**BIOM262: ChIP-Seq workshop – Part 2**

Start by grabbing an interactive node: qsub -I -l nodes=1:ppn=8 -l walltime=4:00:00 -q hotel

Following the steps 1-5 on we did on Tuesday, you should have a tagdirectory for each of the datasets (6 in total) and you should also have the UCSCfiles.

As we discussed, each dataset was created by a different antibody, and they can be divided into three types: TFs (transcription factors) HMs (histone modifications) and global input (background). 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.).

**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***

or

***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**

And in my environment, I ran it like:

findPeaks ~/goren-module5/tagdirs/tfs/oct4-esc/ -i ~/goren-module5/tagdirs/input/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 (*e.g.* ***pos2bed.pl oct4-esc/peaks.txt > oct4-esc.bed***. 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.). 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**

Make a directory for the annotation files (*e.g.*, “annotations”) and convert the peak.txt and region.txt files to bed files:

***for dir in*** *<path>****/hms/\*; do dirname=${dir##\*/}; pos2bed.pl $dir/regions.txt > <path>/annotations/$dirname.bed; done***

And

***for dir in*** *<path>****/tfs/\*; do dirname=${dir##\*/}; pos2bed.pl $dir/peaks.txt > <path>/annotations/$dirname.bed; done***

Copy to your local computer using scp or filezila and load to IGV. Explore the original (previous to conversion to bed) peaks.txt and regions.txt files with ***less -S*** in the command line. For instance, each peak gets a score, see how the ones with high scores look on IGV vs. ones with low scores.

**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***

To annotate peaks from the Oct4 experiment:

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

You can do it using a for loop for the two types of datastes:

***for dir in*** *<path>/****hms/\*; do dirname=${dir##\*/}; annotatePeaks.pl $dir/regions.txt mm9 >*** *<path>/****annotations/$dirname.annotation.txt; done***

and for tfs:

***for dir in*** *<path>/****tfs/\*; do dirname=${dir##\*/}; annotatePeaks.pl $dir/peaks.txt mm9 >*** *<path>/****annotations/$dirname.annotation.txt; done***

**8**. The annotatePeaks.pl program can also be used to create histograms that display the 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 Directory>* ***> 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 near the TSS with the data from the experiments we’ve looked at thus far, run the following:

***annotatePeaks.pl tss mm9 -size 8000 -hist 10 -d*** *<path>****/tagdir/\*/\* >*** *<path>****/annotations/tss-histogram.txt***

Open the “tss-histogram.txt” using a spreadsheet program (R, Excel or similar). 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.

**Motif finding**

**9**. DNA motif finding is a powerful technique to analyze ChIP-seq experiments. Unlike gene expression data, ChIP-seq localizes signals to very specific regions of the genome allowing for accurate identification of the genetic signals responsible for recruiting various transcription factors. To use HOMER’s motif analysis program, run the findMotifsGenome.pl command using peak files from the experiments. In general the command works like this:

***findMotifsGenome.pl*** *<peak/BED file> <genome version> <output directory> [options]*

Common options for motif finding are “**-p** <#cpu>“ for parallel execution, and “**-size** <#>“ to specify the size of the regions you which to search for DNA motifs. For transcription factor motifs, a good size is 100. Make a new directory (*e.g.*, “motifs”). To find enriched motifs, run the following for loop:

***for dir in*** *<path>****/tfs/\*; do dirname=${dir##\*/}; mkdir*** *<path>****/motifs/$dirname; findMotifsGenome.pl $dir/peaks.txt mm9*** *<path>****/motifs/$dirname/ -mask -size 100 -p 10; done***

This command will perform several different steps, including checking for the enrichment of a library of known transcription factor motifs as well as perform a *de novo* search for enriched motifs. The “-mask” tells the program to mask repeat sequences in the genome. Depending on the speed of your server this program may take while to run. Once it’s finished, copy the homerResults.html files to your computer, and use your Internet Browser to open it.

To copy the entire directories:

***scp -r*** *<user>****@tscc.sdsc.edu:****<path>****/motifs/\**** *<directory on your local machine>*

This file will list the top *de novo* motifs found as well as provide stats and best matches to known transcription factor motifs. A second file called knownResults.html contains the enrichment statistics for known motifs.

You might notice that both analyses indicate that Oct4 and Sox2 peaks in ESC are highly enriched for an OCT:SOX composite motif, which has been shown to be very important in establishing pluripotent enhancers.

