An Introduction to ChIP-seq Data Analysis

Myrto Kostadima < *kostadim@ebi.ac.uk* >
Sandra Cortijo < *sandra.cortijo@slcu.cam.ac.uk* >

Advanced ChIP-seq data analysis
University of Cambridge
February, 2018

General information

The following standard icons are used in the hands-on exercises to help you locating:

 Important Information

 General information / notes

 Follow the following steps

 Questions to be answered

 Warning – PLEASE take care and read carefully

 Optional Bonus exercise

 Optional Bonus exercise for a champion

Resources used

BWA: https://github.com/lh3/bwa

Samtools: http://www.htslib.org/

WiggleTools: https://github.com/Ensembl/WiggleTools

Bedtools: http://code.google.com/p/bedtools/

MACS2: https://github.com/taoliu/MACS/

MEME: http://meme.sdsc.edu/meme/cgi-bin/meme.cgi

TOMTOM: http://meme.sdsc.edu/meme/cgi-bin/tomtom.cgi

Additional resources:

Ensembl: http://www.ensembl.org

Original Data from: http://encodeproject.org

Introduction

The goal of this hands-on session is to perform the basic steps of the analysis of ChIP-seq data, as well as some downstream analysis. The first step includes an unspliced alignment for a small subset of raw reads. We will align raw sequencing data to the human genome using *BWA* and then we will manipulate the SAM output in order to visualise the alignment on *Ensembl*. Then based on these aligned reads we will find immuno-enriched areas using the peak caller *MACS2*. We will then perform functional annotation and motif analysis on the predicted binding regions.

Data

The data we will use for this practical comes from the ENCODE (Encyclopedia of DNA Elements) Consortium, a big international collaboration aimed at building a comprehensive catalogue of functional elements in the human genome. As part of this project, many human tissues and cell lines were studied using high-throughput sequencing technologies.

In this practical, we will work on data sets from a blood cell line, GM12878, a lymphoblastoid cell line produced from the blood of a female donor of European ancestry.

Specifically, we will look at binding data for the transcription factor PAX5. PAX5 is a known regulator of B-cell differentiation. Aberrant expression of PAX5 is linked to lymphoblastoid leukaemia.

Prepare the environment

The data used in this practical can be found in the ChIP-seq directory on your desktop. Throughout this practical we will try to identify potential transcription factor binding sites of PAX5 in human lymphoblastoid cells.

Open the Terminal.

First, go to the right folder, where the data are stored.

```
cd ~/Desktop/ChIP-seq
```

The `.fastq` file that we will align is called `PAX5.fastq`. This file is based on PAX5 ChIP-seq data produced by the Myers lab in the context of the ENCODE project. We will align these reads to the human genome.

Alignment

There are a number of competing tools for short read alignment, each with its own set of strengths, weaknesses, and caveats. Here we will use *BWA*, a widely used ultrafast, memory efficient short read aligner.

BWA has various functions. To view them all type

```
bwa
```

BWA uses indexed genome for the alignment in order to keep its memory footprint small. Because of time constraints we will build the index only for one chromosome of the human genome. For this we need the chromosome sequence in fasta format. This is stored in a file named `GRCh38.fa`, under the subdirectory `genome`.

Before running the following command, please create a folder named `bwa_index`, if it does not already exist, under which you will store the indexed genome.

Then index the genome (in our case just one chromosome) using the *bwa index* command:

```
bwa index -p bwa_index/grch38 genome/GRCh38.fa
```

While the indexing is running in a separate terminal window type:

```
bwa index
```

Question

1. What does the option '-p' in the above command do? _____

Now that the genome is indexed we can move on to the actual alignment. This is done in two different steps, using 'bwa aln' that finds the sequence alignment coordinates for each read followed by 'bwa samse' that report the alignments in the SAM format given single-end reads. The latter command also randomly choses one of the multihits to report.

Align the PAX5 reads using *BWA*:

```
bwa aln bwa_index/grch38 PAX5.fastq > PAX5.sai
bwa samse bwa_index/grch38 PAX5.sai PAX5.fastq > PAX5.sam
```

The second command above outputs the alignments in SAM format and stores them in the file `PAX5.sam`.

