Epigenetics Practical Session

Alexey Larionov – December 2024

Contents

Epigenetics data analysis is a broad field with many bioinformatics tools written in different languages (CPP, Python, R etc), for different data types (methylation, chromatin accessibility, TF binding, Hi-C, etc) and for different users (command-line and GUI tools). In this practical session you will study an example of the ATAC-seq data analysis pipeline.

Given the timeframe allocated for epigenomics within the course, you are not expected to write epigenomics pipelines yourself. Instead, you will be given the key elements of the code. Your task will be to understand the pipeline, modify the code so it will run on your account, and run the analysis on the Crescent2 cluster. Finally, you should review and interpret the results from a biological perspective.

Aims of this practical session

- Assemble a multi-step bioinformatic pipeline that uses a wide range of tools, resources, and produces multiple output files.
- Gain experience with ATAC-seq data analysis.

Computational environment

It is expected that you have already learned how to use the Crescent2 HPC and MobaXterm during the previous modules. The Crescent2 HPC has modules for all the tools required for this practical session. One of the tools (HOMER) will be provided as a Singularity container. The source data and resources required for the analysis are available in a shared area of Crescent cluster. You should copy the source data and resources (including the container with HOMER) to ***your own*** Crescent folder and use ***your own*** copies for your pipeline.

Organising your workspace

Before starting the pipeline, you must think about organising your workspace.

During this practical session, you will use multiple scripts, and generate tens of output files, logs and plots. It is therefore essential to properly organise your working space to facilitate such analysis. You should start by making the ***main project folder*** that will contain ***several sub-folders*** for all your project-related files.

Where should you put the main project folder? I suggest putting it into your Crescent2 home folder.

How should you name the project folder? I suggest "epigenetics":

```
…/<user>/epigenetics
```

After creating the main project folder, make five sub-folders in it: for data, resources, scripts, results, and Singularity containers:

Subfolder name	**Purpose**
data	Will be used to copy the source data to
resources	Will be used to keep resources (the Bowtie2 index, file with blacklisted regions, etc)
scripts	Will be used to keep the pipeline scripts and logs
results	Will be used to save results to, will contain many sub-folders for each step of analysis
containers	Will be used for the HOMER container

Of course, you may organise your workspace in a different manner. However, you MUST decide how to organise your working space before you start downloading data or writing/running the scripts.

Important: do NOT use spaces in your folders and file names! Instead consider using underscores (_)

Provided data and resources

During this practical you will analyse data from the study of Dr. Dan Hasson with co-authors: https://doi.org/10.1016/j.celrep.2022.110637 (PDF provided). Of course, it would not be possible to reproduce all the analysis from this paper during our practical session. So, you will only analyse a small sub-set of data and run only a simplified fragment of the original ATAC-seq pipeline.

The source data and other resources for this practical are available in a shared folder on Crescent2 located:

```
/mnt/beegfs/project/Alexey_Larionov/IBIX-PRO-24/epigenetics
```

The provided files and folders are:

Location	Files or folders	Description
.../data/atac_seq	samples.txt ARID2ko_ chr5_R1.fastq ARID2ko _chr5_R2.fastq ARID2wt _chr5_R1.fastq ARID2wt _chr5_R2.fastq	File with samples description, ATAC-seq raw data for the main pipeline.
.../data/chip_seq	ARID2_chr5.bw FOSL2_chr5.bw H3K27ac_chr5.bw H3K4me3_chr5.bw	BigWig files from Chip-seq to compare with ATAC-seq
.../data/rna_seq	Deseq_ARID2Ko_v_ARID2wt_chr5.csv	DEGs to compare with ATAC-seq
.../resources	bowtie2_index/... hg38-blacklist.bed	Index files for alignment, Blacklisted ATAC-seq regions
.../containers	homer_v4.11_hg38.sif	Singularity container with HOMER toolset and resources for motif enrichment analysis

Copy the provided materials to your working folder before starting the analysis. Make sure that you have at least 10GB of disk space in your account before the analysis (e.g. by using `myquota` command). You may copy the data using ***cp*** command, executing it directly on login nodes.

Pipeline summary

The pipeline includes a minimal set of steps that would be used in virtually any ATAC-seq pipeline:

- QC (quality control) and trimming of the source FASTQ files
- Alignment to human reference genome with Bowtie 2
- Filtering and deduplication of BAM files
- Visualising peaks coverage in IGV and Heatmaps (using BigWig files)
- Calling peaks with MACS2, filtering out the blacklisted regions
- Differential peaks analysis (simplified, performed in R)
- Detection of the genes associated with the differential peaks
- Detection of the motifs enriched in the differential peaks
- Reviewing the results of ATAC-seq analysis in connection with ChIP-seq and RNA-seq data

All these tasks have been discussed during the lectures.

You are given a set of scripts (on CANVAS). You should review each script, modify if necessary, and run on the Crescent2 cluster. The output of each script should be placed into a separate sub-folder in your **results** folder. After running each script, you should review and understand the results. You should work with the scripts in the given order as each script requires the output of the previous steps.

For clarity, all scripts are provided without the additional header and footer which are required for running scripts on Crescent2. An example of the header and footer required for submitting batch jobs to Crescent2 is provided separately in the ***crescent2_batch_job.sh*** file.

Important: the scripts should NOT be run directly in login nodes!

You should always execute scripts on compute nodes using ***qsub*** to submit the script as a job.

This handout will provide brief information about each step, in addition to the information provided in the lectures. Each script will be discussed in the class.

Step 1: QC and trimming of FASTQ files

It is assumed that you have already organised the folders structure, and copied the source data and resources as described previously. Review ***s01_qc_and_preprocessing.sh*** script. Add the necessary header and footer to run it on Crescent2. Remember to change the path to ***your*** base folder, the e-mail and the job name in the header. Check that the number of requested cores is appropriate for the multi-threading options used in the script. After updating the script, submit it to the Crescent2 queue:

```
qsub s01_qc_and_preprocessing.sh
```

You should receive the Crescent2 job progress updates by e-mail, and you may check your jobs on cluster by using the **myjobsum** command. You may terminate your jobs with the **qdel** command.

Once your script has completed you should get the log file generated by your script in the ***scripts*** folder, and trimmed FASTQ files in the dedicated sub-folder in your ***results*** folder, as well as FastQC and MultiQC results. The script execution should take **about 5 min**.

