BIOM262: ChIP-Seq workshop

This workshop will walk you through an example of ChIP-seq analysis using HOMER (http://homer.ucsd.edu/homer/). HOMER was created by Chris Benner, and I love the documentation and tutorials. During the workshop, and in general, it is always to type the command and get the notes and use options of the command.

This workshop starts with aligned SAM files (only reads from mouse chr17) and performs many of the basic analysis tasks that one might normally do when analyzing ChIP-seq data.

0. Align FASTQ reads using bwa, bowtie2 or similar genome alignment algorithm. This will produce a SAM or BAM file that can be analyzed using HOMER. HOMER should be already installed and the mm9 genome is loaded in your environment.

1. Download the zip file containing SAM alignment files and unzip the archive.
cp /projects/ps-yeolab/biom262_2017/chip_seq_goren/samfiles.zip
unzip samfiles.zip
The archive should contain the following SAM files that have already been aligned to the mouse **mm9** genome:

- h3k27ac-esc.chr17.2m.sam
- h3k4me2-esc.chr17.2m.sam
- input-esc.chr17.2m.sam
- klf4-esc.chr17.2m.sam
- oct4-esc.chr17.2m.sam
- sox2-esc.chr17.2m.sam

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
Sequencing Data: GSE90893
For this tutorial we extracted the ChIP-seq experiments for several transcription factors and histone modifications performed ESC (embryonic stem cells). To reduce runtimes, only reads that mapped to chr17 (and chr17_random) are included in the SAM files (how would you generate such a file?).
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 data/sams/h3k27ac-esc.chr17.2m.sam | 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 -h data/sams/h3k27ac-esc.chr17.2m.sam | 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 – I highly recommend doing that whenever you use 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 file1> [input SAM 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 sam files>/oct4-esc.chr17.2m.sam
```

The command will take several seconds to run. What it is doing is parsing through the SAM 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 opening the following (either on the command line by typing *less -S <filename>*, or if you prefer by loading to R.

> **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 shell 'for loop'. First,

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

At this point you should have 7 tag directories. Look through the QC stats of the various ones – 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).

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>*esc; 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-login.sdsc.edu:/home/ucsd-train<your number>/<full path to the file> <path to location to be copied to>*; for instance in my environment it is:

  ```
  scp ucsd-train36@tscc-login.sdsc.edu:/home/ucsd-train36/data/tfs/oct4-esc/oct4-esc.ucsc.bedGraph.gz <path>/BIOM262/bedGraphs/
  ```

or for all files:

  ```
  scp -r ucsd-train36@tscc-login.sdsc.edu:/home/ucsd-train36/data/*/*/*.bedGraph.gz <path>/Teaching/BIOM262/bedGraph
  ```

- Open IGV. Make sure you use the right genome (mm9) 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 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.

6. One of the most common tasks with ChIP-seq data is to find 'enriched' regions commonly called "peaks". HOMER contains a command called findPeaks which is used to analyze tag directories for peaks. There are two common ways to use the command:

```
findPeaks <tag directory> -i <control tag directory> -style factor -o auto
findPeaks <tag directory> -i <control tag directory> -style histone -o auto
```

The difference between the two is in the "-style factor/histone" argument, which will tell the program to look for focal, fixed width peaks vs. variable length peaks; the later is more common in the case of histone modifications. To find Oct4 peaks in the data, run the following command:

```
findPeaks <path>/oct4-esc/ -i <path>/input-esc/ -style factor -o auto
```

This command will look for enriched regions and filter them based on several criterion, including ensuring that they have at least 4-fold more reads in peak regions relative to the control experiment (in this case "input-esc/"). The output will be stored in a HOMER-style peak file located in the Oct4 tag directory ("oct4-esc/peaks.txt"). The beginning of this file contains statistics and QC stats from the peak finding, including the number of peaks, number of peaks lost to input filtering, etc.

One field worth paying attention to is the "**Approximate IP efficiency**" which reports what fraction of reads from the experiment were actually found in peaks. For most decent experiments this value ranges from 1% to >30% (remember ChIP is an enrichment strategy... there is plenty of background in the data too!). Below this are the peaks along with enrichment statistics for each region.

One other thing to note is that HOMER reports the results in a 'peak' file, which has a slightly different format from a traditional BED file format. To create a BED file from the peak file, use the tool pos2bed.pl (*i.e.* **pos2bed.pl oct4-esc/peaks.txt > oct4-esc.bed**). BED files can be uploaded to IGV just like a bedGraph file. Also, most HOMER programs will work with either BED or peak files as input.

Next we will find peaks for all samples using two 'for loops' – for the two types of data:

for dir in <path>/**hms/*; do findPeaks $dir -i** <path>**input-esc/ -style histone -o auto; done**
and
for dir in <path>/**tfs/*; do findPeaks $dir -i** <path>**input-esc/ -style factor -o auto; done**

7. Now that we have identified peaks from our ChIP-seq data, it is time to figure out more information about where they are and what genes they might be regulating. HOMER contains a program called annotatePeaks.pl that performs a wide variety of functions using peak/BED files. First, lets use it to perform basic annotation of the peak file. The annotatePeaks.pl program works like this:
annotatePeaks.pl <peak/BED file> <genome version> [options] > output.txt
The "> output.txt" part at the end means that the results will be sent to stdout, and the "> output.txt" is used to capture the output information in a file. To annotate peaks from the Oct4 experiment:

annotatePeaks.pl <path>/**oct4-esc/peaks.txt mm9 > oct4.annotation.txt**

If we view the "oct4.annotation.txt" file with less -S, you'll see several annotation columns. Take note of the columns specifying the nearest gene TSS, the distance, and the annotation of the genomic region the peak is located in. This annotation is split into two separate columns - one is basic (i.e. exon, promoter, intergenic, intron etc.), and a more detailed annotation that describes CpG islands, repeat elements, etc. You might have also noticed while the command was running that stats about annotation enrichment too.

8. The annotatePeaks.pl program can also be used to create histograms that display the relative read enrichment relative to given genomic features, including transcription start sites (TSS) or any other set of regions the user wants to define. Since the TSS is so commonly used for this purpose, HOMER has a built-in annotation for TSS (based on RefSeq transcripts). The key parameters to create a histogram are the "-hist #" and "-size #" options, which control the binning size and total length of the histogram. The other important option is the "-d <tag directory>", which specifies which experiments to compile histograms for. In general:
annotatePeaks.pl <peak/BED file> <genome version> -size <#> -hist <#> -d <Tag Directory1> > output.txt

(note that the peak/BED file can be replaced with the key word "tss" to make a histogram at the TSS). To create a histogram with the experiments we've looked at thus far near the TSS, run the following:

```
annotatePeaks.pl tss mm9 -size 8000 -hist 10 -d <path>/oct4-esc/ <path>/sox2-esc/ <path>/h3k27me3-esc/ <path>/input-esc/ > output.txt
```

Open the “output.txt” using R. You’ll notice that the first column gives the distance offsets from the TSS followed by columns corresponding to the ‘coverage’, ‘+ Tags’, and ‘- Tags’ for each experiment. Try graphing each as X-Y line graph using the first column as the X-coordinate to see the patterns.