Have a look at the SAM format by typing:

```
head -n 10 PAX5.sam
```

Questions

1. Can you distinguish between the header of the SAM format and the actual alignments? _____
2. What kind of information does the header provide you with? _____
3. To which chromosome are the reads mapped? _____

Manipulate SAM output

SAM files are rather big and when dealing with a high volume of HTS data, storage space can become an issue. We can convert SAM to BAM files (their binary equivalent files that are not human readable) that occupy much less space.

Convert SAM to BAM using samtools and store the output in the file `PAX5.bam`. You have to instruct samtools that the input is in SAM format (`-S`), the output should be in BAM format (`-b`) and that you want the output to be stored in the file specified by the `-o` option:

```
samtools view -O BAM -o PAX5.bam PAX5.sam
```

Visualise alignments in Ensembl

It is often instructive to look at your data in a genome browser. Here, we use Ensembl, a web-based browser. Other popular stand-alone browsers, which have the advantage of being installed locally and providing fast access, include IGV and SeqMonk. Genome browsers not only allow for more polished and flexible visualisation, but also provide easy access to a wealth of annotations and external data sources. This makes it straightforward to relate your data with information about repeat regions, known genes, epigenetic features or areas of cross-species conservation, to name just a few. As such, they are useful tools for exploratory analysis.

Visualisation will allow you to get a 'feel' for the data, as well as detecting abnormalities and problems. Also, exploring the data in such a way may give you ideas for further analyses. Ensembl supports various file formats for visualisation. Please check thei

website http://www.ensembl.org/info/website/upload/index.html#formats for all the formats that can be displayed. For our visualisation purposes we will use the bigWig and BAM format.

When uploading a BAM file into the genome browser, the browser will look for the index of the BAM file in the same folder where the BAM files is. The index file should have the same name as the BAM file and the suffix `.bai`. Finally, to create the index of a BAM file you need to make sure that the file is sorted according to chromosomal coordinates.

Sort alignments according to chromosome position and store the result in the file with the prefix `PAX5.sorted`:

```
samtools sort -o PAX5.sorted.bam -O BAM PAX5.bam
```

Index the sorted file.

```
samtools index PAX5.sorted.bam
```

The indexing will create a file called `PAX5.sorted.bam.bai`. Note that you don't have to specify the name of the output index file when running `samtools`.

Another way to visualise the alignments is to convert the BAM file into a bigWig file. The bigWig format is for display of dense, continuous data. The data will be displayed as a graph and the resulting bigWig files are in an indexed binary format.

The BAM to bigWig conversion takes place in two steps. First, we convert the BAM file into a wig, called `PAX5.wig`, using a tool *WiggleTools*.

To find the structure of the command and the mandatory arguments type:

```
wiggletools --help
```

Now generate the wig file, called `PAX5.wig`, by typing:

```
wiggletools write PAX5.wig PAX5.sorted.bam
```

Then we convert the wig into a binary graph, called `PAX5.bw`, using the tool `wigToBigWig` from the *UCSC tool kit*:

Apart from the wig file, we also need to provide the size of the chromosomes for the organism of interest in order to generate the bigWig file. These have to be stored in a tab-delimited file. When using the UCSC Genome Browser, Ensembl, or Galaxy, you typically indicate which species/genome build you are working. The way you do this for bedtools is to create a "genome" file, which simply lists the names of the chromosomes (or scaffolds, etc.) and their size (in basepairs) in a tab-delimited format.

To obtain chromosome lengths for the human genome, if we have already downloaded a multi-fasta file that contains the genome sequence, then we can run:

```
samtools faidx genome/GRCh38.fa
```

This will create a tab-delimited file called `GRCh38.fa.fai` stored under the `genome`folder. Finally to create the bigWig file type:

```
wigToBigWig -fixedSummaries PAX5.wig genome/GRCh38.fa.fai PAX5.bw
```

Now we will load the data onto the *Ensembl* genome browser for visualisation. Open an internet browser and go to http://www.ensembl.org On the middle left of your screen under `Favourite genomes` choose Human GRCh38. On the new window enter the following genomic coordinates '1:45562933-45641920' in the search box and click 'Go'.

