

# ChIP-Seq

December 15, 2020

## 1 Introduction

ChIP-Seq is the combination of chromatin immunoprecipitation (ChIP) assays with high-throughput sequencing (Seq) and can be used to identify DNA binding sites for transcription factors and other proteins. 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. Throughout this practical we will try to identify potential transcription factor binding sites of PAX5 in human lymphoblastoid cells.

### 1.1 Learning outcomes

By the end of this tutorial you can expect to be able to:

- generate an unspliced alignment by aligning raw sequencing data to the human genome using [Bowtie2](#)
- manipulate the SAM output in order to visualise the alignment in [IGV](#)
- based on the aligned reads, find immuno-enriched areas using the peak caller [MACS2](#)
- perform functional annotation and motif analysis on the predicted binding regions

### 1.2 Tutorial sections

This tutorial comprises the following sections:

1. [Introducing the tutorial dataset](#)
2. [Aligning the PAX5 sample to the genome](#)
3. [Manipulating SAM output](#)
4. [Visualising alignments in IGV](#)
5. [Aligning the control sample to the genome](#)
6. [Identifying enriched areas using MACS](#)
7. [File formats](#)
8. [Inspecting genomic regions using bedtools](#)
9. [Motif analysis](#)

### 1.3 Authors

This tutorial was converted into a Jupyter notebook by [Victoria Offord](#) based on materials developed by Angela Goncalves, Myrto Kostadima, Steven Wilder and Maria Xenophontos.

## 1.4 Running the commands from this tutorial

You can run the commands in this tutorial either directly from the Jupyter notebook (if using Jupyter), or by typing the commands in your terminal window.

```
[ ]: pwd
```

```
[ ]: ls -l
```

### 1.4.1 Running commands in the terminal

You can also follow this tutorial by typing all the commands you see into a terminal window. This is similar to the “Command Prompt” window on MS Windows systems, which allows the user to type DOS commands to manage files.

To get started, select the cell below with the mouse and then either press control and enter or choose Cell -> Run in the menu at the top of the page.

```
[ ]: echo cd $PWD
```

Open a new terminal on your computer and type the command that was output by the previous cell followed by the enter key. The command will look similar to this:

```
[ ]: cd /home/manager/course_data/chip_seq
```

Now you can follow the instructions in the tutorial from here.

## 1.5 Prerequisites

This tutorial assumes that you have the following software or packages and their dependencies installed on your computer. The software or packages used in this tutorial may be updated from time to time so, we have also given you the version which was used when writing the tutorial.

Package	Link for download/installation instructions	Version tested
bedtools	<a href="http://bedtools.readthedocs.io/en/latest/content/installation.html">http://bedtools.readthedocs.io/en/latest/content/installation.html</a>	2.25.0
Bowtie2	<a href="http://bowtie-bio.sourceforge.net/bowtie2">http://bowtie-bio.sourceforge.net/bowtie2</a>	2.2.6
IGV	<a href="http://software.broadinstitute.org/software/igv">http://software.broadinstitute.org/software/igv</a>	2.3.8
MACS2	<a href="https://github.com/taoliu/MACS">https://github.com/taoliu/MACS</a>	2.1.0.20150420
meme	<a href="http://meme-suite.org/tools/meme">http://meme-suite.org/tools/meme</a>	4.10.0
samtools	<a href="https://github.com/samtools/samtools">https://github.com/samtools/samtools</a>	1.6
tomtom	<a href="http://web.mit.edu/meme_v4.11.4/share/doc/tomtom.html">http://web.mit.edu/meme_v4.11.4/share/doc/tomtom.html</a>	4.10.0
UCSC tools	<a href="http://hgdownload.cse.ucsc.edu/admin/exe/linux.x86_64">http://hgdownload.cse.ucsc.edu/admin/exe/linux.x86_64</a>	NA

## 1.6 Let's get started!

To get started with the tutorial, head to the first section: [introducing the tutorial dataset](#).

