EpiSyStem MS & Single Cell Workshop: Hands-On Tutorial

Objectives:

- Inspect the read quality
- Map reads on a reference genome
- Compare coverage files
- Call enriched regions or peaks
- Infer cell-type specific histone modification levels in the bone marrow

Estimated time required: approximately 3 hours.

Take home messages:

- Sequence affinity of PA-MNAse can be seen in the fastq files.
- Although scChIC-seq data is sparse at the single cell level, there is enough information to cluster cells into pseudobulk samples that reflect cell-type specific chromatin structures.
- We can define cell-type specific genomic regions by finding enriched signal along genomic regions.

Contributors:

- Anna Alemany
- Buys de Barbanson
- Vivek Bhardwaj
- Christoph Geisenberger
- Alexander van Oudenaarden
- Jake Yeung
- Peter Zeller

Software needed (hopefully already installed):

- fastqc
- samtools
- deeptools
- bwa
- hiddenDomains
- macs2
- R
- perl
- python

Introduction

Within a cell nucleus, the DNA is tightly-packed and the chromatin is spatially distributed with different levels and scales of organizations. At the smallest scale, DNA is packaged into units called nucleosomes, made of eight histone proteins.

Transcription factors (TFs) in concert with histone modifications shape the chromatin landscape of the genome, and thus regulate cell types and cell states. Histone modifications form an adaptable epigenetic regulatory layer that mediate dynamic transcriptional programs. Functional genomics assays, the most popular involving chromatin immunoprecipitation (ChIP), have revealed active and repressive chromatin structures in bulk tissues. However, inefficiencies of ChIP hinder its application in single cells, preventing genome-wide analysis of histone modifications along the continuum of cellular phenotypes. Therefore, how chromatin landscapes change between repressed, poised, and active states during development and homeostasis is relatively unexplored at the single-cell level.

Binding certain proteins to each of the eight histone proteins may modify the chromatin structure and may result in changes in transcription level. For example, the H3K4me3 is adding 3 methyl-group of the 4th Lysine in the histone 3 amino-acid. This modification is known to activate the transcription on nearby genes by opening the chromatin.

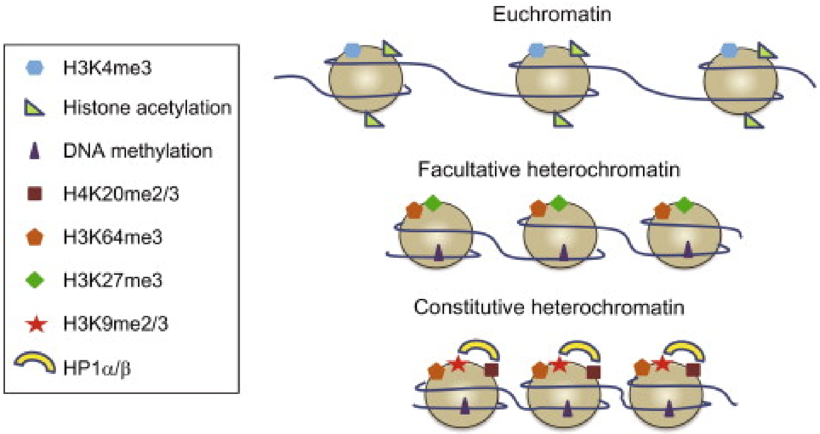

Figure 1: Fadloun et al, 2013

In the upcoming tutorial, we will look at the activator marks H3K4me1 and H3K4me3 scChIC-seq data from mouse bone marrow. We have already performed a dimensionality reduction (using a method called Latent Dirichlet Allocation) on the scChIC-seq in order to cluster cells (using a method called Louvain community detection) with similar histone modification profiles. The cell-cell relationships calculated from this analysis can be visualized in a 2-dimensional plot:

We will use this pre-defined clustering to explore scChIC-seq data. We suspect that the differences across cells could be coming from distinct cell types. In this exercise, we will focus on two clusters for each histone mark: for H3K4me1 clusters 2 and 5; for H3K4me3 clusters 3 and 5. We have already prepared the scChIC-seq data such that the individual cells are grouped into three clusters.

