ATAC-seq

Thomas Juettemann, EMBL-EBI
Lars Grønvold, NMBU
aqua_faang_course@ebi.ac.uk

May 10-12, 2021

1 Introduction

The aim of this section is to investigate open chromatin regions in different tissues. There are 4 sample replicates from brain (B), 4 from liver (L) and 4 from spleen (SP).
We are starting out with the BAM alignment files, which have been downloaded into the folder **train-aquafaang-bioinf/data** in the training home directory. The bam files are sorted by mapping position. Duplicates/multi-mapped reads and reads mapping to mitochondrial chromosome or unplaced scaffolds have been removed from all files.
For simplicity, we are only processing replicate 1 from the brain samples. For the other sample replicates, all files have been made available in the respective folder.
All processing is done using Docker with containerised applications.

Container

Tool	Container	Documentation
samtools	juettemann/samtools:latest	http://www.htslib.org/doc/sam.html
Genrich	juettemann/genrich:0.6	https://github.com/jsh58/Genrich
bedgraphtobigwig	quay.io/biocontainers/ucsc-bedgraphtobigwig:377–h0b8a92_2-	https://genome.ucsc.edu/goldenPath/help/bigWig.html#Ex3
ataqv	quay.io/biocontainers/ataqv:1.2.1 –py27h97b49fa_1	https://github.com/juettemann/ataqv/tree/patch-1

Table 1: Container used in this tutorial

File formats

Extension	Description	Documentation
sam	Sequence Alignment/Map file format	http://www.htslib.org/doc/sam.html
sam	Binary Format of Sequence Alignment/Mapping (SAM)	https://genome.ucsc.edu/ENCODE/fileFormats.html#BAM
bigWig	Genome Browser Signal (Wiggle) Files in Indexed Binary Format	https://genome.ucsc.edu/ENCODE/fileFormats.html#bigWig
narrowPeak	ENCODE narrowPeak: Narrow (or Point-Source) Peaks format	https://genome.ucsc.edu/FAQ/FAQformat.html#format12
faidx	An index enabling random access to FASTA and FASTQ files	http://www.htslib.org/doc/faidx.html

Table 2: File formats used in this tutorial

2 Variables

In order to make commands shorter and allow less chance for errors, we are setting a few environment variables. These have already been set in the .bashrc.

```
export dir_atac="$HOME/train-aquafaang-bioinf/atac-seq"
export mnt_dir_atac="type=bind,source=$dir_atac,target=/mnt"
```

3 Reference Genome

Like other *-seq experiments, also ATAC-seq relies on a reference genome We will be using the *International Cooperation to Sequence the Atlantic Salmon Genome version 2 (ICSASG_v2)* genome from 2015 for this course. General information about the assembly can be found at Ensembl (https://www.ensembl.org/Salmo_salar/Info/Index). On that page you also find links to FASTA files with the DNA sequence (reference genome) and gtf files with the Ensembl gene annotation.

3.1 Download

Your first task is to find, download and unzip the atlantic salmon reference sequence

Task

1. Open a terminal
2. Change directory to **$dir_atac/ref_genome**.
3. On the Ensembl assembly page, find the DNA sequence in FASTA format.
4. Click on the link (you will be redirected to a FTP directory).
5. Locate **toplevel** FASTA file (.fa).
6. Right-click on it and choose *Copy Link Location*
7. In the terminal, download the file using *wget*
8. Unzip (*gunzip*)

Solution

```
cd $dir_atac/ref_genome
# Latex did not break up the link properly, you need to copy
# both parts of the URL seperately
wget \
http://ftp.ensembl.org/pub/release-103/fasta/salmo_salar/dna
/Salmo_salar.ICSASG_v2.dna.toplevel.fa.gz
gunzip Salmo_salar.ICSASG_v2.dna.toplevel.fa.gz
```

3.2 Create FASTA index

Using an fai index file in conjunction with a FASTA file containing reference sequences enables efficient access to arbitrary regions within those reference sequences. The index file typically has the same filename as the corresponding FASTA, with .fai appended. An fai index file is a text file consisting of lines each with five TAB-delimited columns for a FASTA file:

1. NAME Name of this reference sequence
2. LENGTH Total length of this reference sequence, in bases
3. OFFSET Offset in the FASTA/FASTQ file of this sequence's first base
4. LINEBASES The number of bases on each line
5. LINEWIDTH The number of bytes in each line, including the newline

Using a samtools container, index the fasta file (*Salmo_salar.ICSASG_v2.dna.toplevel.fa*). This resulting index file's name should be *Salmo_salar.ICSASG_v2.dna.toplevel.fa.fai*

Task

1. Pull the container **juettemann/samtools:latest**
   (Documentation: http://www.htslib.org/doc/samtools-faidx.html)
2. Using **faidx** from the container, index *Salmo_salar.ICSASG_v2.dna.toplevel.fa*

Solution

```
# docker run pulls the container automatically if not available
docker run \
--mount $mnt_dir_atac \
juettemann/samtools:latest faidx \
/mnt/ref_genome/Salmo_salar.ICSASG_v2.dna.toplevel.fa
```

3.3 Create list of chromosomes

At a later stage in the tutorial, we need a list of all autosomal chromosomes. Using the command line utilities grep and cut, create a file containing only the chromosome names. As mentioned above in the faidx description, the name of the reference sequence is the first column in the FASTA index file, and chromosomes start with *ssa*.

