Tutorial: Deeptools

BIOSC 1540 Fall 2018

0. Using Secure FTP to transfer files between the CRC cluster and your computer

You will need a Secure FTP (SFTP) program to transfer the figures you create on the CRC cluster to your computer. See the tutorial on Motif Finding for how to install an SFTP program.

I. Preparing input files

Heatmaps and aggregate plots summarize read alignment pileup (in this case, ChIP-seq) relative to pre-defined genomic positions. Pileup is stored as **BigWig** format, which is a compressed version of a BEDGRAPH or Wiggle file, while genomic positions are represented as a BED file.

A. BigWig pileup file

There are a few ways to obtain a BigWig file of ChIP coverage. One convenient option is to allow your peak caller (macs2) to output pileup in addition to the peaks files. As an example, here's the `macs2` command from Assignment 5, modified to add the `--bdg` option so that it outputs a BEDGRAPH file (all one line):

```
$ macs2 callpeak --bdg -g 1.2e8 --outdir mychip_with_input -n
mychip_with_input -t my_chip.bam -c ~/hw5/input/my_input.bam
```

After `macs2` completes its run, there will be a couple of extra files in your output directory. The ChIP BEDGRAPH file will be called `mychip_with_input_treat_pileup.bdg` (depending on what you pass to the `-n` parameter, yours will have a different prefix in front of the `_treat_pileup.bdg`).

Next, you need to convert the BEDGRAPH file into a BigWig file. We will use a utility from the UCSC Genome Browser called `wigToBigWig`. `wigToBigWig` requires a file containing the sizes of the chromosomes. Such a file appropriate for zebrafish has been provided for you in the project directory. However in this case, we are using a file for *Drosophila pseudoobscura*:

```
$ wigToBigWig mychip_with_input_treat_pileup.bdg dp4_chr_sizes.txt mychip.bw
```

This command will produce a BigWig file called `mychip.bw` appropriate for Deeptools.

B. BED file

We will be generating heatmaps / aggregate plots that are centered on peaks that were called by macs2. Recall that the peaks BED file is called `mychip_with_input_peaks.narrowPeak` in this example.

However, many of the questions you will address in the project require you to compare two different BED files, e.g. a BED file of strong peaks versus a BED file of weak peaks. While it is possible to analyze each separately using Deeptools, it is more convenient to combine the two BED files together into a single file that Deeptools can interpret as containing two groups of regions.

Deeptools uses intervening lines that start with `#` to indicate different groups of regions. The following is a small example:

```
chr1  1     100
chr3  29    40
#Group1
chr1  49    199
chr2  3     50
chr2  30    40
#Group2
```

In this example, one group of regions (Group1) has two entries, while the other group (Group2) has three entries. The `#` line is inserted after all the regions in each group.

We can use Unix commands to generate such a combined BED file. Assume you have two BED files, `group1.bed` and `group2.bed`. To create `combined.bed`, issue the following commands:

```
$ cat group1 > combined.bed
$ echo "#Group1" >> combined.bed
$ cat group2 >> combined.bed
$ echo "#Group2" >> combined.bed
```

Note the use of the append (`>>`) directive to progressively build the `combined.bed` file. The `echo` command just prints what you tell it to back to the screen, hence `echo`.

II. Running computeMatrix

Deeptools is a software package designed for Seq data visualization. The individual Deeptools programs are already installed on the CRC cluster, and you can explore their functionality here https://deeptools.readthedocs.io/en/develop/index.html . We will use two Deeptools programs in succession.

`computeMatrix` is run first to generate a matrix file that is used for the subsequent commands. Depending on the size of your BigWig, `computeMatrix` may take some time to run. However, you only need to run it once for each BigWig/BED combination; you can reuse the matrix file for many subsequent commands.

