Introduction to compute clustering

Lauren Wasson, Ph.D.

July 30, 2020

Already lost? Here's a reference

- https://github.com/drlaurenwasson

- Github is a software development platform
  - It is open source
  - It is active
  - It is common (teaching, development, etc)

- I currently have two main sections on github
  - R programming
  - Compute clustering

github.com/drlaurenwasson

Search or jump to... / Pull requests Issues Marketplace Explore

Your pins have been updated. Drag and drop to reorder them.

Overview | Repositories 3 | Projects | Packages

Pinned — Customize your pins

Compute_Clustering
Compute Clustering for beginners, using UNC Longleaf as an example

R_tutorials
Beginner R tutorials
R 1

Repositories

Lauren Wasson
drlaurenwasson

I'm a Postdoc at Harvard Medical School, trying to learn bioinformatics.

Edit profile

dr.lauren.wasson@gmail.com

128 contributions in the last year — Contribution settings

Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun Jul
Mon Wed Fri

Learn how we count contributions. Less More

Contribution activity

2020 2019 2018 2017

July 2020

Created 25 commits in 2 repositories
drlaurenwasson/Compute_Clustering 24 commits
drlaurenwasson/R_tutorials 1 commit

Created an issue in bcbio/bcbio-nextgen that received 2 comments — Jul 1

Installing custom genome failure with gff3

Questions about programming that I had

Compute Clustering 101- Why use this?

- The Longleaf cluster is a **Linux**-based computing system available to researchers across the campus free of charge.
- Has nearly 6500 conventional compute cores delivering 13,000 threads.
  - Therefore, you have free access to 6,500 computers, instead of your laptop.

- You can store files, write scripts, execute jobs and do data analysis on the cluster
  - Like a cloud…

UNC Longleaf- /proj space

- Each onyen (when you request access) has 30 GB storage
  - Not really enough to do any work.
  - It is backed up (I will likely keep backups of scripts, etc on here)
- Each onyen also has mass storage (2TB?)
  - This is also backed up, but it’s not designed to do work. Its for long term storage.
- Conlon lab has 5TB space on the cluster
  - Store fastq (raw sequencing reads), bed, bam files
  - Download/upload data from GEO, etc
  - Analyze data

UNC Longleaf- Getting onto the cluster

- 1. Request access (https://its.unc.edu/research-computing/request-a-cluster-account/)
  - Your preferred shell is bash
- Download UNC's vpn (only if you're home).
  - https://help.unc.edu/sp?id=kb_article_view&sysparm_article=KB0010155&sys_kb_id=87af20281b7f4c90b7de21b5ec4bcb99
- Download a gateway (Windows users only) or open a Terminal (Mac users)
  - MobaXterm (Home edition)
  - Git Bash (this is the one I'm familiar with)
  - SSH Secure Shell
  - SecureCRT
- https://its.unc.edu/research-computing/getting-logged-on/

UNC Web portal for Longleaf

- As of March 2020, theres a web portal for Longleaf
  - It seems to work pretty well, but I actually find it more complicated to submit jobs etc.

- https://ondemand.rc.unc.edu/pun/sys/dashboard

Getting onto the cluster

ssh ONYEN@longleaf.unc.edu

```
Lauren@DESKTOP-K0NAI90 MINGW64 ~
$ ssh -X lwasson@longleaf.unc.edu
|-----------------------------------------------------------------------|
|     The University of North Carolina at Chapel Hill                   |
|     ***************************************************               |
|     Unauthorized access to this system is prohibited.                 |
|                                                                       |
| This is a University system intended for University purposes          |
| only. The University reserves the right to monitor the use of         |
| this system as required to ensure its stability, availability,        |
| and security.                                                         |
|                                                                       |
| Please report any problems to "help@unc.edu", or 962-HELP,            |
| or go to https://its.unc.edu/research-computing/contact-us-directly/  |
| to submit a help request.                                             |
|-----------------------------------------------------------------------|

Password:
```

Login Node

pwd = print working directory
(Where am I on the cluster? What folder am I in?)

