html](http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html)

**Setup:** [https://www.msi.umn.edu/content/connecting-hpc-resources](https://www.msi.umn.edu/content/connecting-hpc-resources)

Unlike Mac/OSX, Windows does not have a native UNIX terminal. PuTTY is a terminal emulator for Windows and will allow you to interact with MSI systems using UNIX commands.

**Goals:**
- Log into MSI and connect to the Lab system.
- Learn some basic UNIX commands.
- Learn how to load software using the module system
- Learn how to map Illumina reads to a genome using BWA
- Learn how to call peaks with MACS14
- Learn how to write a bash script to automate running software
- Learn how to convert the bash script to work with PBS to submit jobs to the queue.

**Log into MSI systems**

```
Last login: Wed Sep 17 16:35:05 on ttys001
ljmills-MacBookAir:~ ljmills$ ssh ljmills@login.msi.umn.edu
ljmills@login.msi.umn.edu's password: ************
```
Once you login you will see the welcome screen that contains some nice information.

As the @loginXX after your username in the terminal indicates you are now connected to the login node. While you can move around the filesystem and use UNIX commands while connected to the login node you won't be able to use any of the software installed on MSI systems. In order to gain access to the installed software you will need to connect to one of the HPC or interactive HPC systems. We are going to connect to the HPC system Mesabi and then start an interactive session using the PBS command qsub -l

As the @loginXX after your username in the terminal indicates you are now connected to the login node. While you can move around the filesystem and use UNIX commands while connected to the login node you won't be able to use any of the software installed on MSI systems. In order to gain access to the installed software you will need to connect to one of the HPC or interactive HPC systems. We are going to connect to the HPC system Mesabi and then start an interactive session using the PBS command qsub -l

```bash
ljmills@login03 [~] % ssh mesabi
Password:
ljmills@ln0003 [~] % qsub -I -l nodes=1:ppn=4,mem=8gb,walltime=4:00:00
qsub: waiting for job 469118.mesabim3.msi.umn.edu to start
qsub: job 469118.mesabim3.msi.umn.edu ready
```

qsub
- -l flag creates an interactive session
- -l nodes=1:ppn=4 requests a single nodes and 4 processors
• mem=8gb requests 8gb of memor
• walltime=4:00:00 requests use of the above resources for 4 hours
• You can change each of these requests up to the maximum allowed on the system as dictated by the different queues. See: https://www.msi.umn.edu/queues

Once your session has stared, lets see where we are in the file system. When you log into the system for the first time you are automatically taken to your home directory. Your home directory will always have the format of /home/yourGroup/yourMSIaccount.

```
ljmills@cn0538 [~] % pwd
/home/msistaff/ljmills
ljmills@cn0538 [~] %
```

Lets take a look at what is in your home directory, do you have any files there?

```
ljmills@cn0538[~] % ls
```

What about in your group directory?

```
ljmills@cn0538 [~] % cd ..
ljmills@cn0538 [/home/msistaff] % ls
```

Lets go to a directory that will have some files in it.

```
ljmills@cn0538 [~/chipTutorial] % cd /home/msistaff/public/
ljmills@cn0538 [/home/msistaff/public] % ls
basicChIP chenzler garbe hg19Canonical_PARmaskedOnY.fa qcIllumina
test_trimmomatic.sh ucsc.hg19.fasta ucsc.hg19.fasta.fai
```