The answers to all questions in the tutorial can be found in [answers.ipynb](#).

## 2 Visualising alignments in IGV

It is often instructive to look at your data in a genome browser. Here, we use [IGV](#), a stand-alone browser, which has the advantage of being installed locally and providing fast access. Please check their website (<http://www.broadinstitute.org/igv>) for all the formats that IGV can display.

Web-based genome browsers, like [Ensembl](#) or the [UCSC browser](#), are slower, but provide more functionality. They do 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. For our visualization purposes we will use the BAM and bigWig formats.

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 file 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.

**If you are not in there already, change into the data directory.**

```
[ ]: cd data
```

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

```
[ ]: samtools sort -T PAX5.temp.bam -o PAX5.sorted.bam PAX5.bam
```

`-T` option indicates the temporary file prefix name

**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 index file when running `samtools index`.

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 and the data will be displayed as a graph. 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 bedgraph, called PAX5.bedgraph, using the tool `genomeCoverageBed` from [bedtools](#).

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

```
[ ]: genomeCoverageBed
```

Apart from the BAM file, we also need to provide the size of the chromosomes for the organism of interest in order to generate the bedgraph 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 or genome build you are working with. 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).

**To obtain chromosome lengths for the human genome, type:**

```
[ ]: fetchChromSizes hg19 > genome/hg19.all.chrom.sizes
```

We next want to remove any chromosome length information for the patched chromosomes, which are accessioned scaffold sequences that represent assembly updates. That way we will only keep the information of the current assembly.

**Remove this information using awk:**

```
[ ]: awk '$1 !~ /[_.]/' genome/hg19.all.chrom.sizes > genome/hg19.chrom.sizes
```

**Now generate the bedgraph file, called PAX5.bedgraph, by typing:**

```
[ ]: genomeCoverageBed -bg -ibam PAX5.sorted.bam \
    -g genome/hg19.chrom.sizes > PAX5.bedgraph
```

We then need to convert the bedgraph into a binary graph, called PAX5.bw, using the tool `bedGraphToBigWig` from the UCSC tools.

**To convert the bedgraph type:**

```
[ ]: bedGraphToBigWig PAX5.bedgraph genome/hg19.chrom.sizes PAX5.bw
```

Now we will load the data into the IGV browser for visualisation.

**To launch IGV :**

```
[ ]: igv &
```

On the top left of your screen choose “Human hg19” from the drop down menu. Then in order to load the desired files go to “File → Load from File”.

On the pop up window navigate to the tutorial folder and select the file `PAX5.sorted.bam`.

Repeat these steps in order to load `PAX5.bw` as well.

Select “chr1” from the drop down menu on the top left.

Right click on the name of PAX5.bw and choose “Maximum” under the “Windowing Function”.

Right click again and select “Autoscale”.

---

## 2.1 Questions

**Q1.** Look for gene NASP in the search box. Can you see a PAX5 binding site near the NASP gene?

*Hint: use the “+” button on the top right zoom in more to see the details of the alignment*

**Q2.** What is the main difference between the visualisation of BAM and bigWig files?

---

## 2.2 What’s next?

You can head back to [manipulating SAM output](#) or continue on to [aligning the control sample to the genome](#).

# 3 Manipulating SAM output

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

To convert your SAM file to a BAM file, 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.

**If you are not in there already, change into the data directory.**

```
[ ]: cd data
```

**Convert SAM to BAM using `samtools` and store the output in the file `PAX5.bam`:**

```
[ ]: samtools view -bSo PAX5.bam PAX5.sam
```

---

## 3.1 What’s next?

For a quick recap of what the tutorial covers head back to the [introduction](#).

If you want a reintroduction to the tutorial dataset, head back to [aligning the PAX5 sample to the genome](#).

Otherwise, let's continue on to [visualising alignments in IGV](#).

