**BIOM262: ChIP-Seq workshop – Tuesday Feb 8th 2019**

**Introduction**:

This workshop will walk you through an example of ChIP-seq analysis. We will focus on running tools as command lines and simple bash scripts – I recommend having a cheat-sheet like this one <http://cheatsheetworld.com/programming/unix-linux-cheat-sheet/>.

We will use some common tools such as:

* bowtie2 for alignment (<http://bowtie-bio.sourceforge.net/bowtie2/index.shtml>),
* IGV for visualization (<http://software.broadinstitute.org/software/igv/home>)
* most of the workshop will be done using HOMER (*e.g.*, QC of the data, peak calling etc.; <http://homer.ucsd.edu/homer/>). HOMER was created by Chris Benner at UCSD, and I love the documentation and tutorials and the threaded humor.

During the workshop, and in general, it is always good to type the command and get the notes and use options of the command. Thus *e.g.*, typing bowtie2 would yield this output (capped after several lines):

[ucsd-train45@tscc-login1 ~]$ bowtie2

No index, query, or output file specified!

Bowtie 2 version 2.3.0 by Ben Langmead (langmea@cs.jhu.edu, www.cs.jhu.edu/~langmea)

Usage: bowtie2 [options]\* -x <bt2-idx> {-1 <m1> -2 <m2> | -U <r>} [-S <sam>]

**Part I**

We will start with FASTQ files and performs many of the basic analysis tasks that one might normally do when analyzing ChIP-seq data.

**1.** Align FASTQ reads using bowtie2.

The fastqs are at: ~/biom262\_2019/Module\_5/fastqs/

These files are originally from the following study investigating the roles that reprogramming factors play when transforming MEF (fibroblasts) into embryonic stem cells.

[Chronis et al. Cooperative Binding of Transcription Factors Orchestrates Reprogramming](https://www.ncbi.nlm.nih.gov/pubmed/28111071)  
Sequencing Data: [GSE90893](https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE90893)

For this tutorial we extracted the ChIP-seq experiments for several transcription factors and histone modifications performed on ESC (embryonic stem cells). To reduce runtimes, only reads that mapped to chr17 (and chr17\_random) are included. (the random chromosomes are explained in this link – <http://genome.ucsc.edu/FAQ/FAQdownloads#download10>.)

**Question**: how would you generate such a file with only one chromosome?

**TIP:** If you start overwriting your own line (which means you can no longer see what you typed earlier) try the following command: eval $(resize)

**Note:** if you get a memory error, grab an interactive node: qsub -I -l nodes=1:ppn=8 -l walltime=4:00:00 -q hotel

To align the reads we will use bowtie2. It is always a good practice to look at the manual of each tool you use, so you get an idea of options and documentation.

Initially, to get an idea of the command we will run it on one file as follows:

***bowtie2 -p 8 -x*** *~****/biom262\_2019/Module\_5/mm9/mm9\_index -U******~/biom262\_2019/Module\_5/fastqs/oct4-esc.chr17.2m.fastq | samtools view -bS -t ~/biom262\_2019/Module\_5/mm9/mm9.fa.fai >*** *<path to your bams directory>****/oct4-esc.chr17.2m.bam***

To do it properly, we will use a for loop to get BAMs from all the fastqs in the directory:

***for f in*** *<path to fastqs directory>****/\*fastq; do fname=`basename $f .fastq`; bowtie2 -p 8 -x ~/biom262\_2019/Module\_5/mm9/mm9\_index -U $f | samtools view -bS -t ~/biom262\_2019/Module\_5/mm9/mm9.fa.fai >*** *<path to bams directory>****/$fname.bam ; done***

This will produce BAM files for the 6 datasets. HOMER can analyze SAM files and if it receives BAM files it converts them to SAM so samtools has to be available (you can check that by typing “samtools” in the command line).

It is a good practice to always double check datasets before you start analyzing them. For instance, use samtools to view the files (samtools view -h bams/input-esc.chr17.2m.bam | less) and validate that the files are indeed what they should be (*e.g.*, aligned to chr 17, and have 2M reads). To calculate the number of reads do samtools view bams/input-esc.chr17.2m.bam | wc -l. If you want to understand better the way SAM files are organized you can follow <https://samtools.github.io/hts-specs/SAMv1.pdf> section 1.4.

**2**. Create a “tag directory” for the example Oct4 ChIP-seq experiment using the makeTagDirectory command. Start by typing *makeTagDirectory* (without any options) in your command line, it will provide the usage, some info about the command and a full list of program options – as I mentioned above, I highly recommend doing that whenever you use a new tool and a new command.