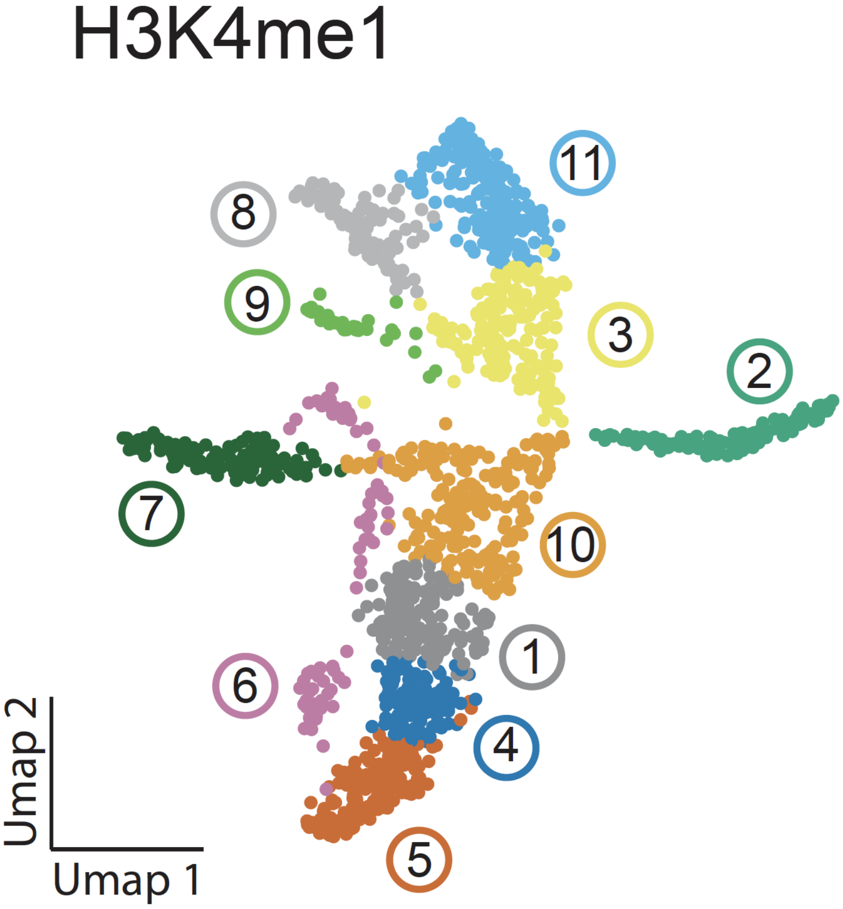

Figure 2: R2 per base seq content

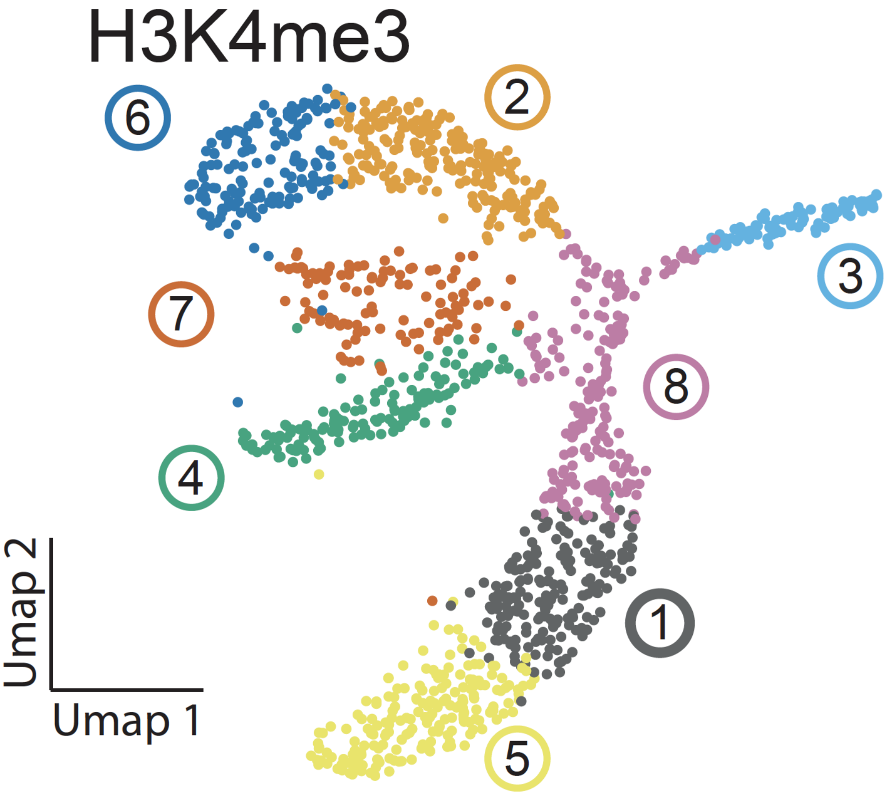

Figure 3: R2 per base seq content

Your job is to infer which cluster corresponds to which cell type.

Overview of files provided in this tutorial:

All file paths are relative to the data directory: `$HOME/Handouts/EpiSyStem_Workshop_Files` Remember you can navigate around the diretctories using `cd` and explore the files present in each direction using `ls`.

- `fastq_raw/raw_fastq_R${read}.fastq` : raw `fastq` files to see what raw data looks like before demultiplexing. You already had a look at these in the previous tutorial. ${read} denotes a bash variable that in this case can be equal to either 1 or 2.
- `fastq_full/demultiplexedR${read}_10000rows.fastq.gz` : demultiplexed `fastq` files for quality control checks before mapping. Again, `${read}` is a bash variable equal to 1 or 2. Note that this fastq files are zipped.
- `references/Mus_musculus.GRCm38.dna_rm.primary_assembly.fa`: reference genome `fasta` file.
- `sorted_bams_filtered/${histone_mark}_cluster_${clstrID}.filtered.bam` : single cell scChIC-seq profiles grouped by clusters. Here, `${histone_mark}` is either H3K4me1 or H3K4me3; and `${clstrID}` will be either 2, 5 or 11 for H3K4me1 and 3, 5 or 6 for H3K4me3. We have already assigned cells to clusters for you, you just have to infer the biological meaning of these clusters (i.e., infer the cell type). We will use these to visualize `bam` files with `IGV`, explore how to calculate number of reads by `MAPQ` quality, and do peak calling. `bam` files are subset to include only four main regions to reduce file size (defined in `regions/regions_to_filter.txt`).
- `regions/regions_to_filter.txt` : File containing the four genomic regions that contain signal in the `bam` files.