## 4 Visualising alignments in IGV

It is often instructive to look at your data in a genome browser. Here, we use [IGV](#), a stand-alone browser, which has the advantage of being installed locally and providing fast access. Please check their website (<http://www.broadinstitute.org/igv>) for all the formats that IGV can display.

Web-based genome browsers, like [Ensembl](#) or the [UCSC browser](#), are slower, but provide more functionality. They do 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. For our visualization purposes we will use the BAM and bigWig formats.

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 file 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.

**If you are not in there already, change into the data directory.**

```
[ ]: cd data
```

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

```
[ ]: samtools sort -T PAX5.temp.bam -o PAX5.sorted.bam PAX5.bam
```

`-T` option indicates the temporary file prefix name

**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 index file when running `samtools index`.

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 and the data will be displayed as a graph. 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 bedgraph, called `PAX5.bedgraph`, using the tool `genomeCoverageBed` from [bedtools](#).

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

```
[ ]: genomeCoverageBed
```

Apart from the BAM file, we also need to provide the size of the chromosomes for the organism of interest in order to generate the bedgraph 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 or genome build you are working with. 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).

**To obtain chromosome lengths for the human genome, type:**

```
[ ]: fetchChromSizes hg19 > genome/hg19.all.chrom.sizes
```

We next want to remove any chromosome length information for the patched chromosomes, which are accessioned scaffold sequences that represent assembly updates. That way we will only keep the information of the current assembly.

**Remove this information using awk:**

```
[ ]: awk '$1 !~ /[_.]/' genome/hg19.all.chrom.sizes > genome/hg19.chrom.sizes
```

**Now generate the bedgraph file, called PAX5.bedgraph, by typing:**

```
[ ]: genomeCoverageBed -bg -ibam PAX5.sorted.bam \  
-g genome/hg19.chrom.sizes > PAX5.bedgraph
```

We then need to convert the bedgraph into a binary graph, called PAX5.bw, using the tool bedGraphToBigWig from the UCSC tools.

**To convert the bedgraph type:**

```
[ ]: bedGraphToBigWig PAX5.bedgraph genome/hg19.chrom.sizes PAX5.bw
```

Now we will load the data into the IGV browser for visualisation.

**To launch IGV :**

```
[ ]: igv &
```

On the top left of your screen choose “Human hg19” from the drop down menu. Then in order to load the desired files go to “File → Load from File”.

On the pop up window navigate to the tutorial folder and select the file PAX5.sorted.bam.

Repeat these steps in order to load PAX5.bw as well.

Select “chr1” from the drop down menu on the top left.

Right click on the name of PAX5.bw and choose “Maximum” under the “Windowing Function”.

Right click again and select “Autoscale”.

## 4.1 Questions

**Q1.** Look for gene **NASP** in the search box. Can you see a **PAX5** binding site near the **NASP** gene?

*Hint: use the “+” button on the top right zoom in more to see the details of the alignment*

**Q2.** What is the main difference between the visualisation of **BAM** and **bigWig** files?

---

## 4.2 What’s next?

You can head back to [manipulating SAM output](#) or continue on to [aligning the control sample to the genome](#).

# 5 Aligning the control sample to the genome

In the ChIP-Seq folder you will find another **.fastq** file called **Control.fastq**.

If you are not in there already, change into the data directory.

```
[ ]: cd data
```

Use the head command to look at this file:

```
[ ]: zless Control.fastq.gz | head
```

Use the information on the **FASTQ** Wikipedia page ([http://en.wikipedia.org/wiki/FASTQ\\_format](http://en.wikipedia.org/wiki/FASTQ_format)) to determine the quality encoding this **FASTQ** file is using. Then, adapting your commands to the quality encoding where needed, follow the steps you used to align the **PAX5** sample to the genome and manipulate the **SAM** file in order to align the control reads to the human genome.

---

## 5.1 What’s next?

You can head back to [visualising alignments in IGV](#) or continue on to [identifying enriched areas using MACS](#).