Review the FastQC and MultiQC results (you should already be familiar with FastQC and MultiQC output after the previous modules). You may copy the output QC files to your laptop, or review them directly on cluster. To view the HTML files directly on cluster you may use **RightClick - Open with default program** option. If you are satisfied with the quality of data at this step, continue to the next step: alignment of the reads to the reference genome.

Step 2: Alignment

Review and update script ***s02_alignment.sh***.

This script includes alignment with **Bowtie 2**. It is possible to use other generic aligners with ATAC-seq data, such as BWA, or STAR. However, Bowtie 2 is conventionally used with ATAC-seq because it was initially designed for alignment of shorter reads. There is also a specialised aligner for ATAC-seq and ChIP-sec data, called Chromap.

After alignment this script uses **Samtools** to convert of SAMs to BAMs, sort, index and collecting flagstats for the produced BAMs. You should already be familiar with **Samtools** from the previous modules. More information about **Samtools** can be obtained here: https://www.htslib.org/doc/samtools.html

Update and run the script (submit it to the queue). The script should complete in **about 5 minutes**.

Note that the script suggests multithreading, using 12 threads for each step.

After the run you should see multiple output files in the dedicated sub-folder in your ***results*** folder, including SAM, BAM files, BAM index files (bai), flagstat results and Bowtie 2 logs.

Review the Bowtie 2 logs and flagstat output. How many reads were in the BAM files in total? Is the number of reads consistent with the FastQC results obtained previously? What proportion of reads have been successfully aligned? What was the main reason for failed alignments?

Step 3: Filtering BAM files

Review script ***s03_bam_filtering.sh***.

Note that the script suggests multithreading, using 6 threads for each step.

This script uses **Samtools** to

- Remove mitochondrial reads (mitochondria don't have histones)
- Remove reads aligned to sequences other than canonical chromosomes (e.g. aligned to unplaced contigs). The downstream tools can only consider genes annotated to canonical chromosomes.
- Remove reads with mapping quality < 20

Update and run the provided script. The script should take less than 1 min to run.

After the run you should see multiple output files in the dedicated sub-folder in your ***results*** folder, including BAM files, BAM index files (bai) and flagstat results.

Compare the flagstat output with flagstat results before filtering. What proportion of reads has been retained after filtering?

Step 4: Removing PCR duplicates

Review script ***s04_bam_deduplication.sh***

If PCR was used during the library preparation, each original read could be amplified many thousand times. Because each library should contain many millions of the original reads, only a certain proportion of the sequenced reads should represent PCR duplicates (e.g. less than 10-15%). However, even this small proportion should be detected and removed. Sometime the library preparation and sequencing may go wrong, causing high proportions of PCR duplicates in the sequencing data (e.g. more than 50%). Usually, such results should not be analysed. Instead, the cause of the problem should be identified, and library prep and sequencing may need to be repeated.

Although FastQC reports some "duplication rates", PCR duplicates in paired ends sequencing should be estimated from BAM files. Often this is done using Picard tools.

Update and run the script (submit it to the queue). The script should complete in **about 3 minutes**.

Find the percent of duplication in the Picard log files. Is the percent of duplication acceptable?

Step 5: Making final set of BAM files

Review script ***s05_make_final_bams.sh***

It's a simple script, which only sorts and index deduplicated BAMs, and makes a merged BAM, which will be needed for the down-stream analysis. All these tasks are done using **Samtools**. In real life, I wouldn't make a separate script for this in a pipeline. Most likely, I would combine it with the scripts for alignment, BAM filtering and deduplication into a single script.

Update and run the script (submit it to the queue). The script should complete in **about 3 minutes**.

Step 6: Making BigWig files

BigWig files are used for visualisation of coverage (or other quantitative trait) along the genomic position in IGV, and in other genome browsers (including online browsers, such as UCSC https://genome.ucsc.edu). The **BigWig** file format is widely used for peaks analysis in ATAC-seq, and it will be needed in several steps in our pipeline. **BigWig** is a binary extension of the **Wiggie** file format, which is described here: https://genome.ucsc.edu/goldenpath/help/wiggle.html

This is a short illustration of a **Wiggie** file:

```
fixedStep chrom=chr5 start=999 step=100

11

22

33

…
```

The example above may describe a coverage track on chromosome 5 starting from position 999 in a 100 base sliding windows. In contrast to **Wiggie** files, **BigWig** is a binary file format, you cannot see content of **BigWig** files directly, using text editors. Instead, we will use IGV to visualise the **BigWig** content.

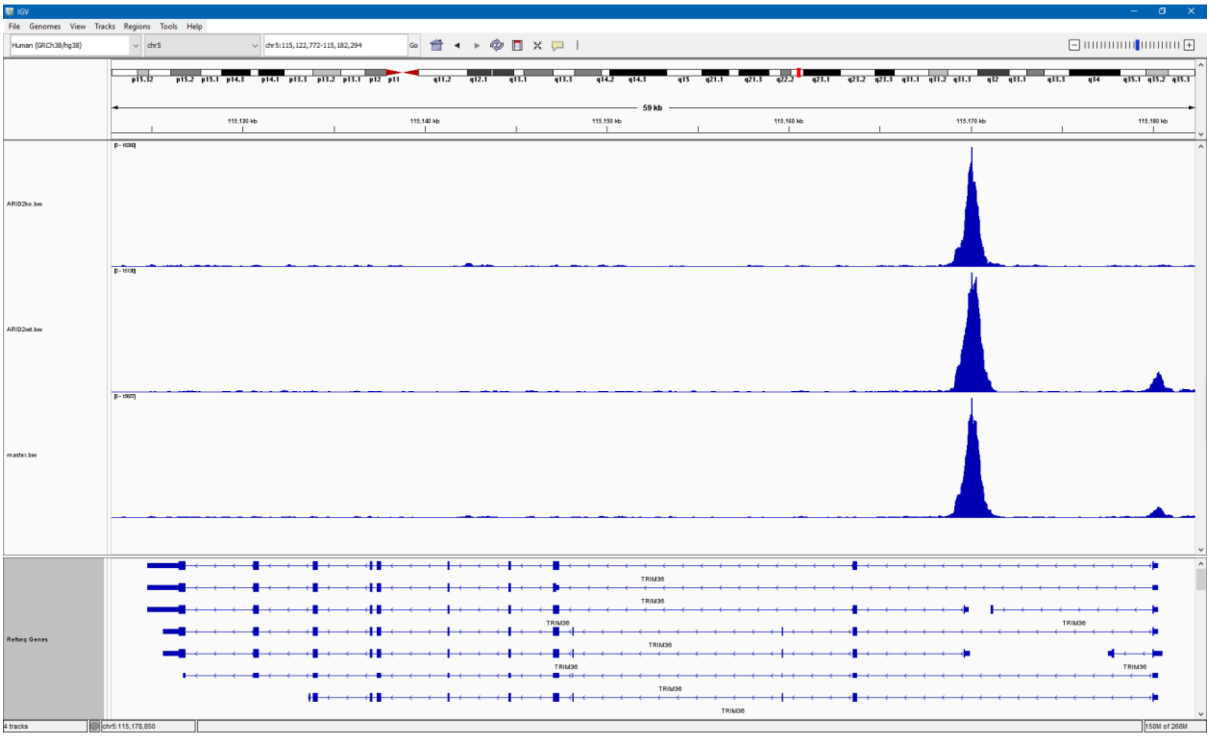
Review script ***s06_make_bigwigs.sh***. The script makes **BigWig** files from **BAMs** using **bamCoverage** tool from **deepTools**: https://deeptools.readthedocs.io/en/develop/content/list_of_tools.html . **deepTools** is a collection of diverse tools for working with BAM and BigWig files. Later we will use other **deepTools** tools to plot heatmaps for ATAC peaks.