The following is the command run with one possible set of parameters suitable for the project. You should definitely review the command options for computeMatrix on the help page https://deeptools.readthedocs.io/en/develop/content/tools/computeMatrix.html to determine if you want to run the command in a different way.

```
$ computeMatrix reference-point --referencePoint center -b 500 -a 500
--sortRegions descend --missingDataAsZero -S mychip.bw -R my_peaks.bed -o
mychip.matrix
```

This creates a matrix called `mychip.matrix` centered in each peak, and extending 500bp to the left (`-b`) and to the right (`-a`). The regions are sorted according to decreasing intensity of pileup.

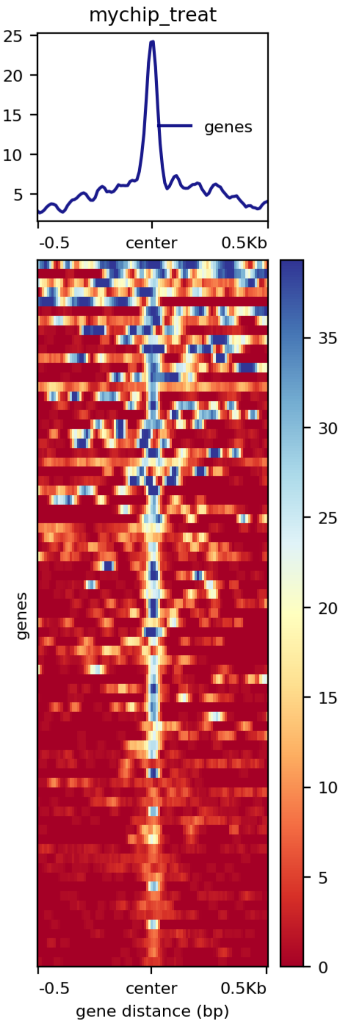
III. Running plotHeatmap

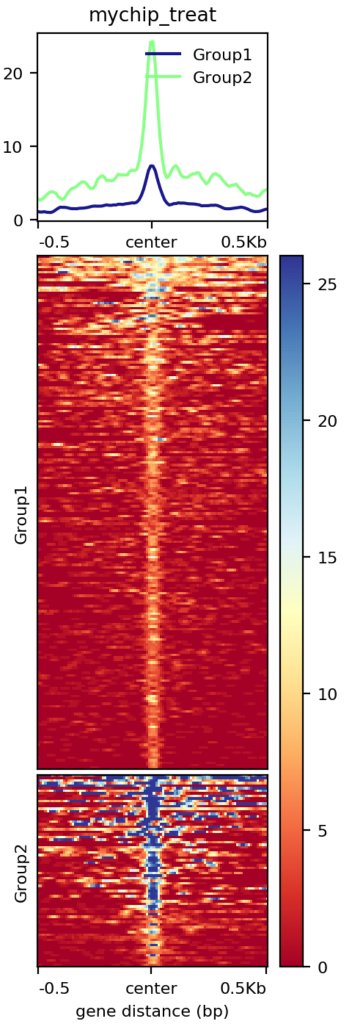
Once your matrix is constructed, you can run the plotHeatmap command:

```
$ plotHeatmap -m mychip.matrix -o mychip_heatmap.png --heatmapHeight 15
```

A PNG file called `mychip_heatmap.png` will be created with your heatmap. There are many image file types supported, which you can choose by simply changing the `.png` suffix to `.jpg` or `.pdf`, for example. For aesthetics, you can also play with the height parameter (it is in cm), along with many other parameters that are documented on the help page.

If you run this command with a standard BED file with one group, you will get a heatmap that looks something like the one on the left (you will need to use SFTP to transfer the image file to your computer to view it). If you run the command with a BED file where 2 groups were specified, it will look like the one on the right. The default command gives you both an aggregate plot and a heatmap. For multiple groups, the heatmaps are stacked, while the aggregate plots are overlaid.





Feel free to explore the other options to, for example, change the heatmap colors.
https://deeptools.readthedocs.io/en/develop/content/tools/plotHeatmap.html