**10**. Now that you’ve found the most enriched motifs in a ChIP-seq experiment, it is worth it to see where those motifs are located (*i.e.*, which peaks, etc.). One of the key outputs from motif finding are “motif files”, which contain the information needed to understand where the motif is located in the genome. For example, the top de novo motifs found during motif finding are located in the output directory in the homerResults/ directory. To make it easier, lets copy the top Oct4 motif to the file “topOct4.motif” in the main directory where we are executing these commands (alternatively you could save the motif file down from the HTML results):

***cp*** *<path>****/motifs/oct4-esc/homerResults/motif1.motif*** *<path>****/topOct4.motif***

Now lets perform two separate analyses: first, lets create a histogram showing the motif positions relative to Oct4 peaks so we can check if it really looks enriched relative to the center of the Oct4 peaks. We can do this using the annotatePeaks.pl program like before, but instead of looking at read densities with the “-d <tag directory>“ option we can look at motif densities using the ‘-m <motif file>“ option instead:

***annotatePeaks.pl*** *<path>/****tagdirs/tfs/oct4-esc/peaks.txt mm9 -size 2000 -hist 10 -m*** *<path>****/topOct4.motif >*** *<path>****/output.Oct4motifHistogram.txt***

Once the command finishes, open **output.Oct4motifHistogram.txt** in a spreadsheet program and create an X-Y plot to examine the distribution of the motif relative to the peaks.  
  
Next, lets see where these motifs are located in the actual genome browser. To do this, we can run annotatePeaks.pl again, but this time we will not make a histogram and instead use the *“****-mbed <bedfile>***“ option to tell it to create a bed file of the motif positions that we can upload to the genome browser. Try the following:

***annotatePeaks.pl*** *<path>****/tagdirs/tfs/oct4-esc/peaks.txt mm9 -mbed*** *<path>****/topOct4.motifTrack.bed -m*** *<path>****/topOct4.motif > peaks-topOct4motif.txt***

Now you can copy and then load the “topOct4.motifTrack.bed” to IGV. In addition, the other output file “peaks-topOct4motif.txt” will contain the peak annotation results with an additional column showing the peaks that contain the given motif.

**11**. Finally, we want to gain some experience comparing ChIP-seq experiments. At first pass it might make sense to make a Venn diagram comparing peaks from two experiments to see how many overlap. This is a ***horrible*** way to analyze ChIP-seq data!!! This is because many peaks are close to the threshold of detection, barely making the cut for statistical significance (or not) in one experiment or another. A good practice is to create a scatter plot comparing the read counts between two experiments directly at all of the sites where there is signal (*i.e.*, peaks). To do this, first lets merge the peak files from the two experiments, collapsing peaks found that overlap:

**mergePeaks** <path>**/oct4-esc/peaks.txt** <path>/**sox2-esc/peaks.txt >** <path>/**oct4andsox2.peaks.txt**

When you run this command, you’ll notice that it will print out the numbers of overlapping and unique peaks from each file (*i.e.,* Venn diagram). This can be useful to give you a general idea of how similar the experiments are, but be careful not to over interpret these values. Now that you have the combined features in the “oct4andsox2.peaks.txt” file, lets use annotatePeaks.pl to quantify the read counts from each experiment at each of the peaks by specify each tag directory with the -d option like the following:

***annotatePeaks.pl*** *<path>/****oct4andsox2.peaks.txt mm9 -d*** *<path>/****oct4-esc/*** *<path>/****sox2-esc/ >*** *<path>****/oct4andsox2-scatter.txt***

We can not open the oct4andsox2-scatter.txt file in a spreadsheet program to directly look at the read counts at each peak to find those with low levels in one experiment or the other. Try creating an X-Y scatter plot of the read counts with log-transformed axes to get a sense for how different (or similar) each experiment is from one another.

Now that we have an appreciation for how similar the experiments are, lets try using the getDifferentialPeaks command to selective find peaks that are ‘specific’ to one experiment relative to another. In the following example we’ll specify “-F 2” to indicate that we want to find Sox2 peaks that are 2-fold higher in the Sox2 experiment relative to the Oct4 experiment:

***getDifferentialPeaks*** *<peak file to check> <target tag directory> <background tag directory>*

***-F*** *<fold change>* ***> output.txt***

and in our case:

**getDifferentialPeaks** <path>/**sox2-esc/peaks.txt** <path>/**sox2-esc/** <path>**/oct4-esc/ -F 2 >** <path>/**sox2-specific-peaks.txt**

Now that we have found Sox2 peaks that are relatively uniquely bound by Sox2 (and not Oct4), look at the peak file using ***less -S***, and look at several of the peaks in the genome browser to convince yourself that we have found interesting peaks.

Finally, lets use motif finding to see if we can identify what is unique about the DNA sequences in the Sox2 -specific peaks that might have lead to a differential recruitment of Sox2 versus Oct4 at these sites. Run motif finding on the Sox2 specific peaks:

***findMotifsGenome.pl*** *<path>/****sox2-specific-peaks.txt mm9*** *<path>/****motifs-sox2-specific/ -size 100 -p 10 -mask***

(reminder: this is the general command for finding peaks:

***findMotifsGenome.pl*** *<peak/BED file> <genome version> <output directory> [options]*)

**12.** Notice anything different about these results relative to the results from all Sox2 peaks that we found in step 9?