Update and run the script (submit it to the queue). The script should complete in **about 5 minutes**.

After completion of the script, you should see 3 BigWig files the dedicated sub-folder in your ***results*** folder. Download these files to your laptop and review them in IGV. It is expected that you already have IGV on your laptop from the previous modules. If not, you may download and install it from here: https://igv.org/doc/desktop .

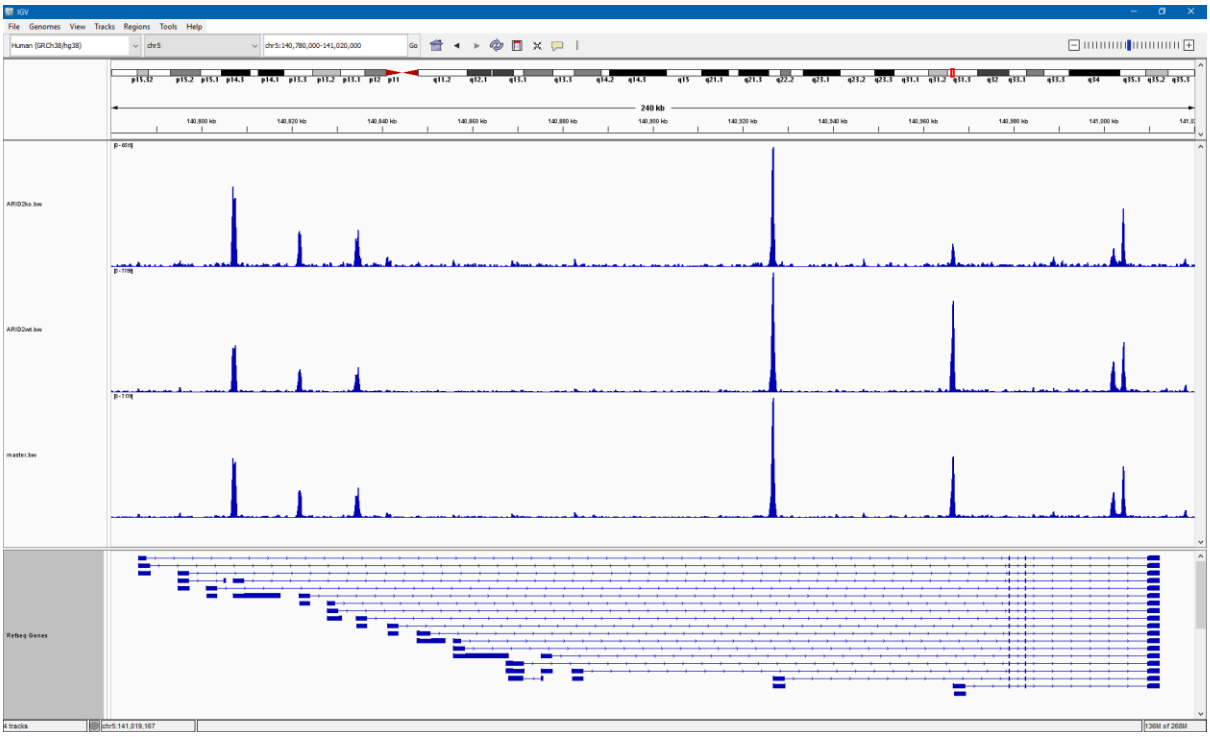
Open all three **BigWigs** (**KO**, **WT** and **Master**) at the same time. Make sure that you use **GRCH38**/hg38 genome. Why there is no coverage outside of chromosome 5?

Navigate to a gene called **TRIM36**. Right click on the **samples tracks**, make sure the "**autoscale**" option is selected, increase track height. Right click on the **genes track**. Make sure that the track is "**expanded**". Then, you may see the image like this:



What is the direction of transcription of this gene? Is the start of transcription on the right or on the left side of this image? Can you see any association between the open chromatin and transcription start sites (**TSS**) / **promoters** ? Can you see any difference in chromatin accessibility between **KO** and **WT** tracks?

Explore other genes, e.g. PCDHAC gene-family (select **chr5:140,780,000-141,020,000** range in IGV, and make the genes track "**squished**"):



Can you suggest any possible intronic **enhancer** in the PCDHAC gene-family region?

Step 7: Call peaks

Review script ***s07_call_peaks.sh.*** This script:

- Calls **peaks** (using the master BAM file, which combines data from KO and WT samples)
- Removes peaks called in "**blacklisted**" regions (known regions with anomalously high rate of peak "coverage", e.g. low complexity repetitive regions etc)
- Synchronises "**summits**" file with the "**peaks**" file, after filtering by the blacklisted regions

7.1 Calling peaks

Calling peaks is done using a tool called **macs2**. This is the most popular tool for peak calling in epigenetics data analysis; it is used not only for ATAC-seq, but for ChIP seq too. Briefly, it takes a BAM file and outputs three files:

- A file with detected **narrow peaks** (BED file format with some additional columns)
- A file with **summits** of the detected peaks (BED file format)
- An Excel file, containing information about peaks (this is a legacy file format, not used for analysis nowadays)

The peaks are detected by the locally excessive coverage, as compared to the background. Because **macs2** was originally developed for single-end sequencing data, it includes a sophisticated modelling step for estimating the sequenced fragments' length (needed for statistical assessment of the coverage). However, nowadays paired-end sequencing is recommended for ATAC-seq analysis. In the paired end sequencing the fragment length could be estimated directly from the paired reads at each end of the fragment. So, our script does not use this modelling step (**--nomodel** parameter).

