ATAC-seq Data Analysis

Denis Seyres < *ds777@medschl.cam.ac.uk* >

ATAC-Seq analysis
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

Samtools: http://samtools.sourceforge.net/

IGV genome browser: http://www.broadinstitute.org/igv/

Ensembl genome browser: http://www.ensembl.org/

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

FSEQ: https://github.com/aboyle/F-seq

DeepTools: http://deeptools.readthedocs.io/en/latest/index.html

Bedtools: http://bedtools.readthedocs.io/en/latest/

Intervene: https://github.com/asntech/intervene

NucleoATAC: https://github.com/GreenleafLab/NucleoATAC

Original data can be found here: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE68983

Introduction

The goal of this hands-on session is to perform some basic tasks in the analysis of ATAC-seq data. After a classical QC and trimming (if needed!) step performed with *FastQC*, the second step includes a short read alignment from raw reads. We will align raw sequencing data to the drosophila genome (dm3) using *Bowtie* (v1.1.1), remove duplicates and low quality reads with *Picard MarkDuplicates* and *samtools* tools and then we will use the BAM output in order to 1) visualise the alignment on the *IGV* browser, 2) find enriched regions using the peak callers *MACS2* (v2.1.0) and *FSEQ* (v1.85), 3) annotate peaks with *PAVIS* and 4) identify nucleosomes positions and nucleosome free regions. Finally, we will ask you to 6) load on *IGV* different tracks (ChIP-seq histones, ChIP-seq TFs, ATAC-seq) and let you walk on the genome and see how these layers combine to each other.

Although the former experiments had 2 technical replicates, we will only be using one replicate for simplicity. We have restricted the reads to chromosome 2L for the purpose of that course.

Prepare the data

We will use one data set in this practical, which can be found in the ATAC-seq directory on your desktop. These data have been published by Koenecke et al. (2016). They are two biological replicates sequenced in paired-end mode. Throughout this practical we will try to identify open chromatin regions on left arm of the chromosome 2 in one biological replicate.

Open the Terminal.

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

```
cd ~/Desktop/ATAC-seq/
cd FASTQ/
```

The .fastq files that we will align are called ATAC_R1_1_trimmed_chr2L.fastq.gz and ATAC_R1_2_trimmed_chr2L.fastq.gz.

Quality check

Before aligning reads to the drosophila genome, we need to check if the sequenced reads meet usual QCs requirements. We have already ran the following commands on the whole genome, un-trimmed reads:

```
cd ~/Desktop/ATAC-seq/
fastqc FASTQ/ATAC_R1_1.fastq.gz --outdir=QC/
fastqc FASTQ/ATAC_R1_2.fastq.gz --outdir=QC/
```

You can go now in the *QC/* directory and open the `.html` files to see the results of FastQC analysis.

Read Trimming

Questions

Use the FastQC analysis to answer these questions.

1. Why do we need to trim reads? Have a look to the adapter contamination page._____
2. Which kind of trimming could we apply? _____

Recommended: Adapter Trimming

We can use the *trim_adapters* to trim the reads (http://atactk.readthedocs.io/en/latest/usage.html). The nice thing about this technique is that you don't need to know which adapter sequences were used.
To save time, we have already trimmed the reads.

We used the following command (you don't need to re-run them):

```
trim_adapters -h
trim_adapters FASTQ/ATAC_R1_1.fastq.gz
trim_adapters FASTQ/ATAC_R1_2.fastq.gz
```

Alternative: Fixed Length Trimming

We can also use the **fastx_trimmer** from the fastx-toolkit. This solution works also well and has been used in the original paper where the data come from (they trimmed all reads after 25bp). Nevertheless, mapping efficiency compared to the adapter trimming is generally lower.