# 6 Finding enriched areas using MACS

**MACS2** stands for **m**odel-based **a**nalysis 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.

**If you are not in there already, change into the data directory.**

```
[ ]: cd data
```

**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` to set the mappable genome size

(with the read length we have, 70% of the genome is a fair estimation)

`--call-summits` to detect all subpeaks in each enriched region and return their summits

`--pvalue` the P-value cutoff for peak detection.

**Now run macs using the following command:**

```
[ ]: macs2 callpeak -t PAX5.sorted.bam -c Control.sorted.bam \  
--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.

## 6.1 What's next?

You can head back to [aligning the control sample to the genome](#) or continue on to [file formats](#).

# 7 File Formats

## 7.1 BED files

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 chr1 and one on chr2.

chromosome	start	end
chr1	50	100
chr1	500	1000
chr2	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.

chromosome	start	end	description
chr1	0	1	I-am-the-first-position-on-chrom-1

Using the bed format documentation found at <http://genome.ucsc.edu/FAQ/FAQformat.html#format1> answer the following questions.

### 7.1.1 Questions

**Q1.** The simplest bed file contains just three columns (chromosome, start, end) and is often called **BED3** format. What extra columns does **BED6** contain?

*Hint: look for information about columns 4 to 6 in the documentation <http://genome.ucsc.edu/FAQ/FAQformat.html#format1>*

**Q2.** In the above examples, what are the lengths of the intervals?

**Q3.** Can you output a **BED6** 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?

*Hint: you will need one line per exon*

---

## 7.2 narrowPeak files

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.

**If you are not in there already, change into the data directory.**

```
[ ]: cd data
```

**View the first 10 lines in PAX5\_peaks.narrowPeak using the head command:**

```
[ ]: head -10 PAX5_peaks.narrowPeak
```

NarrowPeak files can also be uploaded to **IGV** or other genome browsers.

**Try uploading the peak file generated by MACS2 to IGV.**

### 7.2.1 Questions

**Q4. 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*

**Q5. Does the first peak that was called look convincing to you?**

---

## 7.3 GTF files

A second popular format is the **GTF** format. Each row in a GTF formatted file denotes a genomic interval. The GTF format documentation can be found at <http://mblab.wustl.edu/GTF2.html>.

The three intervals from above might be:

seqid	source	type	start	stop	score	strand	phase	attributes
chr1	gene	exon	51	100	.	+	0	gene_id "001";transcript_id "001.1";
chr1	gene	exon	501	1000	.	+	2	gene_id "001";transcript_id "001.1";
chr2	repeat	exon	601	800	.	+	.	

The 9th column permits intervals to be grouped and linked in a hierarchical fashion. This format is thus popular to describe gene models. Note how the first two intervals are linked through a common transcript\_id and gene\_id.

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 GENCODE v18 geneset is available in the genome folder.

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

```
[ ]: zcat genome/gencode.v18.annotation.gtf | head -n10
     zless -S genome/gencode.v18.annotation.gtf
```

---

### 7.3.1 Questions

**Q6.** In the small example table above, why have the coordinates changed from the BED description?

---

## 7.4 What's next?

You can head back to [identifying enriched areas using MACS](#) or continue on to [inspecting genomic regions using bedtools](#).

## 8 Inspecting 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.

**If you are not in there already, change into the data directory.**

```
[ ]: cd data
```

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/hg19.chrom.sizes
```

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:

```
[ ]: gunzip genome/gencode.v18.annotation.gtf.gz

awk '$3=="gene"' genome/gencode.v18.annotation.gtf \
> genome/gencode.v18.annotation.genes.gtf
```

Next, count the total number of PAX5 peaks:

```
[ ]: wc -l PAX5_peaks.narrowPeak
```

Then use `bedtools` to find the number overlapping GENCODE genes:

```
[ ]: bedtools intersect -a PAX5_peaks.narrowPeak \
-b genome/gencode.v18.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.v18.annotation.genes.gtf | head
```