You can find more details about macs2 here:

https://hbctraining.github.io/Intro-to-ChIPseq/lessons/05_peak_calling_macs.html

Illustrations in the above link show that peaks detected from single-end sequencing have a characteristic bimodal shape, coming from the reads in forward and reverse strands. Our paired end data do not show such a shape.

7.2 Filtering peaks by blacklisted regions

The blacklisted regions are removed using **bedtools**, a very popular toolset for working with **bed** files. The bed file format is widely used in many areas of bioinformatics. It's a simple tab-separated text file format (with some versions, like **bed6**, **bed12** etc.) described here:

https://genome.ucsc.edu/FAQ/FAQformat.html#format1

As it was noted above, the **narrowPeak** files are following the bed format rules, with addition of some extra-columns. However, because the initial columns of **narrowPeak** files are exactly following the bed format specifications, **bedtools** can work well with this type of file.

The **bedtools intersect** command used in our script is explained here (note option **-v**):

https://bedtools.readthedocs.io/en/latest/content/tools/intersect.html

7.3 Synchronising summit file with the filtered peaks file

To synchronise the summits with filtered peaks the script uses AWK syntax for two input files, as discussed in the lecture:

```
# Synchronise summits file with the peaks file w/o blacklisted regions
# Note that in both input files the peak name is in the 4th column
awk 'FNR==NR {a[$4]; next} $4 in a' \
  "${peaks_folder}/master_peaks_bl.narrowPeak" \
  "${peaks_folder}/master_summits.bed" \
  > "${peaks_folder}/master_summits_bl.bed"
```

Update and run the script (submit it to the queue). The script should complete in **about 1 minute**.

Review and compare content of the **peaks** ("narrowPeak") and **summits** ("bed") files.

Step 8: Peaks coverage

If you reach this step before the end of Wednesday, then you work faster than expected: you may take a (short!) break 😊

Review script ***s08_peaks_coverage.sh***

This script uses function **featureCounts** (from a package called **Subread**) to counts reads (=features) covering peaks (=regions) identified during the previous step. **featureCounts** is a popular generic function, which initially was more commonly used in RNA-seq analysis, to calculate the genes expression.

To calculate coverage over the peaks, **featureCounts** requires the BAM file, and the genomic coordinates of the peaks, provided in **SAF** format (Simplified Annotation Format, a tab-separated file with 5 columns: Name, Chr, Start, End and Strand). It could be easily obtained from the **narrowPeak** file using **AWK** (see provided script). Of course, the coordinates should be given for the genome version used in the BAM.

The **featureCounts** options are described here https://subread.sourceforge.net/featureCounts.html

Update and run the script (submit it to the queue). The script should complete in **about 5 minutes**.

After the run you should see the file with counts and a summary file in the dedicated sub-folder in your ***results*** folder. Review the content of these files. Was the count performed in single-end or in paired-end mode? Can you update the provided script to reflect the mode of sequencing used in this study? Estimate the **Fraction of Reads in Peaks (FRiP)**. It is one of the key QC metrics of ATAC-seq data. According to ENCODE ATACseq data standards FRiP should be at least 20%.

Step 9: Differential chromatin accessibility

As it was discussed during the lectures, the ATAC-seq peaks coverage reflects the chromatin accessibility. So, the purpose of this step is to identify the peaks, which coverage changed after the ARID2 knock-down. Of course, for a reliable detection of the changes caused by the knock-down, we would need several replicas of the knock-down sell line (ideally, with different versions of CRISPR guides). In such case the peaks count matrix (that would be generated during the previous step) is usually imported in R, and the peaks with differential chromatin accessibility are detected using the same packages as used for the differential genes expressions (e.g. DESeq2, edgeR or Limma-Voom).

In this practical, we don't have replicas: we only have one KO sample vs one WT. We still will import the counts in R. However, instead of the proper detection of the significantly changed peaks, we will only calculate the log-fold change: log2(KO/WT). Then we will select the peaks with at least 2-fold change to either side (Up or Down) and use these for the downstream analysis as differential peaks.

Of course, before calculating the log-fold change we will normalise the data by total count per sample (using CPM normalisation: Counts Per Million).

Review and update script **s09_dif_peaks.r** that implements the above steps.

R scripts cannot be directly executed by **qsub** on compute nodes on cluster, because by default the nodes don't have R, until you load the R module. So, to submit the R script to a compute node we should use a launcher *shell* script.

Review, update and execute (submit to a compute node with **qsub**) the launcher shell script provided (**s09_dif_peaks.sh**).

The script should take less than **1 min** to run.

After completion of the script, you should have the files with differential peaks in a dedicated sub-folder of your ***results*** folder. What is the numbers of detected up- and down- regulated peaks?

Step 10: Dif. Summits

The down-stream analysis steps (plotting heatmaps, identifying genes and motifs) will need summits coordinates, rather than the full information about peaks. So, we need to select summits that correspond to the previously identified differential peaks.

Review script **s10_dif_summits.sh** . It uses the AWK syntax for 2 input files, which we already used before, e.g.:

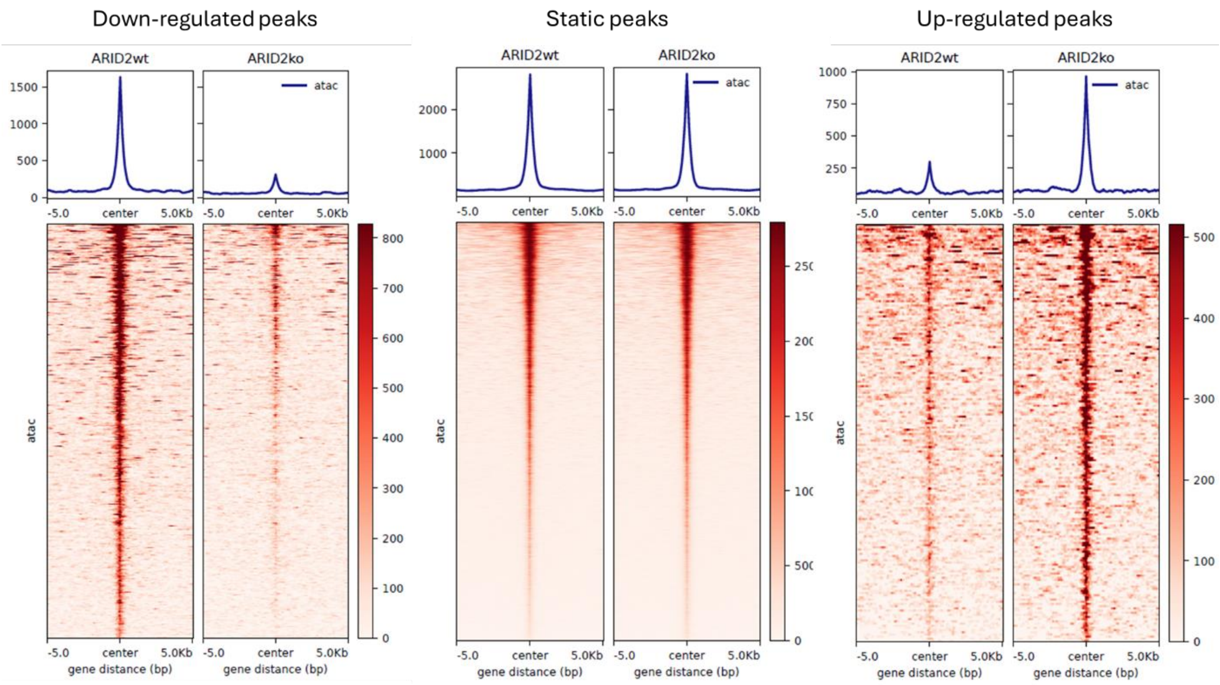
```
# Summits for peaks with increased coverage
# (i.e. more open chromatin after KO)
awk 'FNR==NR {a[$1]; next} $4 in a' \
  "${dif_peaks_folder}/master_peaks_up.txt" \
  "${peaks_folder}/master_summits_bl.bed" \
  > "${dif_summits_folder}/master_summits_up.bed"
```

To understand how it works, you should look at the format of the input files, and note that the peak IDs are in **1st** column of the **master_peaks_up.txt** and in the **4th** column of the **master_summits_bl.bed** .

Update and run **s10_dif_summits.sh** script (submit it to the queue). The script should complete in less than **1 minute**. After completion of the script, make sure that you have the expected output files in a dedicated sub-folder of your ***results*** folder.

Step 11: Dif. Summits heatmaps

Heatmaps like these are commonly used for visualising the ATAC-seq (and ChIP-seq) results:



You can see from this example that such heatmaps well illustrate the trends (in this case, the differential coverage) in the different groups of peaks.

Such plots could be generated using **deepTools** (we already used deepTools in step 6 to make **BigWig** files). The ATAC-seq heatmaps are created by **deepTools** in two steps:

- First the **computeMatrix** function is used to prepare data for the plot from BigWig files. The prepared matrix is saved in a zipped intermediate file.
- Then **plotHeatmap** function is used to actually draw the plot using the previously prepared matrix.

You may find more information about making heatmaps with **deepTools** here:

https://deeptools.readthedocs.io/en/develop/content/tools/computeMatrix.html

https://deeptools.readthedocs.io/en/develop/content/tools/plotHeatmap.html

Review **s11_dif_heatmaps.sh** script. The script makes separate plots for up- down- regulated and for static peaks, as shown above. Update and run the script (submit it to the queue). The script should complete within **5 minutes**. After completion of the script, make sure that you have the expected output files in a dedicated sub-folder of your ***results*** folder.

Step 12: Dif. Genes

12.1 Detecting genes in the peak's proximity

The main mechanism linking chromatin accessibility to the function is through regulating genes expression. Thus, knowing the genes in proximity of the differential ATAC peaks may help in understanding the biological consequences of the changed chromatin accessibility.

In this step we will use the HOMER (Hypergeometric Optimisation of Motif EnRichment) toolset. As you can see from the name of this toolset its main function is to identify the DNA motifs within the peaks in ATAC-seq or ChIP-seq studies. However, it also provides many other functions. One of them is the **Perl** script called **annotatePeaks.pl**. It provides much more annotations than only the nearest genes, see details here: http://homer.ucsd.edu/homer/ngs/annotation.html

However, for our analysis we will only use the genes for the downstream analysis.

HOMER is the only tool for which I decided not to use a module (although HOMER module is installed in Crescent2). In addition to the toolset itself, HOMER analysis is strongly dependent on the resources installed along with the tool. To facilitate reproducibility of research it could be preferrable to use containers in such situation. The container may include all resources, along with the tools. So, using the same container we can be sure that analysis is also performed within the same environment. Also, containers make it much easier to share the functional software, because it comes together with all resources and dependencies. Physically, the container is just a large file (see HOMER container in the resources provided for this practical). Using containers, requires specialised software. There are two main types of containers at the moment: **Docker** and **Singularity** containers. The software that runs **Docker** containers requires admin privileges, which, of course, is not provided to users on clusters. In contrast, **Singularity** containers can be used on clusters because they don't require admin rights. We will therefore use the module with Singularity software:

```
# Load required module
module load Singularity/3.11.0-1-system
singularity --version
```

After loading the module, we can run the annotatePeaks.pl script from within our HOMER container (which also has all the resources and dependencies) in the following way:

```
  # Run homer in container
  singularity exec "${containers_folder}/homer_v4.11_hg38.sif" \
    annotatePeaks.pl "${summit_file}" \
      hg38 \
      -size 200,200 \
      > "${dif_genes_folder}/${type}_annotated.txt"
```

Review script **s12_dif_genes.sh**. What does the "**-size"** parameter do? (look in the web page with documentation referred above). Could there be any typo in the provided code? Update and run the script. This script execution may take about **3 min**. After completion of the script, review the files produced in the dedicated sub-folder in your ***results*** folder. What annotations are available in addition to the gene names?

12.2 Functional interpretation of the gene lists

Often, the list of genes associated with the differential peaks are used for some functional interpretation.