In case you want to try by your-self, here are the commands (you don't have to run them):

```
fastx_trimmer -h
gunzip -c FASTQ/ATAC_R1_1.fastq.gz | fastx_trimmer -f 1 -l 25 -Q 33
    -o FASTQ/ATAC_R1_1_trimmed.fastq.gz -i -
gunzip -c FASTQ/ATAC_R1_2.fastq.gz | fastx_trimmer -f 1 -l 25 -Q 33
    -o FASTQ/ATAC_R1_2_trimmed.fastq.gz -i -
```

Alignment

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

Bowtie has a number of parameters in order to perform the alignment.

To view them all type

```
bowtie --help
```

Bowtie 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 drosophila genome (chr2L). For this we need the DNA sequence in fasta format. First, download the DNA sequence for that chromosome in the directory 'bowtie_index' :

```
cd ~/Desktop/ATAC-seq/
mkdir bowtie_index
wget ftp://ftp.flybase.net/genomes/Drosophila_melanogaster\
/dmel_r5.57_FB2014_03/fasta/dmel-2L-chromosome-r5.57.fasta.gz \
-P bowtie_index/
```

You have to uncompressed the file.

```
gunzip bowtie_index/dmel-2L-chromosome-r5.57.fasta.gz
```

The indexed chromosome (or genome) is generated using the command:

```
bowtie-build bowtie_index/dmel-2L-chromosome-r5.57.fasta \
bowtie_index/dm3
```

This command will output 6 files with the prefix dm3 that constitute the index.

To view if the files have been successfully created type:

```
ls -l bowtie_index
```

In general before you run *Bowtie*, you have to know which fastq format you have. The available fastq formats in bowtie are:

```
--phred33-quals input quals are Phred+33 (default, same as
   --solexa-quals)
--phred64-quals input quals are Phred+64 (same as solexa1.3-quals)
--solexa-quals input quals are from GA Pipeline version < 1.3
--solexa1.3-quals input quals are from GA Pipeline version >= 1.3
--integer-quals qualities are given as space-separated integers (not
   ASCII)
```

The fastq files format we are working on is of Sanger format (Phred+33), which is the default for *Bowtie*.

Now that the genome is indexed we can move on to the actual alignment. The first argument make sure that Bowtie reports only uniquely mapped reads using the `-m 1` option, the second argument allows for at most two mismatches in the alignment using the `-v 2`, the third argument limit the fragment size to 1000bp by using the `-X 1000` option, the fourth argument `-p 2` the number of processors you want to use and the fifth argument `-S` makes sure that Bowtie output format is SAM. The following argument for bowtie is the basename of the index for the genome to be searched; in our case it is `dm3`. The last argument is the name of the fastq file. Note that we uncompress the fastq file 'on the fly' to avoid writting intermediate

files on the disk. We pipe the *Bowtie* command with a *samtools* command to transform the output SAM into BAM files in order occupy much less space. SAM files are rather big and when dealing with a high volume of NGS data, storage space can become an issue.

Bowtie will take approximatively 10 minutes to align the file with 2 processors. This is fast compared to other aligners that sacrifice some speed to obtain higher sensitivity.

```
cd ~/Desktop/ATAC-seq/
mkdir BAM
bowtie -m 1 -v 2 -X 1000 -p 2 -S bowtie_index/dm3 \
-1 <( gunzip -c FASTQ/ATAC_R1_1_trimmed_chr2L.fastq.gz ) \
-2 <( gunzip -c FASTQ/ATAC_R1_2_trimmed_chr2L.fastq.gz ) \
| samtools view -bS - > BAM/ATAC_R1_trimmed.bam
```

The above commands output the alignment in BAM format and store them in the file `ATAC_R1.bam`. `-@ 4` allows 4 processors to the tool. Some tools could require that the BAM file is sorted and indexed. Sorting BAM files also reduces the file size and help you saving disk space.

```
samtools sort -@ 4 -o BAM/ATAC_R1_trimmed.sorted.bam \
BAM/ATAC_R1_trimmed.bam
```

Index the sorted files.

```
samtools index BAM/ATAC_R1_trimmed.sorted.bam
```

Remove duplicated and low quality reads

We will now remove low quality reads and PCR duplicates. The idea is to first mark reads identified as PCR duplicates and then remove them. You could also just have Picard remove them, if you have no need for them later in the pipeline.