Now in order to load the desired files go to on the left hand side menu and click on Custom tracks. Since our BAM file is larger than the 20MB limit posed by Ensembl, we will load it through a URL link. Select an appropriate name for the PAX5 BAM file and add the following URL in the 'Data' box: http://www.ebi.ac.uk/~kostadim/ChIP-seq_EDI_2015/PAX5.sorted.bam. Then select the correct file format from the drop down menu and click 'Add Data'. Once the file is loaded, click anywhere on the screen to return to the genome browser view.

Follow the same steps in order to load the PAX5 bigWig file.

Question

1. Can you see a PAX5 binding site near the NASP gene? _____

_____

Using the "+" button on the top right zoom in more to see the details of the alignment.

2. What is the main difference between the visualisation of BAM and bigWig files?

_____

Alignment of the control sample

In the main data folder you will find another `.fastq` file called `Control.fastq`. Follow the steps described above in order to align the control reads to the human genome as well.

Finding enriched areas using MACS

MACS2 stands for model based analysis of ChIP-seq. It was designed for identifying transcription factor binding sites. *MACS2* captures the influence of genome complexity

to evaluate the significance of enriched ChIP regions, and improves the spatial resolution of binding sites through combining the information of both sequencing tag position and orientation. *MACS2* can be easily used for ChIP-Seq data alone, or with a control sample to increase specificity.

Consult the *MACS2* help file to see the options and parameters.

```
macs2 --help
macs2 callpeak --help
```

The input for *MACS2* can be in ELAND, BED, SAM, BAM or BOWTIE formats (you just have to set the `--format` flag). Options that you will have to use include:

- `-t` to indicate the input ChIP file
- `-c` to indicate the name of the control file
- `--format` the tag file format. If this option is not set MACS automatically detects which format the file is.
- `--name` to set the name of the output files
- `--gsize` This is the mappable genome size. With the read length we have, 70% of the genome is a fair estimation. Since in this analysis we include only reads from chromosome 1, we will use as gsize 70% of the length of chromosome 1 (197 Mb). MACS also offers shortcuts for human, 'mm' for mouse 'ce' for C. elegans and 'dm' for fruitfly.
- `--call-summits` when this option is set MACS detects all subpeaks in each enriched region and returns their summits
- `--pvalue` the P-value cutoff for peak detection.

Now run macs using the following command:

```
macs2 callpeak -t [PAX5 aligned bam file] -c [Control aligned bam file]
   --format BAM --name PAX5 --gsize 138000000 --pvalue 1e-3
   --call-summits
```

MACS2 generates its peak files in a file format called `.narrowPeak` file. This is a BED format describing genomic locations. Many types of genomic data can be represented as (sets of) genomic regions.

In the following section we will look into the BED format in more detail, and we will perform simple operations on genomic interval data.

Operating on genomic regions using `bedtools`

In this section we perform simple functions, such as overlaps, on the most common file type used for describing genomic regions, the BED file. We will examine the results of the ChIP-seq peak calling you have performed on the transcription factor PAX5 and

perform simple operations on these files, using the **bedtools** suite of programs. You will then annotate the *MACS2* peaks with respect to genomic annotations. Finally, we will select the most significantly enriched peaks, and extract the genomic sequence flanking their summits, the point of highest enrichment.

File formats

Over the years a set of commonly used file formats for genomic intervals have emerged. Most of these file formats are tabular where each row consists of an interval and columns have a pre-defined meaning, describing chromosomes, locations, scores, etc. The *UCSC* web browser has an informative list of these at http://genome.ucsc.edu/FAQ/FAQformat.html.

The BED format is the simplest file format of these. A minimal bed file has at least three columns denoting chromosome, start and end of an interval. The following example denotes three intervals, two on chromosome 1 and one on 2:

1	50	100
1	500	1000
2	600	800

Bed files follow the *UCSC* Genome Browser's convention of making the start position 0-based and the end position 1-based. In other words, you should interpret the "start" column as being 1 base pair higher than what is represented in the file. For example, the following BED feature represents a single base on chromosome 1; namely, the 1st base.

1	0	1	I-am-the-first-position-on-chrom-1

Using the bed format documentation (which can be found at http://genome.ucsc.edu/FAQ/FAQformat.html#format1)answer the following questions.