Lets get a different view of the directory contents.

```
ljmills@cn0538 [/home/msistaff/public] % ls -lh
total 6.7G
drwxr-s---. 2 ljmills msistaff 4.0K Oct 23 10:46 basicChIP
drwxrwsrwx. 2 chenzler msistaff 4.0K Sep 23 08:38 chenzler
drwxr-xr-x. 4 jgarbe msistaff 4.0K Jun 13 08:43 garbe
drwxrwsrwx. 2 ljmills msistaff 4.0K Sep 22 11:13 qcIllumina
-rwxr-x---. 1 ljmills msistaff 789 Sep 18 10:14 test_trimmomatic.sh
-rw-rw-r--. 1 jgarbe msistaff 3.0G Dec 11 2013 ucsc.hg19.fasta
-rw-------. 1 chenzler msistaff 3.5K Dec 11 2013 ucsc.hg19.fasta.fai
```
The data we will need for the tutorial is in the basicChIP directory. Move into that
directory list what is in it and then view the G1E_CTCF.fastq FASTQ file there
using the less UNIX command. What does this file look like?

```
ljmills@cn0538 [/home/msistaff/public] % cd basicChIP/
ljmills@cn0538 [/home/msistaff/public/basicChIP] % ls -lh
total 128K
drwxrwsrwx 2 ljmills msistaff 4.0K Oct 23  2014 annotations
drwxrwsrwx 2 ljmills msistaff 4.0K Dec  3 08:38 bams
drwxrwsrwx 2 ljmills msistaff 4.0K Dec  3 08:43 fastq
drwxrwsrwx 2 ljmills msistaff 4.0K Dec  3 08:44 scripts
ljmills@cn0538 [/home/msistaff/public/basicChIP] % less
fastq/G1E_CTCF.fastq
```

Move back to your home directory. There are three ways to do this, directly type
in your home directoy after cd, just use cd or use the ~ which represents your
home direcotry.

```
ljmills@cn0538 [ ] % cd /home/msistaff/ljmills/
ljmills@cn0538 [~] % cd
ljmills@cn0538 [~] % cd ~
```

Create a directory for the G1E_CTCF and G1E_input FASTQ files named
tutorial, move into that directory, then copy the fastq files into this folder. Don’t
remember the name of the files, ls to take a look again. Tab completion will also
help you here. When typing the names of folder or files pressing Tab will
complete the name for you.

```
ljmills@cn0538 [~] % mkdir tutorial
ljmills@cn0538 [~] % cd tutorial
ljmills@cn0538 [~/tutorial] % cp /home/msistaff/public/basicChIP/fastq/G1E_CTCF.fastq ~/tutorial
ljmills@cn0538 [~/tutorial] % cp /home/msistaff/public/basicChIP/fastq/G1E_input.fastq ~/tutorial
ljmills@cn0538 [~/tutorial] % ls
G1E_CTCF.fastq  G1E_input.fastq
```

The first step when working with ChIP-Seq data is to map the back to the organism’s
genome. In this case we have sequence from DNA isolated from mouse G1E cell line.
We will use a program call BWA to map these reads back to the mouse (mm9)
genome.

BWA requires a specially made index files for any genome that you want to map reads to. These index files are part of the magic that allows BWA to map reads so quickly. BWA indexes can be found for many genomes here: /panfs/roc/rissdb/genomes

```
ljmills@cn0538 [~/tutorial] % module avail bwa
ljmills@cn0538 [~/tutorial] % module load bwa/0.7.4
ljmills@cn0538 [~/tutorial] % bwa mem /panfs/roc/rissdb/genomes/Mus_musculus/mm9_canonical/bwa/mm9_canonical.fa G1E_CTCF.fastq > G1E_CTCF.sam
```

SAM (sequence alignment map) is the human readable alignment output. More about SAM format can be found here: http://samtools.github.io/hts-specs/SAMv1.pdf

Let's look at the SAM file.

```
ljmills@cn0538 [~/tutorial] % less G1E_CTCF.sam
```

What does this file contain? Can you find the alignment information?

While SAM files are great the peak finder we are using MACS14 does not take SAM formatted files. Instead we will convert these SAM files to BAM files (binary alignment map). BAM files are not human readable but contain the same information as the SAM files

```
ljmills@labq01 [] % samtools view -S -b G1E_CTCF.sam > G1E_CTCF.bam
```

While running jobs straight from the command line is useful there are some disadvantages:

1) You have to type the commands perfectly.
2) You don't have a record of what you did.
3) It is not easy to run lots of commands in a row or to run the same command again.
4) You have to wait around for the software to finish before you can do something else.
5) YOU DON'T HAVE A RECORD OF WHAT YOU DID!!!
Don’t worry there is an easy (ish) way to overcome all of these issues… Submitting jobs via and PBS script!

Copy bwa_mem_aln.sh from /home/msistaff/public/basicChIP into your tutorial directory. The open this file in Komodo Edit, File -> Open -> Remote File. You will need to select MSI from the Server drop down menu then navigate to the tutorial directory.