First, mark PCR duplicates:

```
java -jar /applications/local/picard-tools/picard-tools/picard.jar \
MarkDuplicates INPUT=BAM/ATAC_R1_trimmed.sorted.bam \
OUTPUT=BAM/ATAC_R1.markDup.sorted.bam \
METRICS_FILE=BAM/ATAC_R1.markDup.txt REMOVE_DUPLICATES=false \
ASSUME_SORTED=true VALIDATION_STRINGENCY=LENIENT TMP_DIR=./
```

The output will be a BAM file containing all of Bowtie's output, with duplicate reads marked.

Then we apply some filters to remove unwanted reads:

```
samtools view -b -h -f 3 -F 4 -F 8 -F 256 -F 1024 -F 2048 -q 15 \
BAM/ATAC_R1.markDup.sorted.bam | \
samtools sort -o BAM/ATAC_R1.q15.rmdup.sorted.bam -
```

The arguments:

-b: requests BAM output
-h: requests that the header from the input BAM file be included.

The next few use SAM flags to filter alignments. There's a detailed specification for SAM files, which describes the flags that tools can use to annotate aligned reads.

-f 3: only include alignments marked with the SAM flag 3, which means "properly paired and mapped"
-F 4: exclude aligned reads with flag 4: the read itself did not map
-F 8: exclude aligned reads with flag 8: their mates did not map
-F 256: exclude alignments with flag 256, which means that Bowtie mapped the read to multiple places in the reference genome, and this alignment is not the best
-F 1024: exclude alignments marked with SAM flag 1024, which indicates that the read is an optical or PCR duplicate (this flag would be set by Picard)
-F 2048: exclude alignments marked with SAM flag 2048, indicating chimeric alignments, where Bowtie decided that parts of the read mapped to different regions in the genome. These records are the individual aligned segments of the read. They usually indicate structural variation. We're not going to base peak calls on them.

Finally, we use a basic quality filter, -q 15, to request high-quality alignments.

Index the sorted files.

```
samtools index BAM/ATAC_R1.q15.rmdup.sorted.bam
```

By using the *samtools flagstat* command with the appropriates BAM files and by opening the Picard markDuplicates log file, answer the following questions:

Questions

What is the number of:
1. total reads? _____
2. unmapped reads? _____
3. PCR duplicates? _____
4. final number of mapped reads? of fragments (valid pairs of reads)? _____

Visualise alignments in IGV

It is always instructive to look at your data in a genome browser. Here, we use IGV, a standalone browser, which has the advantage of being installed locally and providing fast access. 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.

Please check IGV website http://www.broadinstitute.org/igv/ for all the formats that IGV can display. For our visualisation purposes we will use bigWig format and, later, BED files containing ATAC-seq peaks coordinates.

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. They are more lighter than BAM files.

The BAM to bigWig conversion takes place in one step. We will use the Deeptools bamCoverage tool. We ask for a bin size of 1bp (highest resolution) and a *bigwig* conversion format. Convert the uniquely mapped reads BAM file to bigwig.

```
mkdir ~/Desktop/ATAC-seq/BW
cd ~/Desktop/ATAC-seq/
bamCoverage -b BAM/ATAC_R1.q15.rmdup.sorted.bam -o BW/ATAC_R1.bw \
-of bigwig --binSize 1
```

Now we will load the data into the IGV browser for visualization. In order to launch IGV type the following on your terminal:

```
igv &
```

On the top left of your screen choose from the drop down menu **D. melanogaster (dm3)**. Then in order to load the desire files go to:

File –> Load from File

On the pop up window navigate to **Desktop > ATAC-seq > BW** folder and select your newly created bigwig files.

Remenber we aligned reads only on chromosome 2L so select chr2L from the drop down menu on the top left. Select all tracks, right click on their name and choose Maximum under the Windowing Function. Right click again and select Autoscale.