Questions

1. The simplest bed file contains just three columns. This is sometimes called BED3 format. What does BED6 contain? _____

2. In the above examples, what are the lengths of the intervals? _____

3. The BED format contains support for non-consecutive intervals. Can you output a bed-format with a transcript called "loc1", transcribed on the forward strand and having three exons of length 100 starting at positions 1000, 2000 and 3000?

The narrowPeak format is a BED6+4 format used to describe and visualise called peaks. Previously, we have used *MACS2* to call peaks on the PAX5 ChIP-seq data set. Look at the first 10 lines of the `PAX5_peaks.narrowPeak` by typing:

```
head -n 10 PAX5_peaks.narrowPeak
```

Questions

1. What additional information is given in the narrowPeak file, beside the location of the peaks? _____

Hint: See http://genome.ucsc.edu/FAQ/FAQformat.html#format12 for details.

NarrowPeak files cannot be uploaded onto Ensembl, but would be possible to be uploaded to *IGV*, which is a standalone genome browser. To visualise the peaks onto Ensembl, we would have to convert it to a simple BED3 format, extracting only the first 3 columns for each peak; the chromosome, the start and the end of the peak. To do so type:

```
cut -f 1-3 PAX5_peaks.narrowPeak > PAX5_peaks.bed
```

Upload the peak file generated by *MACS2* to *Ensembl*. Find the coordinates of a peak in the file (use the head command to view the beginning of the `.bed` file we just generated).

Questions

1. Does the first peak that was called look convincing to you? _____
2. Zoom out a bit and look at other nearby peaks? Do they all look convincing to you? _____

A second popular format is the GTF format. Each row in a GTF formatted file denotes a genomic interval. The 9th column permits intervals to be grouped and linked in a hierarchical fashion. This format is thus popular to describe gene models.

The three intervals from above might be:

1	gene	exon	51	100	.	+	0	gene_id "001"; transcript_id "001.1";

1	gene	exon	501	1000	.	+	2	gene_id “001”; transcript_id “001.1”;
2	repeat	exon	601	800	.	+	.	

Note how the first two intervals are linked through a common transcript_id and gene_id.

Questions

The gtf format documentation can be found at http://mblab.wustl.edu/GTF2.html.

1. In the small example above, why have the coordinates changed from the BED description? _____

The aim of the GENCODE project is to annotate all evidence-based genes and gene features in the entire human genome at a high accuracy. Annotation of the GENCODE gene set is carried out using a mix of manual annotation, experimental analysis and computational biology methods. The latest GENCODE gene set is available through *Ensembl* and can be found in the genome folder, called `Homo_sapiens.GRCh38.86.gtf`.

Look at the first 10 lines of the GENCODE annotation file:

```
head -n 10 genome/gencode.v27.annotation.gtf
```

The **bedtools** package permits complex, interval-based manipulation of BED and GTF files. They are also very fast. The general invocation of **bedtools** is **bedtools** <COMMAND>.

To get an overview of the available commands, simply call **bedtools** without any command or options in the terminal window.

```
bedtools
```

To get help for a command, type **bedtools** <COMMAND>. Extensive documentation and examples are available at https://bedtools.readthedocs.org/en/latest/

We will now use bedtools to calculate simple coverage statistics of the peak calls over the genome (keep in mind that only peaks on Chromosome 1 are in the file).

To bring up the help page for the **bedtools genomecov** command, type:

```
bedtools genomecov
```

Calculate the genome coverage of the PAX5 peaks:

```
bedtools genomecov -i PAX5_peaks.narrowPeak -g genome/GRCh38.fa.fai
```

Questions

1. What percentage of chromosome 1 do the peaks of PAX5 cover? _____

In order to biologically interpret the results of ChIP-seq experiments, it is useful to look at the genes and other annotated elements that are located in proximity to the identified enriched regions. We will now use **bedtools** to identify how many PAX5 peaks overlap GENCODE genes.