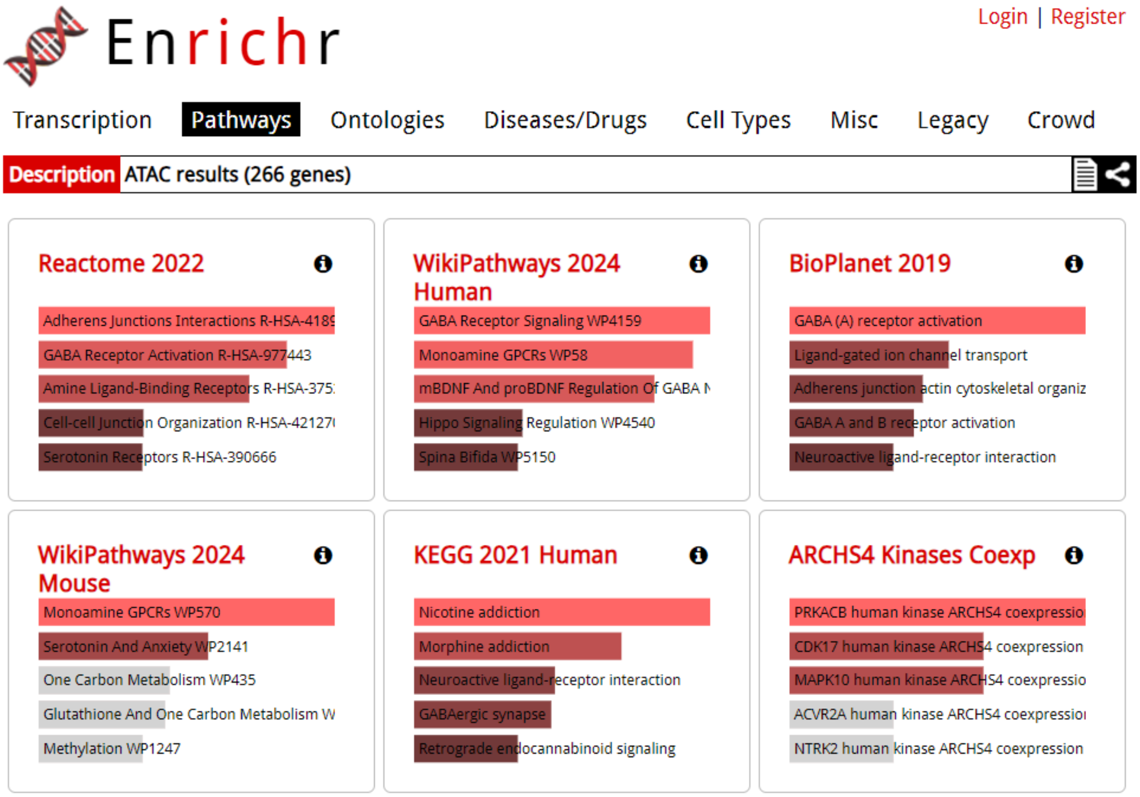
Functional interpretation of the gene lists is needed in many fields of bioinformatics. Initially, many tools and resources for functional interpretation of the gene lists were developed for analysis of differentially expressed genes in transcriptomics. However, these tools can also be used for epigenetics.

Typically, the functional interpretation includes some sort of enrichment analysis: comparing of the gene list with some background (e.g. with all human genes) and detecting if certain function is significantly more frequently observed in the gene list than in the total set of genes. Many tools and resources have been developed for functional interpretation of the genes. Detailed discussion of even the main tools and resources would be beyond the scope of this introductory epigenetics module. So, in this practical session you will try just one of them.

The script executed during the previous step generated a list of genes associated with differential peaks (***changed_genes.txt***). Copy this list from Crescent cluster to your laptop. Then, use this file for the functional analysis in the ***Enrichr*** web site: https://maayanlab.cloud/Enrichr .

It could be a good idea to explore the outputs generated by ***Enrichr*** after you started the next section's script (***s13_dif_motifs.sh***), because that script takes about 1 hour to run.

The ***Enrichr*** output may look like the following:



ATAC results (266 genes)

Reactome 2022

Adherens Junctions Interactions R-HSA-4189
GABA Receptor Activation R-HSA-977443
Amine Ligand-Binding Receptors R-HSA-375
Cell-cell Junction Organization R-HSA-42127
Serotonin Receptors R-HSA-390666

WikiPathways 2024 Human

GABA Receptor Signaling WP4159
Monoamine GPCRs WP58
mBDNF And proBDNF Regulation Of GABA N
Hippo Signaling Regulation WP4540
Spina Bifida WP5150

BioPlanet 2019

GABA (A) receptor activation
Ligand-gated ion channel transport
Adherens junction actin cytoskeletal organiz
GABA A and B receptor activation
Neuroactive ligand-receptor interaction

WikiPathways 2024 Mouse

Monoamine GPCRs WP570
Serotonin And Anxiety WP2141
One Carbon Metabolism WP435
Glutathione And One Carbon Metabolism W
Methylation WP1247

KEGG 2021 Human

Nicotine addiction
Morphine addiction
Neuroactive ligand-receptor interaction
GABAergic synapse
Retrograde endocannabinoid signaling

ARCHS4 Kinases Coexp

PRKACB human kinase ARCHS4 coexpressio
CDK17 human kinase ARCHS4 coexpression
MAPK10 human kinase ARCHS4 coexpressio
ACVR2A human kinase ARCHS4 coexpressio
NTRK2 human kinase ARCHS4 coexpression

Review what resources and algorithms have been used by ***Enrichr***. Are all these resources relevant to our experiment? Exploring ***Enrichr*** results focus on bioinformatics tools and resources. Do not over-interpret the biology in this case: remember that the source data were limited to chromosome 5 only, and we did not have replicas for the proper detection of the differential peaks during step 9.