Finding enriched areas using MACS2

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
--format (-f) the tag file format. If this option is not set MACS
   automatically detects which format the file is.
--name (-n) to set the name of the output files
--gsize (-g) This is the mappable genome size. With the read length
   we have, 90% of the genome is a fair estimation. Since in this
   analysis we include only reads from chromosome 2L, we will use
   as gsize 90% of the length of chromosome 2L (23 Mb). MACS also
   offers shortcuts for human, 'mm' for mouse 'ce' for C. elegans
   and 'dm' for fruitfly.
--qvalue Minimum FDR (q-value) cutoff for peak detection.
--broad: request that adjacent enriched regions be combined into
   broad regions
```

Now run `macs2` using the following command and call peaks:

```
mkdir ~/Desktop/ATAC-seq/peakCalling
mkdir ~/Desktop/ATAC-seq/peakCalling/MACS2
cd ~/Desktop/ATAC-seq/
macs2 callpeak -t BAM/ATAC_R1.q15.rmdup.sorted.bam -g 2.1e7 \
-n ATAC_R1 -f BAMPE --broad \
--outdir peakCalling/MACS2 --qvalue 0.05
```

MACS2 generates its peak files in a file format called .broadPeak file. This is a simple text format containing genomic locations, specified by chromosome, begin and end positions, and information on the statistical significance of each called peak.

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

Upload the peak files generated by MACS to IGV.

Questions

1. MACS2 estimates a fragment length based on data. What is the fragment length of your data? _____
2. Compare it to the size of nucleosome DNA fragment. _____
3. What do you think of the called peaks? _____

Annotation: From peaks to biological interpretation

In order to biologically interpret the results of ATAC-seq experiments, it is usually recommended to look at the genes and other annotated elements that are located in proximity to the identified enriched regions. This can be easily done using PAVIS http://manticore.niehs.nih.gov/pavis2/. PAVIS requires the files in BED format, which is a tab-delimited file that contains information on chromosome, start and end position for each region. See http://genome.ucsc.edu/FAQ/FAQformat.html#format1 for details.

To convert MACS2 `.broadPeak` files to BED file, select the first 4 columns:

```
cut -f1-4 peakCalling/MACS2/ATAC_R1_peaks.broadPeak \
> peakCalling/MACS2/ATAC_R1.bed
```

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

In the **Species/Genome Assembly/Gene Set** dropdown menu select **D.Melanogaster flybase R5.57/dm3**.

Fill in the location of the BED peak file, 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 are most likely opened.

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.

Identification of nucleosomes positions

The "bead" or nucleosome is the smallest packaging unit of the chromatin fiber and it consists of 147bp DNA wrapped 1.65 turns around a histone octamer core, consisting of two H2A/H2B and H3/H4 heterodimers(Luger et al., 1997). Nucleosomes are then further assembled into various higher order structures that can potentially be unwrapped as needed (for review see (Luger and Hansen, 2005)). Nucleosome "calling" or positionning analysis is the next level of analysis and will greatly help in accurate identification of nucleosome-depleted regions.

The required files are: 1. BAM files filtered for low quality and duplicated reads 2. FASTA file of the genome (here we only use chromosome 2L sequence). It needs to be indexed (using `samtools faidx`) 3. Sorted BED file with regions for which nucleosome analysis is to be performed. It is potentially advisable to extend these regions a bit (e.g. using `bedtools slop`). Regions should not overlap so it is advisable to use `bedtools merge` on these regions.

Prepare the BED file

```
mysql --user=genome --host=genome-mysql.cse.ucsc.edu \
-A -e "select chrom,size from dm3.chromInfo" > dm3.genome

bedtools slop -b 500 -i peakCalling/MACS2/ATAC_R1.bed \
-g dm3.genome > peakCalling/MACS2/ATAC_R1_sloped.bed

bedtools merge -i peakCalling/MACS2/ATAC_R1_sloped.bed \
> peakCalling/MACS2/ATAC_R1_sloped_merged.bed
```

Then we will run the nucleoATAC command on the first 100 peaks (for time considerations).