Tag directories are analogous to sorted bam files and are the starting point for most HOMER operations like finding peaks, creating visualization files, or calculating read densities. The command also performs several quality control and parameter estimation calculations. The command has the following form:

***makeTagDirectory <Output Tag Directory> [options] <input SAM/BAM file1> [input SAM/BAM file2] ...***

To create a tag directory for the Oct4 experiment, run the following command with recommended options:

***makeTagDirectory <****path to where your tag directories will be****>/oct4-esc -genome mm9 -checkGC <****path to your bam files****>/oct4-esc.chr17.2m.bam***

The command will take several seconds to run. What it is doing is parsing through the BAM file, removing reads that do not align to a unique position in the genome, separating reads by chromosome and sorting them by position, calculating how often reads appear in the same position to estimate the clonality (*i.e.* PCR duplication), calculating the relative distribution of reads relative to one another to estimate the ChIP-fragment length, calculating sequence properties and GC-content of the reads and performs a simple enrichment calculation to check if the experiment looks like a ChIP-seq experiment (vs. an RNA-seq experiment).

The command creates a new directory, in this case named “**oct4-esc**”. Inside the directory are several text files that contain various QC results. Try looking at the following by typing

less -S <filename>, or for all files use less -S tagdirs/oct4-esc/\*txt and scroll between files with :n for next and :p for previous.

**tagInfo.txt** - summary information from the experiment, including read totals.

**tagFreqUniq.txt** - nucleotide frequencies relative to the 5’ end of the sequencing reads.

**genomeGCcontent.txt** - distribution of ChIP-fragment GC%

**tagAutocorrelation.txt** - relative distribution of reads found on the same strand vs. different strands.

**tagCountDistribution.txt** - number of reads appearing at the same positions.

**3**. Now we will create tag directories for all samples, by following using a ‘for loop’.

***for f in <path to your bam files>/\*.bam; do fname=`basename $f .chr17.2m.bam`; makeTagDirectory <****path to where your tag directories will be****>/$fname -genome mm9 -checkGC $f; done***

So in my case I ran it like:

for f in ~/goren-module5/bams/\*bam; do fname=`basename $f .chr17.2m.bam`; makeTagDirectory ~/goren-module5/tagdirs/$fname -genome mm9 -checkGC $f; done

At this point you should have 6 tag directories. Look through the QC stats of the various ones.

**4**. Next we will visualize the ChIP-seq experiments by creating bedGraph files from the tag directories and using the IGV genome browser to look at the results. We will do this using the makeUCSCfile command. For most ChIP-seq experiments all you need to do is specify the tag directory and specify “-o auto” for the command to automatically save the bedGraph file inside the tag directory:

***makeUCSCfile*** *<Tag Directory>* ***-o auto***

For a specific dataset, *e.g.* Oct4, the command would be:

**makeUCSCfile <**path to your tag directories**>/oct4-esc/ -o auto**

This creates the file “oct4-esc/oct4-esc.ucsc.bedGraph.gz”. This file format specifies the normalized read depth at variable intervals along the genome (use *zmore* and the filename to view the file format for yourself).

Now make these for all samples:

***for dir in*** *<path to tag directories>/\*****; do makeUCSCfile $dir -o auto; done***

In my case I ran like:

for dir in ~/goren-module5/tagdirs/\*; do makeUCSCfile $dir -o auto ; done

To view the file in the genome browser, do the following:

* Download the files to your computer (***scp ucsd-train****<your number****>@tscc.sdsc.edu:/home/ucsd-train****<your number>****/****<full path to the file> <path to location to be copied to>*

or for all files:

***scp -r ucsd-train****<your number>****@tscc.sdsc.edu:/home/ucsd-train****<your number>/<full path to the directory>****\*/\*.gz*** *<path to location to be copied to>*

In my environment it is:

scp -r ucsd-train45@tscc.sdsc.edu:/home/ucsd-train45/goren-module5/tagdirs/\*/\*.gz ~/Google\ Drive//000-all-directories/0-GorenLab/Teaching/BIOM262/2019-biom262/data-import/

You can also use FileZilla for this if you have set up a connection with the cluster.

* Open IGV. Make sure you use the right genome (mm9; it is a good practice to have!) and drag the file to the center window (or select file -> load from file).
* The read pileups will display the relative density of ChIP-seq reads at each position in the genome. We only have data for chr17 in this example, so we can stick to that chromosome.

**5**. See if there are any interesting patterns in the data that catch your eye. Try visiting the Pou5f1 locus (the gene for Oct4) by typing the gene name into the search bar at the top. Once at the Pou5f1 locus, zoom out (alt+click or scale on top right) to see if there any nearby sites that might resemble enhancers.

Each dataset was created by a different antibody, and they can be divided into three types: TFs, HMs and global input. Since we will need to treat each type differently, I recommend making a directory for each – input, tfs and hms and move the tag directories to the relevant one (*e.g.* *tfs/oct4-esc/*, etc).