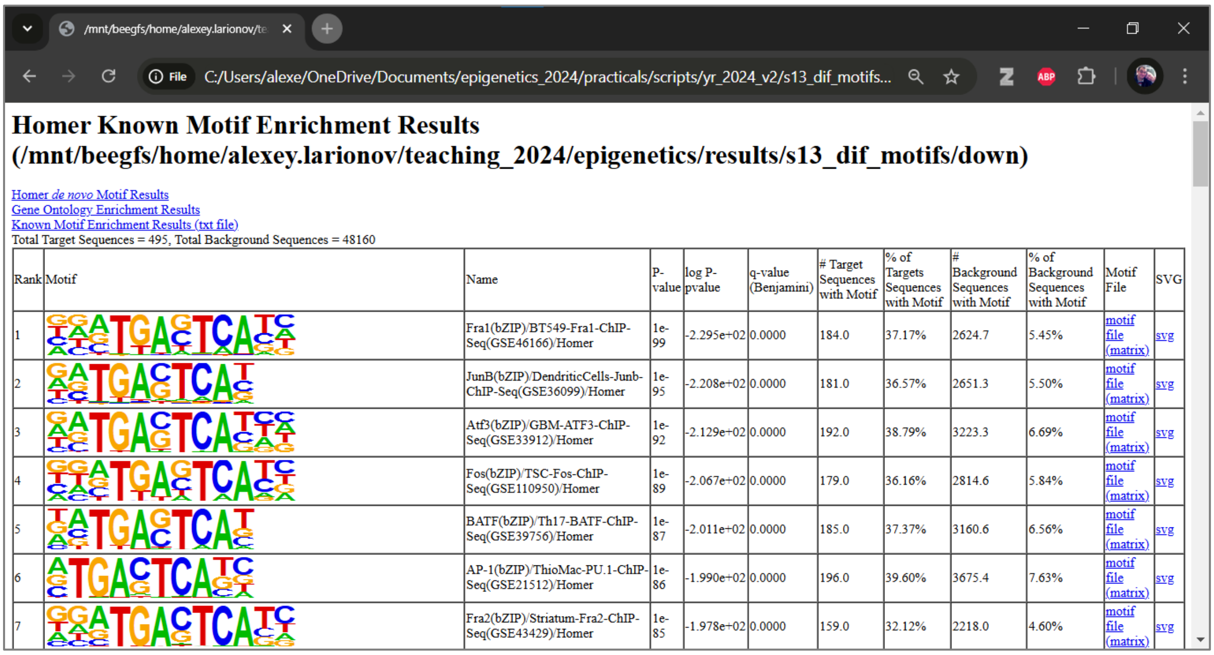
Step 13: Dif. Motifs

Review script **s13_dif_motifs.sh**.

As mentioned above, HOMER's main goal is the motif enrichment analysis. This functionality is implemented in the **findMotifsGenome.pl** Perl script:

http://homer.ucsd.edu/homer/ngs/peakMotifs.html

Update and run the script (note that it requires 12 CPUs to run). This script takes **about 1hr to run.** So, allocate sufficient time in the PBS directives (e.g. **-l walltime=03:00:00** and **-q three_hour**). Review the output after the script completion. The output files will include motifs from already known databases (**knownResults.html**) and potential new motifs suggested by HOMER (**homerResults.html**). The output for the known motifs may look like the following:


/mnt/beegfs/home/alexey.larionov/te
File
C:/Users/alexe/OneDrive/Documents/epigenetics_2024/practicals/scripts/yr_2024_v2/s13_dif_motifs...

Homer Known Motif Enrichment Results (/mnt/beegfs/home/alexey.larionov/teaching_2024/epigenetics/results/s13_dif_motifs/down)

Homer *de novo* Motif Results
Gene Ontology Enrichment Results
Known Motif Enrichment Results (txt file)
Total Target Sequences = 495, Total Background Sequences = 48160

Rank	Motif	Name	P-value	log P-pvalue	q-value (Benjamini)	# Target Sequences with Motif	% of Targets Sequences with Motif	# Background Sequences with Motif	% of Background Sequences with Motif	Motif File	SVG
1		Fra1(bZIP)/BT549-Fra1-ChIP-Seq(GSE46166)/Homer	1e-99	-2.295e+02	0.0000	184.0	37.17%	2624.7	5.45%	motif file (matrix)	svg
2		JunB(bZIP)/DendriticCells-Junb-ChIP-Seq(GSE36099)/Homer	1e-95	-2.208e+02	0.0000	181.0	36.57%	2651.3	5.50%	motif file (matrix)	svg
3		Atf3(bZIP)/GBM-ATF3-ChIP-Seq(GSE33912)/Homer	1e-92	-2.129e+02	0.0000	192.0	38.79%	3223.3	6.69%	motif file (matrix)	svg
4		Fos(bZIP)/TSC-Fos-ChIP-Seq(GSE110950)/Homer	1e-89	-2.067e+02	0.0000	179.0	36.16%	2814.6	5.84%	motif file (matrix)	svg
5		BATF(bZIP)/Th17-BATF-ChIP-Seq(GSE39756)/Homer	1e-87	-2.011e+02	0.0000	185.0	37.37%	3160.6	6.56%	motif file (matrix)	svg
6		AP-1(bZIP)/ThioMac-PU.1-ChIP-Seq(GSE21512)/Homer	1e-86	-1.990e+02	0.0000	196.0	39.60%	3675.4	7.63%	motif file (matrix)	svg
7		Fra2(bZIP)/Striatum-Fra2-ChIP-Seq(GSE43429)/Homer	1e-85	-1.978e+02	0.0000	159.0	32.12%	2218.0	4.60%	motif file (matrix)	svg