```
head -n 50 peakCalling/MACS2/ATAC_R1_sloped_merged.bed > \
peakCalling/MACS2/ATAC_R1_sloped_merged_top50.bed
```

```
mkdir ~/Desktop/ATAC-seq/nucleoATAC/
nucleoatac run \
--bed peakCalling/MACS2/ATAC_R1_sloped_merged_top50.bed \
--bam BAM/ATAC_R1.q15.rmdup.sorted.bam \
--fasta bowtie_index/dmel-2L-chromosome-r5.57.fasta \
--out nucleoATAC/ATAC_R1 --cores 4
```

Among the output files, you can find *output_ATAC_R1.nuc_dist.eps* showing a plot with estimate of fragment size at nucleosomes, *output_ATAC_R1.occ_fit.eps* showing a plot of model for NFR (Nucleosome free Regions) and nucleosomal distributions. You can also find a serie of *ATAC_R1.nfrpos.bed.gz* file which gives you the location of nucleosome free regions, *ATAC_R1.nucpos.bed.gz* which gives you the position of the nucleosomes and *ATAC_R1.nucleoatac_signal.smooth.bedgraph* which is signal file.

Upload them into IGV together with the ATAC bigwig files you have produced from BAM files and the peaks called with MACS2. We also advise you to load H3K27ac you have previously created during the ChIP-seq practical and Nejire TF located on 'other_data' which is coming from the paper.

Questions

1. Does the estimated fragment size distribution fits with the -X 1000 option we used for MACS2? _____
2. What do you observe in IGV when you compare all together MACS2 peaks, NFR peaks, nucleosome positions, H3K27ac, Nejire TF? _____

The regions where ATAC peaks are detected are open chromatin regions, where you always have a mixture of signals, both NFR, mono, and maybe di-or trinucleosomes. Inferring these features requires more specialized software, such as NucleoATAC. The peakcalling gives you a region where transposase cutting events are enriched and in which "you can dig deeper".

Analysis centered around annotations

You have seen until now how to perform peak-centered analyses of ATAC-seq data. However, sometimes it is also of interest to analyse ATAC-seq profiles around annotations (Transcription start sites, genes, etc…), without performing peak calling. This is often also the case when analysing histone modifications, in order to investigate the chromatin environment around gene features and regulatory elements.

We will firstly visualize how the ATAC-seq signal fits to the H3K27ac signal on ATAC-seq peaks and on gene promoters. You will need to start an R session and launch seqplots.

```
R
library(seqplots)
run()
```

It opens a page in your web-browser.

You need to upload files. Click on 'Add files' and 'Add files…'. Select the different bigwig files (ATAC-seq,H3K27ac,Nejire) and the different feature files (ATAC-seq peaks, gene TSS (in *other_data* directory), H3K27ac peaks). Select the right genome and 'Start upload'.

Close the window and click on 'New plot set'. Under the tab 'Tracks' select all the bigwig files. Under the 'Features' track, select all the feature files. At the bottom of the page, select 'Midpoint features' in plot type (to center the signal on the feature, e.g. the promoter regions). Then, 'Run calculation'. Once finished, select the tracks you want to display in the table and click 'Profile' in the left panel.

Questions

1. You can now view the different aggregation plots and heatmaps in your folder. Where does the majority of the different histone mark, ATAC signal and TFs appear around TSS? on ATAC-seq peaks? on ChIP-seq peaks? _____
2. Did you expected these patterns? _____

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 Exercises

You first need to process (from the mapping step) the second replicate ATAC_R2 by following all the steps you followed for the first replicate ATAC_R1.

Bonus Exercise I: Check correlation between replicates

This step is an important step in the data pre-processing. You need to be sure that your replicates (either biological or technical) are replicates. A quick way is to compute the overall similarity between them based on read coverage. You can use either bigwig or BAM files. Here is an example with bw files.

Go to the appropriate directory:

```
cd ~/Desktop/ATAC-seq
```

We compute the average scores for each of the files for the entire genome by running the program in bins mode, but you can specify regions of interest (BED-file mode).