- `sorted_bigwigs/${histone_mark}_cluster_${clstrID}.bw` : `bigwig` files of scChIC-seq profiles, providing genome-wide coverage of scChIC-seq grouped by their clusters (pseudobulk). We will use `bigwig` files to correlate across pseudobulk samples and visualize the on the `IGV`.
- `chromsizes/chromsizes.${genome}.filt.txt` : size of genomes which are used as input in `hiddenDomains`.

Step 1: Quality control and treatment of the sequences

Let's first assess the quality of the demultiplexed `fastq` files. Demultiplexed means UMI and cell barcodes have been clipped from the `fastq` file and placed in the header.

Hands-on: First look at the `fastq` files

1. Using the terminal, go to the directory `Handouts/EpiSyStem_Workshop_Files/fastq_full`

2. Inspect the `fastq` files by using the `less` command. What differences do you detect when comparing the reads in these files to the reads in the `fastq` files available in the `fastq_raw` directory?
3. Run the command `head` on any of the two files. What is the output? Why do you think this does not work?
4. In order to explore `zipped` files, the `head/tail/cat` commands are not good enough. For this reason, there is the command `zcat` which has the same function as `cat` but only works with `zipped` files. Try it out by running the command `zcat demultiplexedR1_10000rows.fastq.gz`.

During sequencing, errors are introduced, such as incorrect nucleotides being called. These are due to the technical limitations of each sequencing platform. Sequencing errors might bias the analysis and can lead to a misinterpretation of the data.

Sequence quality control is therefore an essential first step in your analysis. We use here similar tools as described in "Quality control" tutorial: FastQC. Interpretation of fastqc outputs can be found here FastQC Output.