First we use awk to filter out only the genes from the gtf file:

```
awk '$3=="gene"' genome/gencode.v27.annotation.gtf  >
   genome/gencode.v27.annotation.genes.gtf
```

Count the total number of PAX5 peaks:

```
wc -l PAX5_peaks.narrowPeak
```

Use **bedtools** to find the number overlapping GENCODE genes:

```
bedtools intersect -a PAX5_peaks.narrowPeak -b
   genome/gencode.v27.annotation.genes.gtf | wc -l
```

You can use the **bedtools closest** command to find the closest gene to each peak.

```
bedtools closest -a PAX5_peaks.narrowPeak -b
   genome/gencode.v27.annotation.genes.gtf | head
```

Questions

1. What proportion of PAX5 peaks overlap genes? Does this proportion surprise you? _____
2. Which gene was found to be closest to MACS peak 2? _____

Transcription factor binding near to the transcript start sites (TSS) of genes is known to drive gene expression or repression, so it is of interest to know which TSS regions are bound by PAX5. To determine this, we will first create a BED file of the GENCODE TSS using the GTF.

You can use this awk command to create the TSS BED file:

```
awk 'BEGIN {FS=OFS="\t"} { if($7=="+"){tss=$4-1} else { tss = $5 } print
   $1,tss, tss+1, ".", ".", $7, $9}'
   genome/gencode.v27.annotation.genes.gtf | sort -k1,1 -k2,2n >
   genome/gencode.v27.annotation.tss.bed
```

Now use the **bedtools closest** command again to find the closest TSS to each peak:

```
bedtools closest -a PAX5_peaks.narrowPeak -b
   genome/gencode.v27.annotation.tss.bed > PAX5_closestTSS.txt
```

Use **head** to inspect the results.

You have now matched up all the PAX5 transcription factor peaks to their nearest gene transcription start site. You will see later in the course how you can search for functional enrichments within gene sets.

Annotation: From peaks to biological interpretation

Another way to look at the genes and other annotated elements that are located in proximity to the identified enriched regions is through the use of a tool called *PAVIS* http://manticore.niehs.nih.gov/pavis2/.

In a web browser, go to http://manticore.niehs.nih.gov/pavis2/

In the **Species/Genome Assembly/Gene Set** dropdown menu select **Ensembl_GRCh38/hg38 all genes**.

Fill in the location of the peak file `PAX5_peaks.bed`, and leave the default parameters for the remaining options.

Click on **SUBMIT** to run the tool.

Have a look at the pie charts shown on the results page to get an idea at which genomic locations Oct4 is most often found to bind.

This list of closest downstream genes found under the link **The Full Annotation File** can be the basis of further analysis. For instance, you could look at the Gene Ontology terms associated with these genes to get an idea of the biological processes that may be affected. Web-based tools like DAVID (http://david.abcc.ncifcrf.gov) or GOstat (http://gostat.wehi.edu.au) take a list of genes and return the enriched GO categories.

Motif Analysis

It is often interesting to find out whether we can associate the identified binding sites with a sequence pattern or motif. To do so, we will identify the summit regions of the strongest PAX5 binding sites, retrieve the sequences associated with these regions, and use *MEME* for motif analysis.

Since many peak-finding tools merge overlapping areas of enrichment, the resulting peaks tend to be much wider than the actual binding sites. Sub-dividing the enriched areas by accurately partitioning enriched loci yields a finer-resolution set of individual binding sites. The location of strongest enrichment signal in a peak is often called the *summit*. The summit and its vicinity are the best estimate for the true protein binding site, and so it is here where we look for repeated sequence patterns, called *motifs*, to which the transcription factor may preferentially bind.

Since many peak-finding tools merge overlapping areas of enrichment, the resulting peaks tend to be much wider than the actual binding sites. Sub-dividing the enriched areas by accurately partitioning enriched loci into a finer-resolution set of individual binding sites, and fetching sequences from the summit region where binding motifs are most likely to appear enhances the quality of the motif analysis. Sub-peak summit sequences have already been called by *MACS2* with the `--call-summits` option.

De Novo motif finding programs take as input a set of sequences in which to search for repeated short sequences. Since motif discovery is computationally heavy, we will restrict our search for the PAX5 motif to the genome regions around the summits of the 300 most significant PAX5 subpeaks on Chromosome 1.