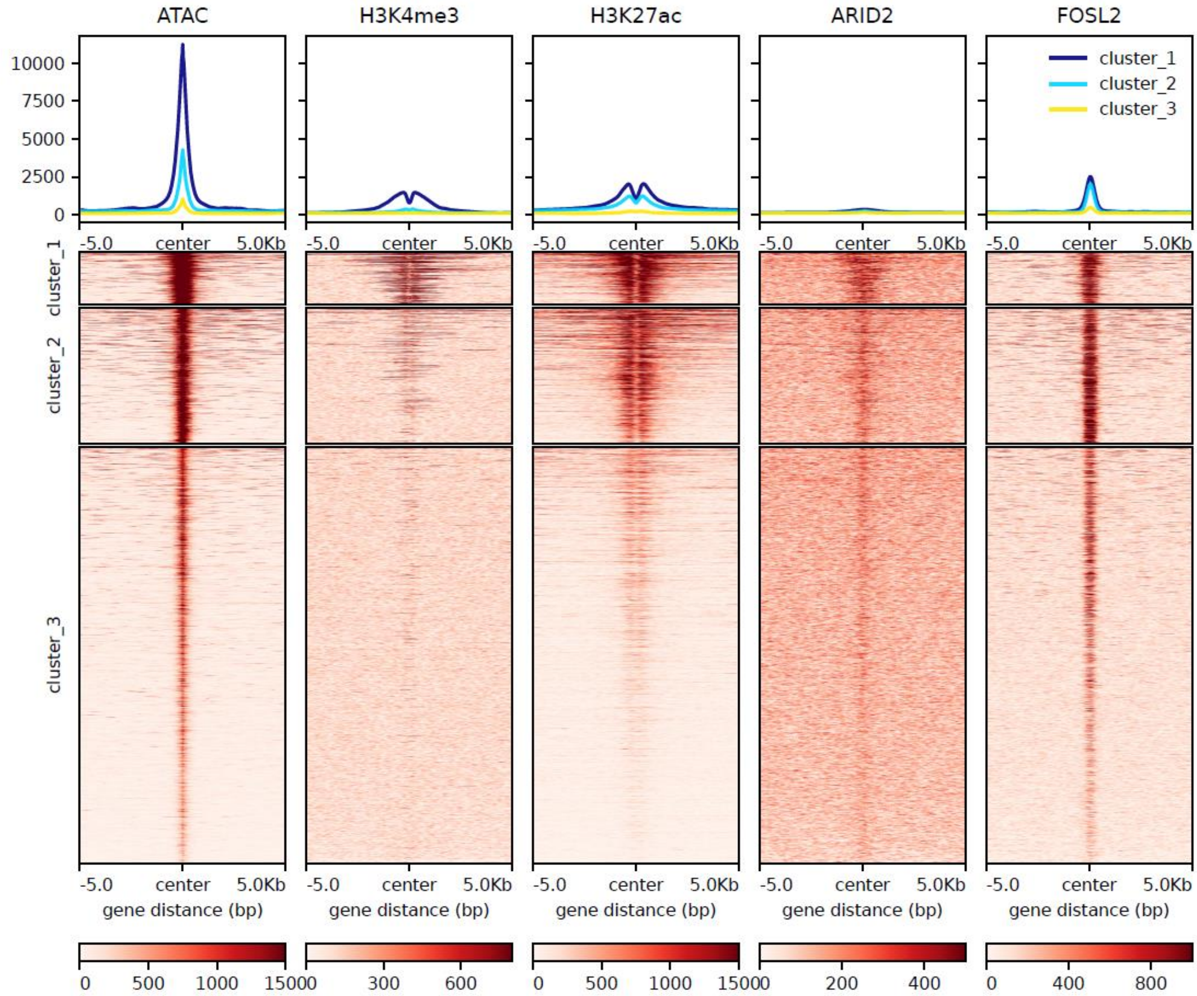
Step 14: Combining ATAC-seq and ChIP-Seq (optional)

ATAC-seq data are never analysed in isolation. The open chromatin regions could be located near promoters or enhancers (see lecture about the corresponding **ChIP-seq** markers). Also, the open chromatin regions are often occupied by transcription factors (**TFs**).

Script **s14_combining_atac_and_chip_seq.sh** makes a joint heatmap, adding some **ChIP-seq** data obtained from the same melanoma cell line. The script makes heatmaps using **deepTools** (as we did previously in Step 11). However, it uses the additional BigWig files and applies clustering on the heatmaps. The additional **ChIP-seq** BigWig files were provided in the ***data*** folder.

Review, update and run the script, it may take up to **10 min** to run.

The script may produce the following figure:



Can you conclude whether the ATAC peaks are stronger in enhancers or promoters in this cell line?

Which of the shown transcriptional factors is equally associated with both enhancers and promoters, and which is more common in the promoters?

Step 15: Combining ATAC-seq and RNA-seq (optional)

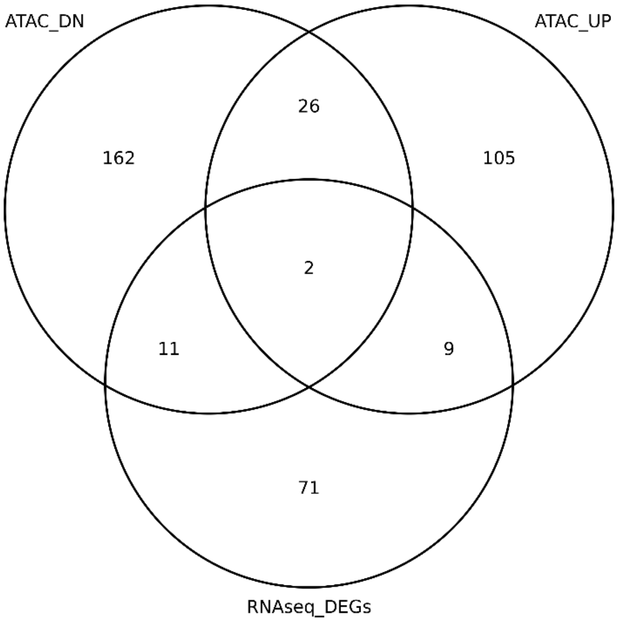
Finally, the differentially expressed genes (DEGs) are also provided to you. These genes were obtained by **RNA-seq** in the same knock-down experiment.

Explore the provided R-script **s15_combining_atac_and_rna_seq.r**

Update and run it (remember to use the shell launcher script to run it on cluster!)

It should take less than 1 minute to run. The script should show overlaps between the genes associated with differential **ATAC-seq** peaks (step 12) and the DEGs reported by **RNA-seq**.

Depending on the settings used in the previous analysis (step 12) the output may look like this:



It looks like some genes might be associated with both: up- and down- regulated ATAC-seq peaks.

How could this be?

Explore in IGV the KO and WT results for the DEGs overlapping with differential ATAC-seq genes (as we did in step 6)

Conclusion

This practical guided you through a complex bioinformatics pipeline which included 15 steps and more than 10 different tools: FastQC, MultiQC, Trim-Galore, Bowtie 2, Samtools, deepTools, Subread, BedTools, HOMER, AWK, shell and R scripting. You practiced using a High-Performance Computing cluster. You learned key steps of ATAC-seq data analysis, including alignment, peak calling, differential peak detection, peaks visualisation in IGV and using heatmaps, finding genes and motifs associated with the peaks, and interpreting ATAC-seq data in context of ChIP-seq and RNA-seq results.

Not least, you learned how to organise scripts, logs, data, results and resources in a complex project that involved many tens of files as input and output; and you got a hands-on experience with analysis of real-life epigenetics data.

Well done!

You have completed the Epigenetic Practical Session.

PS: keep your scripts, logs (including the job output files provided buy PBS on cluster) and other outputs produced during this practical: they may be needed for your assignment.