Hands-on: Quality control

1. Run **FastQC** on the fastq files

- `fastqc H3K4me3_demultiplexedR1_10000rows.fastq.gz` and `fastqc H3K4me3_demultiplexedR2_10000rows.fastq.gz`

2. Inspect the generated HTML files. To do so, run the command `open demultiplexedR1_10000rows_fastqc.html` and Enable the view.
3. Go back to the terminal by selecting with your mouse the corresponding tab in the top of your screen, and run the command `open demultiplexedR2_10000rows_fastqc.html`. Again by clicking with the mouse in the corresponding tab, you will be available to explore the contents of each HTML file.

Questions

1. How is the quality of the read 1 and read 2? Why do the lengths of the reads differ? Hint: Remember that the `fastq` files have been demultiplexed.
2. What are the most common start sequences for the `fastq` files? How does it differ for `R1` and `R2`?
3. Have the adapters been trimmed in the demultiplexed `fastq` files?

Step 2: Mapping of the reads

We obtain sequences corresponding to a portion of DNA linked to the histone mark of interest, H3K4me3 in this case. H3K4me3 is associated with active chromatin. It would be interesting to know if there is a difference across cell types.

Running BWA

Mapping requires a database of the genome to which you are mapping. These files often are downloaded from publicly available genome browsers such as Ensembl or UCSC Genome Browser. We have already downloaded the mouse genome (genome assembly `mm10`) and created an index used for mapping. The index file is used often in mapping programs to allow fast and efficient access to the large genome.

`bwa` is a widely used software that allows to map reads in a `fastq` file to their most likely position in a reference genome or reference transcriptome. This software can be used as a command in the terminal. If you type `bwa` alone in the termnal you will see a list of options that you can use to run it and perform the mapping. In this tutorial we will use the option `mem`. Now, if you type `bwa mem` in the terminal, a new list of options will be displayed on screen.

Hands-on: Mapping

1. Run `bwa` on the fastq files, using an mm10 reference genome (the forward slash `\` is only used to split a long single line into multiple lines, so the long command can be printed on a PDF page).

```
bash bwa mem ../references/Mus_musculus.GRCm38.dna_rm.primary_assembly.fa
\ demultiplexedR1_10000rows.fastq.gz demultiplexedR2_10000rows.fastq.gz
\ demux_map.sam
```

The output file containing all the mapping information is a `sam` file (in the example above, we gave it the name `demux_map.sam`, but you can change it an give it another name). An explanation of the SAM format can be found (HERE).

2. Inspect the reference file, which can be found in:

`Handouts/EpiSyStem_Workshop_Files/references` with the name `Mus_musculus.GRCm38.dna_rm.primary_assembly.fa`.

There is another command line, `grep`, which is very usefull to find string patterns in your files. Using the `cd` command, go to the `Handouts/EpiSyStem_Workshop_Files/references` directory, and type there:

```
grep '>' Mus_musculus.GRCm38.dna_rm.primary_assembly.fa
```

This command line prints on your screen only the lines of the file Mus_musculus.GRCm38.dna_rm.primary_assembly.fa that contain the character >. What are these lines?

3. Inspect the mapping stats. Go back to the folder where you created your sam file (Handouts/EpiSyStem_Workshop_Files/fastq_full) and explore your sam file with the commands that we learned so far (less, head, tail, cat,. . .).

   Questions

   - How many lines does the sam file have?
   - Have a look at the first 67 lines of your sam file. What do you see?
   - How do we see the length of each chromosome used in the mapping?

   Find all the lines in the sam file containing mapping information for the read with name (long line is split into multiple lines with \):

   ```
   Is:NS500414;RN:518;Fc:H2GV2BGX9;La:1;Ti:11101;CX:23815;CY:1073;\
   Fi:N;CN:0;aa:CACTCA;aA:CACTCA;aI:32;\
   LY:PZ-BM-m1-H3K4me1-2_H2GV2BGX9_S11;RX:CCT;RQ:GGG;BI:175;bc:\
   TGCTAATG;BC:TGCTAATG;QT:GGKKKKKK;MX:NLAIII384C8U3
   ```

   - What is each line? Help: visit this (link)
   - Where has this read been mapped?
   - Which is the quality of the mapping?