Sort the PAX5 peaks by the height of the summit (the maximum number of overlapping reads).

```
sort -k5 -nr PAX5_summits.bed > PAX5_summits.sorted.bed
```

Using the sorted file, select the top 300 peaks and create a BED file for the regions of 60 base pairs centred around the peak summit.

```
awk 'BEGIN{FS=OFS="\t"}; NR < 301 { print $1, $2-30, $3+29 }'
   PAX5_summits.sorted.bed > PAX5_top300_summits.bed
```

The human genome sequence is available in FASTA format in the `bowtie_index` directory. You can now use `bedtools` to extract the sequences around the PAX5 peak summits in FASTA format, which we save in a file named `PAX5_top300_summits.fa`.

```
bedtools getfasta -fi genome/GRCh38.fa -bed PAX5_top300_summits.bed -fo
   PAX5_top300_summits.fa
```

We are now ready to perform `de novo` motif discovery, for which we will use the tool *MEME*.

Open a web browser, go to the MEME website at http://meme-suite.org/, and choose the 'MEME' tool. Fill in the necessary details, such as:

- the sub-peaks fasta file `PAX5_top300_summits.fa` (will need uploading), or just paste in the sequences.
- the number of motifs we expect to find (1 per sequence)

- the width of the desired motif (between 6 to 20) in the Advanced options
- the maximum number of motifs to find (3 by default). For PAX5 one classical motif is known.

Start Search. The results page will refresh automatically and once the tool has finished running, please follow the link **"MEME html output"**

Scroll down until you see the first motif logo. We would like to know if this motif is similar to any other known motif. We will use TOMTOM for this. Click under the option **Submit/Download** and choose the TOMTOM button to compare to known motifs in motif databases, and on the new page choose to compare your motif to those in the Human and Mouse (Jolma2013) database.

Questions

1. Which motifs were found to be more similar to your motif? _____

CONGRATULATIONS! You've made it to the end of the practical.

Hope you enjoyed it!

Don't hesitate to ask any questions and feel free to contact us any time (email addresses on the front page).

Bonus Exercise I

One of the most frequent questions that come up in ChIP-seq experiments is whether the sequencing depth is sufficient.

Questions

1. Can you think of any way that we can check if the sequencing depth is indeed sufficient? _____

The more we sequence a ChIP-seq library, the more peaks of low fold change we will identify. Therefore, the only way to answer that question is to look for the number of peaks id entified when we down sample our library.

To test for sufficient sequencing depth in our sample we will down sample our ChIP and Control datasets to 10%, 20%, .., 90% of the initial library size and call peaks. To do so, we will use the functions `randsample` and `callpeak` from `macs2`, respectively:

```
i=10
while [[ $i -lt 100 ]]
do
        echo "Looking at ${i}% of the reads"
        macs2 randsample -t PAX5.bam -p $i -o PAX5.perc${i}.bed --outdir
           macs2_downsample -f BAM
        macs2 randsample -t Control.bam -p $i -o Control.perc${i}.bed
           --outdir macs2_downsample -f BAM

        macs2 callpeak -t macs2_downsample/PAX5.perc${i}.bed -c
           macs2_downsample/Control.perc${i}.bed --gsize hs --format BED
           --name macs2_downsample/PAX5.perc${i} --pvalue 1e-3
           --call-summits
        ((i = i + 10))
done
```

Then launch *R* and type:

```
rm(list = ls())

options(stringsAsFactors=F)
setwd("~/Desktop/ChIP-seq")

fc.thres <- 4

no.peaks <- c()
for(row in seq(from=10, to = 90, by = 10))
{
        print(row)
        peaks <- read.table(paste("macs2_downsample/PAX5.perc", row,
           "_peaks.narrowPeak", sep=""))
        peaks <- peaks[peaks[, 7] > fc.thres, ]

        no.peaks <- c(no.peaks, nrow(peaks))
}

peaks <- read.table("PAX5_peaks.narrowPeak")
peaks <- peaks[peaks[, 7] > fc.thres, ]
no.peaks <- c(no.peaks, nrow(peaks))

plot(seq(from=10, to = 100, by = 10), no.peaks, , type="o", col="blue",
   xlab="Percentage of reads", ylab="Number of peaks")
```

Questions

1. Do you think that we have sequenced enough? _____