[Onyen@longleaf-login#]

ls -l lists all of the files in the directory you are in

Here I am in my “home” directory (~)

lwasson@longleaf-login2:~

```
[lwasson@longleaf-login2 ~]$ pwd
/nas/longleaf/home/lwasson
[lwasson@longleaf-login2 ~]$ ls -l
total 16
drwxr-xr-x 4 lwasson its_undrgrad_psx 4096 Jun 17 14:31 lwaldron_home
lrwxrwxrwx 1 root    root               20 Jun 12 18:10 ms -> /ms/home/l/w/lwasson
drwxr-xr-x 3 lwasson its_undrgrad_psx 4096 Jul 29 15:25 ondemand
-rw-r--r-- 1 lwasson its_undrgrad_psx    0 Jun 27 12:29 slurm-63117728.out
-rw-r--r-- 1 lwasson its_undrgrad_psx    0 Jun 27 12:30 slurm-63117731.out
drwxr-xr-x 2 lwasson its_undrgrad_psx 4096 Jun 27 12:30 testdir
-rw-r--r-- 1 lwasson its_undrgrad_psx    0 Jun 27 12:30 test.err
-rw-r--r-- 1 lwasson its_undrgrad_psx  142 Jun 27 12:30 test.sh
[lwasson@longleaf-login2 ~]$
```

You log into the login node (obvs). You can submit jobs on the login node, but you can’t do interactive stuff on the login node (this will make more sense later)

/proj/conlonlb

cd = change directory

It changed from my home (~) to conlonlb

lwasson@longleaf-login2:/proj/conlonlb

```
[lwasson@longleaf-login2 ~]$ cd /proj/conlonlb/
[lwasson@longleaf-login2 conlonlb]$ ls -l
total 362
drwxr-sr-x  9 lwasson rc_conlonlb_psx   731 Jul  8 20:06 bcbio
drwxr-sr-x  2 lwasson rc_conlonlb_psx     0 Jun 30 14:49 bcbiotx
drwxr-sr-x 12 lwasson rc_conlonlb_psx   517 Jun 29 15:27 cloudbiolinux
drwxrwsr-x 22 root    rc_seq-in_psx     855 Dec 11  2019 HTSF
-rwxr--r--  1  182695 rc_conlonlb_psx 21286 May 24  2013 macs2
drwxrws--- 15 root    rc_conlonlb_psx   509 Jul 27 05:09 users
[lwasson@longleaf-login2 conlonlb]$
```

/proj/conlonlb/users

```
drwxr-sr-x 15 hepperla rc_conlonlb_psx 5895 Jan  9  2018 Austin
drwxr-sr-x  6 cslagle  rc_conlonlb_psx  267 Sep  4  2014 ceslagle
drwxrwxrwx  5   227033 rc_conlonlb_psx  135 Apr 13  2016 cwilczew
drwxr-sr-x  4 kberkoff rc_conlonlb_psx  113 Jul 20 18:05 _Inline
drwxr-sr-x 15 kberkoff rc_conlonlb_psx  519 Jul 29 15:33 kberkoff
drwxr-sr-x  2 kdehghan rc_conlonlb_psx    0 Jul 22 15:30 kdehghan
drwxr-sr-x  9 lwasson  rc_conlonlb_psx 1243 Jul 29 16:13 lwasson
drwxr-sr-x  2   182695 rc_conlonlb_psx 2792 Oct 15  2012 mm10
drwxrws--T  9 fconlon  rc_conlonlb_psx  298 Feb 28  2014 nirav
drwxr-sr-x 15   176167 rc_conlonlb_psx  352 Feb 16  2015 ptandon
-rw-r--r--  1 wedward2 rc_conlonlb_psx  680 Jul  7 14:35 Tbx20eh1submission070720.csv
drwxr-sr-x  4 tvital   rc_conlonlb_psx   64 Feb 19 12:14 tvital
drwxr-sr-x  3 wedward2 rc_conlonlb_psx   24 Jun 25 14:54 Whitney
```

Basic linux commands that I use a lot

- https://github.com/drlaurenwasson/Compute_Clustering/blob/master/Handy_Unix_Tips.Md

- Ones that I used in previous slides
  - ls –l (list)
  - cd (change directory)
  - pwd (print working directory)

- Google is your friend here

What I can teach you to do on the cluster

- Load modules
- Submit jobs to do
  - RNA-seq processing and analysis
  - ChIP-seq processing and analysis
  - Processing bed files (bedtools)

Load modules

- UNC has a lot of pre-installed modules on the cluster (like packages in R, or an app on your phone)
  - To see what is available type “module spider <what you want>

lwasson@longleaf-login2:/proj/conlonlb

```
[lwasson@longleaf-login2 conlonlb]$ module spider macs

-----------------------------------------------------------------------------------------------------------------
  macs:
-----------------------------------------------------------------------------------------------------------------
     Versions:
        macs/2.1.2
        macs/2.2.7.1
        macs/2016-02-15

     Other possible modules matches:
        Core/emacs  Core/gromacs  Core/macs  emacs  gromacs

-----------------------------------------------------------------------------------------------------------------
  To find other possible module matches do:
      module -r spider '.*macs.*'

-----------------------------------------------------------------------------------------------------------------
  For detailed information about a specific "macs" module (including how to load the modules) use the module's full nam
e.
  For example:

     $ module spider macs/2016-02-15
-----------------------------------------------------------------------------------------------------------------
```