Bwa_mem_aln.sh

```bash
#!/bin/bash

#SBATCH -n nodes=1:ppn=4,mem=15GB,walltime=4:00:00
#SBATCH -m ae
#SBATCH -j oe
#SBATCH -N BWA_mem

cd /home/msistaff/ljmills/chipTutorial
module load bwa
module load samtools

bwa mem -t $PBS_NP /panfs/roc/rissdb/geneomes/Mus_musculus/mm9_canonical/bwa/mm9_canonical.fa G1E_CTCF.fastq > G1E_CTCF.sam
samtools view -S -b G1E_CTCF.sam > G1E_CTCF.bam
```

This PBS script is written in a scripting language called bash. The very first line is called the Sha-Bang and tells the system you are on how to interpret the following commands. Bash is a scripting language that has the ability to do lots and lots of things. My favorite Bash scripting guide can be found here: http://www.tldp.org/LDP/abs/html/

Lines that begin with `#PBS` are commands that will be interpreted by the PBS queuing program. Like `isub` the PBS commands reserve a specific amount of computational resources to be used to complete the items in your script.

The rest of the script are the same commands that you typed into the terminal to run FastQC and Trimmomatic. What elements of this script do you need to change so that it will work for you? Hint: `cd` to your directory.

Let’s submit this PBS script to Mesabi using the `qsub` command.

```
ljmills@cn0538 [~/tutorial] % qsub bwa_mem_aln.sh
672036.mesabim3.msi.umn.edu
```
The number that pops up when you submit a job is the jobid and is confirmation that your job was submitted to the queue.

You can check the status of your job using `qstat`, you will also get an email when your job finishes or if it is cancelled because of an error (aborts). Using the `-u` flag will let you see only your jobs.

```
ljmills@cn0538 [~/tutorial] % qstat -u ljmills
```

You should have two jobs running, one will be the `isub` job that you started to connect to the Lab system, the other will be the job you just submitted. The `S` column is the status of your job: `R` is running, `Q` is queued and `C` is cancelled. When a job finishes it will go through the `C` state even if there wasn’t an error.

What is in your tutorial directory once the job finishes? What do the error and output files contain?

While `bwa_mem_aln.sh` is a nice record of what you did it is not very flexible and can only be used on these specific files. Let’s look at another PBS bash script that is a bit more flexible. Copy `bwa_mem_aln_v.sh` and `bwa_mem_aln_for.sh` into your tutorial directory.

```
#!/bin/bash

#PBS -l nodes=1:ppn=4,mem=15GB,walltime=4:00:00
#PBS -m ae
#PBS -j oe
#PBS -N BWA_mem

cd /home/msistaff/ljmills/tutorial

module load bwa
module load samtools

INDEX=/panfs/roc/risddb/genes/Mus_musculus/mm9_canonical/bwa/mm9_canonical.fa
FASTQ=G1E_CTCF.fastq

bwa mem -t $PBS_NP $INDEX $FASTQ > $FASTQ.sam
samtools view -S -b $FASTQ.sam > $FASTQ.bam
```
In this PBS bash script variable INDEX and FASTQ are used in the bwa mem call instead of directly inputting these files. This would allow you to re-run the same commands using different FASTQ files and indexes. Again this script is not as flexible as it could be, how could you write a script to map multiple FASTQ files?

Bwa_mem_aln_for.sh

```bash
#!/bin/bash

#PBS -l nodes=1:ppn=4,mem=15GB,walltime=4:00:00
#PBS -m ae
#PBS -j oe
#PBS -N BWA_mem

cd /home/msistaff/public/basicChIP
module load bwa
module load samtools
INDEX=/panfs/roc/rissdb/genomes/Mus_musculus/mm9_canonical/bwa/mm9_canonical.fa

for FASTQ in G1E_CTCF.fastq G1E_input.fastq
  do
    bwa mem -t $PBS_NP $INDEX $FASTQ > $FASTQ.sam
    samtools view -S -b $FASTQ.sam > $FASTQ.bam
  done
```

While this script looks very simple it has a few new concepts in it.