   Now have a look at the mapping information for read:

   ```
   Is:NS500414;RN:518;Fc:H2GV2BGX9;La:1;Ti:11101;CX:23241;\
   CY:4823;Fi:N;CN:0;aa:CACTCA;aA:CACTCA;aI:32;\
   LY:PZ-BM-m1-H3K4me1-2_H2GV2BGX9_S11;RX:ACA;RQ:GGG;BI:175;\
   bc:TGCTAATG;BC:TGCTAATG;QT:GGKKKKKK;MX:NLAIII384C8U3
   ```

   - Is it mapped? Where? What is the quality of the mapping?

SAMTOOLS

The sam file is so important that is has a command line on its own to speed up its inspection. This is samtools. First, let's type samtools in the terminal with no arguments: a summary of all the options will be displayed.

We will first inspect the flagstat option. Type in the terminal:

```
samtools flagstat demux_map.sam
```

> > **Questions**
> >
> > - What is the output telling us?
> > - How many pair reads are mapped to the same chromosome in `demux_map.sam`?
> > - How many reads are not mapped at all?
> > - In how many pair reads only one of the reads is mapped?
>
> We will now inspect the `view` option. What happens when you type the following in the terminal?
>
> ```
> samtools view demux_map.sam
> ```
>
> Importantly, commands in the terminal can be concatenated (or pipped). For that, the character `|` is used. For instance, type to following examples in the terminal. What are we getting in each case?
>
> ```
> samtools view demux_map.sam | wc
> samtools view demux_map.sam | head
> samtools view demux_map.sam | grep 'BC:CGTAGTGC'
> ```
>
> If you now type:
>
> ```
> samtools view
> ```
>
> a full list of new options is displayed. Answer the following questions using `samtools view` with the different options. You might need to pipe some commands to get the answer. > ### Questions > - How many reads do have the flag 99? > - How many reads do not have the flag 99? > - How many reads do have a mapping quality equal or higher than 60? > - List the first 5 reads that have a mapping quality equal or higher than 60 with a flag equal to 163. > - Print the SAM header only. > - How many reads with a mapping quality equal or higher than 60 do have the string `BC:CGTAGTGC` in their name?

From SAM to BAM format

Once you get used, the `sam` format is very convenient to explore how your sequencing data maps to the reference. However, the files become very big very fast. For this reason, the `bam` file was invented, which contains the same information as the `sam` file but in a compress manner. To convert your `sam` file into a `bam` file, type in the terminal:

```
samtools view -Sb demux_map.sam > demux_max.bam
```

, and you will see that the new `bam` file has been created (use `ls` to check). Commands such as `cat, head, tail` will not work on the `bam` file, but they do in combination with `samtools`. For instance, compare the outputs of the two following command lines:

```
head demux_max.bam
samtools view demux_max.bam | head
```

Correlation between samples

We will compare genome-wide correlation of H3K4me3 and H3K4me1 for different cell clusters.

To compute the correlation between the samples we are going to to use the QC modules of deepTools, a software package for the QC (quality check), processing and analysis of NGS data. Before computing the correlation a time consuming step is required, which is to compute the read coverage (number of unique reads mapped at a given nucleotide) over a large number of regions from each of the inputed BAM files. For this we will use the tool **multiBamSummary** . Then, we use **plotCorrelation** from deepTools to compute and visualize the sample correlation. This is a fast process that allows to quickly try different correlation methods and visual outputs.

In this tutorial we are interested in assessing H3K4me3 and H3K4me1 scChIC-seq samples. To save time, we have already done that and summarized the scChIC-seq data as bigwig files, which contains the read coverage.

Hands-on: Correlation between samples

1. Find the `bigwig` files (wit extension `.bw`) in the `EpiSyStem_Workshop_Files` directory. You should find two files for each chromatin modification (H3K4me1 clusters 2 and 5; for H3K4me3 clusters 3 and 5).

2. Compare all bigwigs using `multiBigwigSummary`. You can type `multiBigwigSummary` in the terminal to see all the options. We will use the following options:

- *"Choose computation mode"*: `Bins`
- *"Bin size in bp"*: `100000`
- *"Input bigwig files"*: the four imported `bigwig` files
- *"Output file"*: results.npz

Using these parameters, the tool will take bins of 100000 bp. For each bin the overlapping reads in each sample will be computed and stored into a matrix.