Load modules

- To load a module:

```
[lwasson@longleaf-login2 conlonlb]$ module load macs/2.2.7.1
[lwasson@longleaf-login2 conlonlb]$
```

- If you try to run a MACS2 command without loading the module first, nothing will happen, and you'll get an error
- Modules have to be loaded EVERY TIME (you can build it into your scripts). Modules get wiped every time you log off

What I can teach you to do on the cluster

- Load modules
- Submit jobs to do
  - RNA-seq processing and analysis
  - ChIP-seq processing and analysis
  - Processing bed files (bedtools)

Doing things on the cluster

- The UNC cluster (and many clusters) uses the SLURM workload manager to submit jobs.
  - https://slurm.schedmd.com/quickstart.html
- To “do things” on the cluster:
  - 1: just type it in

“Just type it in”

```
[lwasson@longleaf-login2 peaks]$ module spider bedtools
Rebuilding cache, please wait ... (not written to file) done

-----------------------------------------------------------------------------------------
  bedtools:
-----------------------------------------------------------------------------------------
     Versions:
        bedtools/2.23.0
        bedtools/2.25.0
        bedtools/2.26
        bedtools/2.29

     Other possible modules matches:
        Core/bedtools

-----------------------------------------------------------------------------------------
  To find other possible module matches do:
      module -r spider '.*bedtools.*'

-----------------------------------------------------------------------------------------
  For detailed information about a specific "bedtools" module (including how to load the modules) use the
 module's full name.
  For example:

     $ module spider bedtools/2.29
-----------------------------------------------------------------------------------------

[lwasson@longleaf-login2 peaks]$ module load bedtools/2.29
[lwasson@longleaf-login2 peaks]$
```

“Just type it in…”

```
[lwasson@longleaf-login2 peaks]$ ls -l
total 8372
drwxr-sr-x 2 lwasson rc_conlonlb_psx     339 Jul 27 16:29 1e2
drwxr-sr-x 2 lwasson rc_conlonlb_psx     156 Jul 27 15:27 1e3
drwxr-sr-x 2 lwasson rc_conlonlb_psx     405 Jul 27 15:57 5e3
-rw-r--r-- 1 lwasson rc_conlonlb_psx 4239342 Jul 27 16:07 Combined_CHD4_e10_q0.01_peaks.bed
-rw-r--r-- 1 lwasson rc_conlonlb_psx  571863 Jul 27 16:07 Combined_CHD4_e10_q0.01_peaks_final.bed
-rw-r--r-- 1 lwasson rc_conlonlb_psx  219125 Jul 28 17:39 Combined_CHD4_H3K4me3_overlap.bed
-rw-r--r-- 1 lwasson rc_conlonlb_psx 1813900 Jul 28 17:38 Combined_H3K4me3_e10.bed
[lwasson@longleaf-login2 peaks]$ bedtools intersect -a Combined_CHD4_e10_q0.01_peaks_final.bed -b Combined_H3K4me3_e10.bed
 > Combined_CHD4_H3K4me3_overlap.bed
[lwasson@longleaf-login2 peaks]$
```

Here I have typed a command “bedtools intersect” to intersect two bed files- CHD4 ChIP-seq data and H3K4me3 data
I have written the output (>) to a file called “Combined_CHD4_H3K4me3_overlap.bed”

Eventually, you will perform a job that you can’t do on the login node (memory etc is small on the login node and its bad practice to do work on the login node)

- You have two choices:
  - Get on a compute node in “interactive mode”
  - Write a script and submit to the cluster.

To get on an interactive node: submit your first job- srun

```
[lwasson@longleaf-login2 peaks]$ srun -t 5:00:00 -p interact -N 1 -n 1 --mem=20G --pty /bin/bash
srun: error: Ignoring conflicting option "x11" in plugin "x11"
srun: job 65245479 queued and waiting for resources
srun: job 65245479 has been allocated resources
[lwasson@c0801 peaks]$
```

Now we are on compute node 0801
Now you can type in all the commands you want!
- But you'll have to reload your modules.