Variables- both FASTQ and INDEX are variables. When the variable is first set you only need to give the name of the variable (i.e. FASTQ) but when you then refer to (try to use) the variable you will need to add a $ (i.e. $FASTQ).

For Loop- let you run the same command over a list. The general structure of a for loop in bash is

```
for arg in [list]
do
  command(s)
done
```

The list can be as long as you need it to be. It can also be a UNIX command that lists the files for you so you don’t have to type them in yourself. Edit your script to match the for loop below.
Lets run `bwa_mem_aln_for.sh` so we can get BAM files for both of the FASTQ files in your tutorial directory.

```
ljmills@cn0538 [~/tutorial] % qsub bwa_mem_aln_for.sh
672037.mesabim3.msi.umn.edu
```

Great now we have all of the files we need to start to call peaks in our data. There are a couple more BAM files in `/home/msistaff/public/basicChIP` that we will also use when we call peaks. Copy `G1E_ER4_CTCF.fastq.bam` and `G1E_ER4_input.fastq.bam` into your tutorial folder using the `cp` command. Also, copy `macs14.sh` into your tutorial folder.

```
ljmills@labq01 [] % cp /home/msistaff/public/basicChIP/bams/G1E_ER4_*_.bam ~/tutorial
ljmills@labq01 [] % cp /home/msistaff/public/basicChIP/scripts/macs14.sh ~/tutorial
```

We are using a program called MACS to call peaks in our data. MACS uses the alignment files to try and identify regions of the genome where more reads align as compared to the background signal. MACS will also use the alignment from your input sample to make sure the build up of reads is because of your antibody isolation and not an artifact of genome structure.


Open `macs14.sh` in Komodo Edit.

```
#!/bin/bash -l

#PBS -l nodes=1:ppn=4,mem=15GB,walltime=4:00:00
#PBS -m ae
#PBS -j oe
#PBS -N macs

cd /home/msistaff/public/basicChIP

module load macs/1.4.1
module load R/

macs14 -t G1E_CTCF.fastq.bam -c G1E_input.fastq.bam -f BAM -g mm --bw 300 -w -S -n G1E_CTCF_macs

macs14 -t G1E_ER4_CTCF.fastq.bam -c G1E_ER4_input.fastq.bam -f BAM -g mm --bw 300 -w -S -n G1E_ER4_CTCF_macs
```
To call peaks with the most confidence MACS needs both a control (-c input sample) and treatment (-t ChIP sample) alignment file. MACS also wants to know the size of the genome (-g) and bandwidth or sonication fragment size (-bw). The combination of the –w and –S flag have MACS write a single WIG file from the data that encompasses the whole genome. The –n flag gives a name to the output.

Output files
1. NAME_peaks.xls is a tabular file that contains information about called peaks. You can open it in excel and sort/filter using excel functions. Information include: chromosome name, start position of peak, end position of peak, length of peak region, peak summit position related to the start position of peak region, number of tags in peak region, -10*log10(pvalue) for the peak region (e.g. pvalue = 1e-10, then this value should be 100), fold enrichment for this region against random Poisson distribution with local lambda, FDR in percentage. Coordinates in XLS is 1-based which is different with BED format.

2. NAME_peaks.bed is BED format file that contains the peak locations. You can load it to UCSC genome browser or Affymetrix IGB software. The 5th column in this file is the -10*log10pvalue of peak region.

3. NAME_summits.bed is in BED format, which contains the peak summits locations for every peaks. The 5th column in this file is the summit height of fragment pileup. If you want to find the motifs at the binding sites, this file is recommended.

4. NAME_negative_peaks.xls is a tabular file containing information about negative peaks. Negative peaks are called by swapping the ChIP-seq and control channel.

5. NAME_model.r is an R script which you can use to produce a PDF image about the model based on your data. Load it to R by: R --vanilla < NAME_model.r

6. Then a pdf file NAME_model.pdf will be generated in your current directory. Note, R is required to draw this figure.

7. NAME_treat/control_afterfiting.wig.gz files in NAME_MACS_wiggle directory are wiggle format files which can be imported to UCSC genome browser/GMOD/Affy IGB. The .bdg.gz files are in bedGraph format which can also be imported to UCSC genome browser or be converted into even smaller bigWig files.

Take a look at these files to see what information they contain.