```
multiBigwigSummary bins -b BW/ATAC_R1.bw BW/ATAC_R2.bw -o
    BW/results.npz
```

Finally we can estimate the correlation and draw it with a scatter plot.

```
plotCorrelation -in BW/results.npz --corMethod pearson --skipZeros \
--plotTitle "Pearson Correlation of Average Scores ATAC-seq" \
--whatToPlot scatterplot \
-o BW/scatterplot_PearsonCorr_bigwigScores_ATACseq.png
```

Bonus Exercise II: Alternative peak caller

It is often recommended to call peaks with an other peak caller and then compare both lists of peaks. You expect high concordance between tools. Alternatively to MACS2, one can use Fseq. It stands for 'Feature Density Estimator for High-Throughput Sequence Tags'.
Prepare the output directories:

```
cd ~/Desktop/ATAC-seq/
mkdir ~/Desktop/ATAC-seq/peakCalling/FSEQ
mkdir ~/Desktop/ATAC-seq/peakCalling/FSEQ/ATAC_R1
```

FSEQ takes a BED file as input. So you need to transform the BAM files into BED file.

```
bamToBed -i BAM/ATAC_R1.q15.rmdup.sorted.bam > \
peakCalling/FSEQ/ATAC_R1.bam.bed
```

Then run *FSEQ*.

```
fseq -h
fseq -o peakCalling/FSEQ/ATAC_R1 -of bed \
-t 6 peakCalling/FSEQ/ATAC_R1.bam.bed
```

FSEQ produces one peak file for each chromosome. Here we don't need to concatenate files because we are working with only one chromosome. but we advise you to concatenate them all when you will analyze complete datasets.

```
cat peakCalling/FSEQ/ATAC_R1/*bed > \
peakCalling/FSEQ/ATAC_R1_fseq.bed
```

Repeat these commands for the second replicate.

Upload the peak files generated by FSEQ into IGV.

You can plot the size distribution of the reads. We will use R. In the terminal, type:

```
R
macs2=read.table('peakCalling/MACS2/ATAC_R1_peaks.broadPeak',sep="\t",header=F)
fseq=read.table('peakCalling/FSEQ/ATAC_R1_fseq.bed',sep="\t",header=F)
fseq$size=fseq$V3-fseq$V2
fragmentLength=175
pdf('~/Desktop/ATAC-seq/peakCalling/peakSizeDistribution_replicate1.pdf')
par(mfrow=c(1,1))
hist(macs2$V5,main="MACS2 peak size
   distribution",xlim=c(0,1000),breaks=1000)
abline(v=fragmentLength,col="red")
hist(fseq$size,main="Fseq peak size
   distribution",xlim=c(0,1000),breaks=1000)
abline(v=fragmentLength,col="red")
dev.off()
```

Press Ctrl+D to close *R*.

Bonus Exercise III: Compare set of peaks

You can now run comparisons within replicates and between *FSEQ* and *MACS2*. We will use intervene to summarize all comparisons. We will first draw a Venn diagram. When the number of sets exceeds three or four, Venn diagrams become difficult to read and interpret. An alternative and more effective approach is to use UpSet plots to visualize the intersections.

```
intervene venn --names=fseq_R1,fseq_R2,MACS2_R1,MACS2_R2 \
-i peakCalling/FSEQ/ATAC_R1_fseq.bed \
peakCalling/FSEQ/ATAC_R2_fseq.bed \
peakCalling/MACS2/ATAC_R1.bed peakCalling/MACS2/ATAC_R2.bed \
-o peakCalling
intervene upset --names=fseq_R1,fseq_R2,MACS2_R1,MACS2_R2 \
-i peakCalling/FSEQ/ATAC_R1_fseq.bed \
peakCalling/FSEQ/ATAC_R2_fseq.bed \
peakCalling/MACS2/ATAC_R1.bed peakCalling/MACS2/ATAC_R2.bed \
-o peakCalling
```

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

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).