The main difference is that **srun** is **interactive and blocking** (you get the result in your terminal and you cannot write other commands until it is finished), while **sbatch** is batch processing and non-blocking (results are written to a file and you can submit other commands right away)

Sbatch --wrap

sbatch -p general -t 0-4 --mem=16G --wrap "bedtools intersect -a Combined_CHD4_e10_q0.01_peaks_final.bed -b Combined_H3K4me3_e10.bed > Combined_CHD4_H3K4me3_overlap.bed"

```
[lwasson@c0802 peaks]$ sbatch -p general -t 0-4 --mem=16G --wrap "bedtools intersect -a Combined_CHD4_e10_q0.01_peaks_final.bed -b Combined_H3K4me3_e10.bed > Combined_CHD4_
H3K4me3_overlap.bed"
Submitted batch job 65266317
[lwasson@c0802 peaks]$ sacct
       JobID    JobName  Partition    Account  AllocCPUS      State ExitCode
------------ ---------- ---------- ---------- ---------- ---------- --------
65265947           bash   interact rc_fconlo+          1    RUNNING      0:0
65265947.ex+     extern            rc_fconlo+          1    RUNNING      0:0
65265947.0         bash            rc_fconlo+          1    RUNNING      0:0
65266317           wrap    general rc_fconlo+          1    PENDING      0:0
[lwasson@c0802 peaks]$
```

-p = partition (this will always be “general” on the unc cluster
-t = time (the amount of time you are asking for resources)
--mem = memory (the amount of memory you are asking for)

To check the status of your jobs (for the last 24 hours)
> sacct

Sbatch -- wrap

- It’s basically a way to get around writing a script. If you’re doing an analysis for the first time, I recommend using this method of submission, because you can check your work at every step.

- It’s “just type it in”, just elevated one step, as you don’t have to wait for the previous step to finish before starting a new one (this is helpful if you want to run the same command on multiple samples, for example) .

Writing a script and submitting with sbatch

- There are lots of scripts!
  - .pl = Perl scripts
  - .py = python scripts
  - .sh = shell scripts

Example of a shell script

```
#!/bin/bash

#SBATCH -p general
#SBATCH --job-name=example_script_%j
#SBATCH --mail-user=lwasson@ad.unc.edu
#SBATCH --mail-type=END
#SBATCH -c 1
#SBATCH -t 1:00:00
#SBATCH --mem=8G
#SBATCH -e example_script_%j.err
#SBATCH -o example_script_%j.out

#Load the modules that you need
module load bedtools/2.29

#Run your code

#change directory to where the files are
cd /proj/conlonlb/users/lwasson/ChIP/CHD4_e10_5/peaks

#Do the intersect
bedtools intersect -a Combined_CHD4_e10_q0.01_peaks_final.bed -b Combined_H3K4me3_e10.bed > Combined_CHD4_H3K4me3_overlap.bed


example.sh (END)
```

Submitting a batch script (sh)

```
[lwasson@c0802 lwasson]$ sbatch example.sh
Submitted batch job 65266501
[lwasson@c0802 lwasson]$ ls -l
total 261
drwxr-sr-x 5 lwasson rc_conlonlb_psx   94 Jul 30 11:41 ChIP
-rw-r--r-- 1 lwasson rc_conlonlb_psx    0 Jul 30 12:00 example_script_65266460.err
-rw-r--r-- 1 lwasson rc_conlonlb_psx    0 Jul 30 12:00 example_script_65266460.out
-rw-r--r-- 1 lwasson rc_conlonlb_psx    0 Jul 30 12:00 example_script_65266462.err
-rw-r--r-- 1 lwasson rc_conlonlb_psx    0 Jul 30 12:00 example_script_65266462.out
-rw-r--r-- 1 lwasson rc_conlonlb_psx    0 Jul 30 12:07 example_script_65266501.err
-rw-r--r-- 1 lwasson rc_conlonlb_psx    0 Jul 30 12:07 example_script_65266501.out
-rw-r--r-- 1 lwasson rc_conlonlb_psx  565 Jul 30 12:07 example.sh
-rw-r--r-- 1 lwasson rc_conlonlb_psx  328 Jul  7 15:52 generate_yaml.sh
drwxr-sr-x 7 lwasson rc_conlonlb_psx 1161 Jul 30 11:41 RNA-seq
[lwasson@c0802 lwasson]$
```

UNC Research Computing Resources

- https://its.unc.edu/research-computing/research-computing-presentations/#longleaf

My office hours

Call peaks using MACS2

- Carries MACS2 command:

macs2 -t NS50244_160302_NS500489_AHHHG5BGXX.TS-UNC-1.1.bam -c NS50244_160302_NS500489_AHHHG5BGXX.TS-UNC-4.1.bam -f BAM -g mm -B -q 0.01 --nomodel --shiftsize 100 -n ./NS50244_160302_NS500489_AHHHG5BGXX.TS-UNC-1.1_macsdone3