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.v18.annotation.genes.gtf > genome/gencode.tss.bed
```

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

```
[ ]: sortBed -i genome/gencode.tss.bed > genome/gencode.tss.sorted.bed
```

```
[ ]: bedtools closest -a PAX5_peaks.narrowPeak \
-b genome/gencode.tss.sorted.bed > PAX5_closestTSS.txt
```

Use `head` to inspect the results:

```
[ ]: head PAX5_closestTSS.txt
```

You have now matched up all the PAX5 transcription factor peaks to their nearest gene transcription start site.

---

## 8.1 Questions

**Q1.** Looking at the output of the `bedtools genomecov` we ran, what percentage of chromosome 1 do the peaks of PAX5 cover?

**Q2.** Looking at the output from `bedtools intersect`, what proportion of PAX5 peaks overlap genes?

**Q3.** Looking at `PAX5_closestTSS.txt`, which gene was found to be closest to MACS peak 2?

---

## 8.2 What's next?

You can head back to [file formats](#) or continue on to [motif analysis](#).

# 9 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. 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.

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 Oct4 motif to the genome regions around the summits of the 300 most significant PAX5 subpeaks on Chromosome 1.

**If you are not in there already, change into the data directory.**

```
[ ]: cd data
```

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.

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`.

```
[ ]: gunzip genome/HS19.fa.gz

samtools faidx genome/HS19.fa

bedtools getfasta -fi genome/HS19.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.**

Your MEME analysis will now be queued and will run on a server in the US. The results page will refresh automatically and once the tool has finished running there will be a link to the results. Depending on how busy the servers are your analysis may take a longer or shorter time to run.

You can check the load of the server here:

<http://meme-suite.org/opal2/dashboard?command=statistics>

## 9.1 Analyse the results from MEME

We would like to know if this motif is similar to any other known motif. We will use the results from [TOMTOM](#) for this.

On either the results from the web MEME run or the local run please follow the link “MEME html output”. Scroll down until you see the first motif logo.

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 JASPAR CORE and UniPROBE Mouse database.

## 9.2 Running MEME locally

If you want to speed things up you may want to run MEME on your own machine. You can try to do this as well if you wish, or skip the following bonus exercise and go to the next section.

To bring up the help page for the local installation of MEME, type:

```
[ ]: meme
```

Run MEME locally, setting the output directory with the option `-o` (e.g. `-o meme_out`).

```
[ ]: meme PAX5_top300_summits.fa -o meme_out -dna -nmotifs 1 -minw 6 -maxw 20
```

Once MEME has finished running look in this directory for the file `meme.html` and open it in a web browser. You can do this by either copying the path to the file to the address bar in Firefox or double click on the `.html` file.

Alternatively, you can run the following command to automatically open the HTML file in Firefox:

```
[ ]: firefox meme_out/meme.html
```

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](#) and a set of known motif databases stored in `motif_databases` for this.

To compare your newly found motifs to the motif databases JASPAR CORE and UniPROBE Mouse you can run:

```
[ ]: tomtom -o tomtom_out meme_out/meme.html \  
motif_databases/JASPAR/JASPAR_CORE_2016_vertbrates.meme \  
motif_databases/MOUSE/uniprobe_mouse.meme
```

Once again, once TOMTOM has finished running look in `tomtom_out` for the file `tomtom.html`.

Open `tomtom.html` in a web browser.

```
[ ]: firefox tomtom_out/tomtom.html
```



### 9.3 Questions

Q1. Which motif was found to be the most similar to your motif?

---

### 9.4 Congratulations, you have reached the end of this tutorial!

We hope you've enjoyed our ChIP-Seq tutorial. You can find the answers to all of the questions in this tutorial in [answers.ipynb](#). You can revisit [inspecting genomic regions using bedtools](#) or, go back to the [beginning](#).