4. Check results using `plotCorrelation`. Remember that you can type `plotCorrelation` by itself in the terminal to see the list of options. Use to following parameters to explore the file `results.npz`:

- *"Correlation method"*: `Pearson` or `Spearman` (which do you think is more appropriate here? Check by inspecting the scatterplots)

- Plot `heatmap` or `scatterplot`.
- In the scatter plot, you can plot output in log scale (–log1p) for visualization. What happens if you do not use this option?

You can visualize the output by using the command `open` in the terminal. To go back to the terminal, select the corresponding tab in the top of your screen.

Questions

- From the correlation plot, can you infer which clusters correspond to the same cell type in H3K4me1 and H3K4me3?

Step 4: Exploring `bam` files on the IGV browser

Now, go to the directory `EpiSyStem_Workshop_Files/sorted_bams_filtered`, which contains a 6 different `bam` files. These files have been prepared by the instructors and only contain only reads falling in specific genomic regions, in order to reduce the file size. For each modification, we have clustered single cells into three separate `bam` files, associated with one of three cell types: erythroblast, granulocytes, and B-cells. Your job is to explore which `bam` file is associated with which cell type by looking at cell-type specific regions in the genome browser.

Download the `.bam` and `.bam.bai` files onto your computer (`.bam.bai` files are important for hte IGV browser to be able to read the contents of the `bam` file). In order to download the files, select with your mouse the `Files` tab, go to the `EpiSyStem_Workshop_Files/sorted_bams_filtered` folder, select all the files present and create a zip folder. Next, download the zip folder onto yoru computer and unzip it.

Open first the 3 `bam` files belonging to the modification H3K4me1 using IGV, which should already be installed on your computer. And explore the following cell-type specific regions:

chr7	114972165	116898716
chr7	103325741	104425479
chr3	90523036	90870443
chr11	44114099	45269522

Questions:

1. Can you infer which clusters correspond to which cell types based on the coverage around the four regions?

Hint: *Hbb* is a standard marker for erythroblast, *S100a8* is a standard marker for granulocytes, and *Ebf1* is a standard markers for B cells.

Repeat the same procedure with the modification H3K4me3.

Step 5: Detecting enriched regions (peak calling)

We could see in the scChIC-seq data some enriched regions that differ across samples. We now would like to call these regions to obtain their coordinates, using `hiddenDomains`. Perhaps the number of peaks predicted in each region across each sample may be clues to what cell type each sample is.

Go back to CoCalc, and using the terminal go to the directory (in case you are not there):

```
EpiSyStem_Workshop_Files/sorted_bams_filtered
```

Hands-on: Peak calling with `hiddenDomains`

1. Calling peaks with `hiddenDomains`. Remember you can write `hiddenDomains` by itself in the terminal and a list of options will be shown. You can also find more information here. For now, we need the following required inputs:

- -g Size of chromosomes for the mouse genome. Can you find this file?
- -q Minimum MAPQ score. What is an appropriate MAPQ score to use? A low MAPQ score may include reads that are poor quality, while high MAPQ score keeps only high quality reads. For now, we will keep reads with a quality threshold equal to 60.
- -p A threshold to remove domains called with probabilities less than `p`. Set to a value such as 0.5. You can play around with this value to see how it changes the output.
- -b minimum length. Use the default 1000 bp.

Run this for all the bam files.

Questions

1. Which type of files were generated? What do they include?
2. How do the peaks called differ between samples? Can you infer cell types based on this analysis?
3. Download all the `BED` files that have been produced by `hiddenDomains` in your computer and open them using `IGV`. For that, you need to go the tab `Files` in the top of your screen with your mouse, and navigate throught the directories until you find the `BED` files. Once there, select them, create a compress folder, and download that folder onto your computer. In your

computer, you can unzip the folder and load the files in `IGV`.

Questions

1. How do the cell-type specific regions look like?
2. How do the peaks called differ between samples? Can you infer cell types based on this analysis?

Conclusion

We learned to explore `fastq` and `bam` files as well as do calculations on them. We visualized `bam` files and `bigwig` files to see how reads are mapped on the genome. Finally, we inferred genomic regions of interest by defining cell-type specific levels of histone marks.