- Austin's MACS2 command

macs2 callpeak –t ../bowtie_out/CHD4_Rep1_clean_sync_filt.bam -c ../bowtie_out/input_Rep1_clean_sync_filt.bam -f BAMPE -g mm -n CHD4_Rep1_clean_sync_filt

- My command:

macs2 callpeak -t NS50244_160302_NS500489_AHHHG5BGXX.TS-UNC-1.1.sorted.bam -c NS50244_160302_NS500489_AHHHG5BGXX.TS-UNC-4.1.sorted.bam -f BAM -g mm -n CHD4_e10_1_q0.01 -B -q 0.01

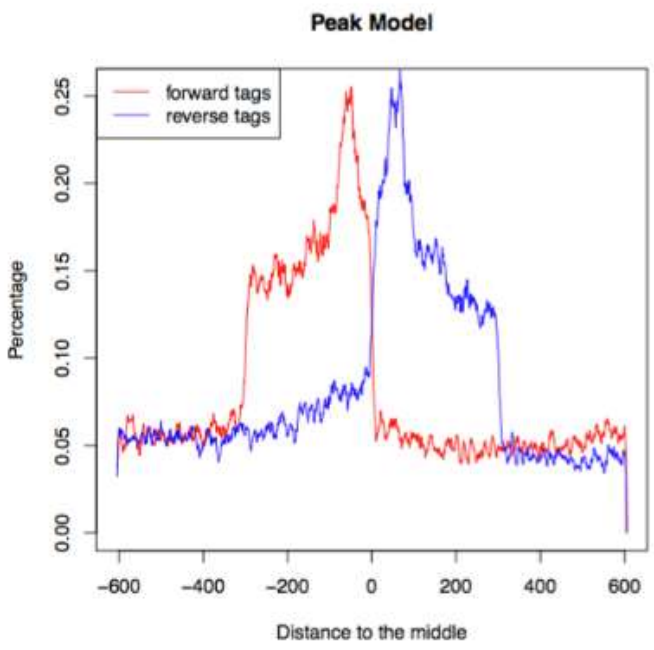
Shift size and model explanation

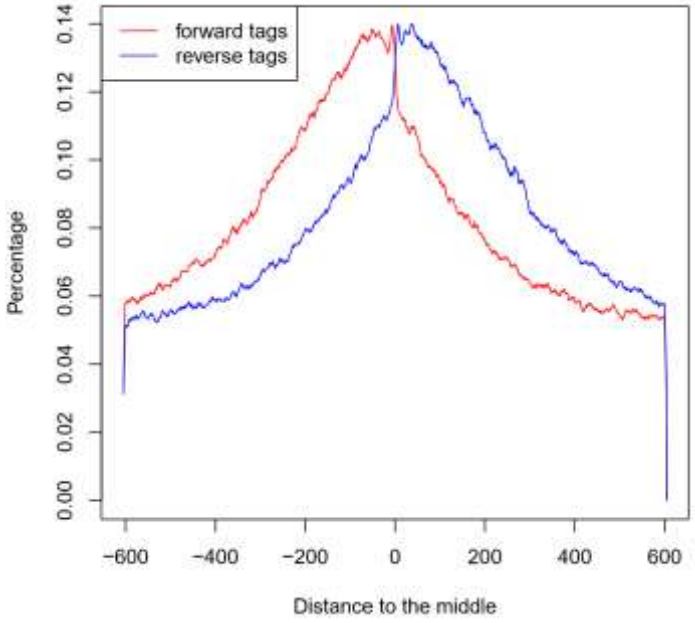
Modeling the shift size

The tag density around a true binding site should show a **bimodal enrichment pattern** (or paired peaks). MACS takes advantage of this bimodal pattern to empirically model the shifting size to better locate the precise binding sites.

To find paired peaks to **build the model**, MACS first scans the whole dataset searching for highly significant enriched regions. *This is done only using the ChIP sample!* Given a sonication size (`bandwidth`) and a high-confidence fold-enrichment (`mfold`), MACS slides two `bandwidth` windows across the genome to find regions with **tags more than `mfold` enriched relative to a random tag genome distribution**.

Open up the pdf file for Nanog-rep1. The first plot illustrates **the distance between the modes from which the shift size was determined.**





CHD4 sample 1
q = 0.01

Peak calls

At q=0.01- Lauren

- 1= 34758
- 2= 26324
- 3= 18518

High confidence: 23811 (present in 2 of 3)

At q = 0.01- Carrie

- 1 = 67353
- 2 = 48800
- 3 = 40349

With no q- Austin

1 = 84271

2 = 60749

3 = 50232

High confidence: 43818 (present in 2 of 3)