Task

1. Using **grep** and **cut**, extract all chromosome names
2. Store the names in in the **$dir_atac/ref_genome** folder in a file named **Ssal_chrm.txt**

Solution

```
cd $dir_atac/ref_genome/
grep ^ssa Salmo_salar.ICSASG_v2.dna.toplevel.fa.fai | \
cut -f 1 > Ssal_chrm.txt
```

3.4 Create table with chromosome sizes

We will be converting bedgraph files to bigWig later. For that conversion, we need a file containing chromosome name and its size. We already know that the chromosome name is the first column, and looking at the description of samtools faidx above, we see that the length is stored in the second column.

Task

1. Using **grep** and **cut**, extract all chromosome names and their length.
2. Store the names in the **$dir_atac/ref_genome** folder in a file named **Ssal_chrm_sizes.txt**

Solution

```
cd $dir_atac/ref_genome/
grep ^ssa Salmo_salar.ICSASG_v2.dna.toplevel.fa.fai | \
 cut -f 1,2 >Ssal_chrm_sizes.txt
```

4 Peak Calling

The tool we are using for peak calling is Genrich. Genrich was developed by John Gaspar, previous of Harvard FAS Informatics. The default method is for ChIP-seq peak-calling, but there is an ATAC-seq specific mode. To account for the manner in which the Tn5 enzyme inserts itself into the genome, Genrich centers a 100 bp interval at each end of each fragment. It uses the log-normal distribution to calculate p-values for each base of the genome.
Some advantages over MACS include how much faster it runs (it is written in C), as well as the capability of calling peaks for multiple replicates collectively. Unfortunately, Genrich is yet to be published.

4.1 Sort by read names

Genrich requires that the BAM files are sorted by read names, not by mapping position, which they currently are.
Sort bam file by reads, name it *B-1.sortedByRead.bam* and store it in *results/genrich/*. Samtools documentation: http://www.htslib.org/doc/samtools.html

Task

1. Using the **juettemann/samtools:latest** container **sort** $dir_atac/data/B-1.bam by read names
   (Documentation http://www.htslib.org/doc/samtools-sort.html)
2. Name the output file **B-1.sortedByRead.bam** and store it in the folder **$dir_atac/results/genrich/**
3. Set number of sorting and compression threads to 8

Solution

```
docker run \
--mount $mnt_dir_atac \
juettemann/samtools:latest sort -n -@ 8 \
-o /mnt/results/genrich/B-1.sortedByRead.bam
/mnt/data/B-1.bam
```

4.2 Peak calling

Next step is the peak calling with Genrich.

Task

1. Pull the container **juettemann/genrich:1.0**
   (Documentation https://github.com/jsh58/Genrich)
2. Using **$dir_atac/results/genrich/B-1.sortedByRead.bam** as input, create:
   - ENCODE narrowPeak format
   - bedgraph-ish file for pileups and p-values
   - Set:
     - Maximum q-value = 0.05
     - print status to stderr
     - ATAC-seq mode
   - Redirect the STDERR output to **genrich_B-1.log**

Solution

```
docker run \
--mount $mnt_dir_atac \
juettemann/genrich:0.6 \
-t  /mnt/results/genrich/B-1.sortedByRead.bam \
-o /mnt/results/genrich/B-1.narrowPeak \
-k /mnt/results/genrich/B-1.k.bedGraph \
-q 0.05 \
-v -j \
2> genrich_B-1.log
```

5 Quality Control

We will be using two tools for investigating the results of the peak calling, Integrated Genome Browser (IGV) and ataqv.

5.1 IGV

A common way to get an idea about the quality of the results is looking at the data in a genome browser. As most publicly available web based browsers have a limit on the size of the file that you can upload, we will be using IGV. Visualising the read density along the chromosomes (bigwig files) and the peaks defined by Genrich(narrowPeak file).

5.1.1 Convert bedGraph to BigWig

The bigWig format is for display of dense, continuous data that will be displayed as a graph. The main advantage of the bigWig files is that only the portions of the files needed to display a particular region are transferred, so for large data sets bigWig is considerably faster than any non-indexed formats.

Create bedGraph

The .bedGraph-ish out from Genrich can easily be converted to bigWig. To create a compliant bedGraph file, extra columns and headers must be removed.

Task

1. Using **grep** and **cut**, extract all chromosome names and columns 1-4
2. Store the names in the **$dir_atac/results/genrich/** folder in a file named **B-1.k.fixed.bedGraph**

Solution

```
cd $dir_atac/results/genrich/
grep ^ssa B-1.k.bedGraph | \
cut -f 1-4 > B-1.k.fixed.bedGraph
```

UCSC provides a tool to convert bedGraph to bigWig (part of the Kent tools). It takes 3 arguments:

bedGraphToBigWig <*in.bedGraph*> <*chrom.sizes*> <*outBigWig.bw*>

Task

1. Using **quay.io/biocontainers/ucsc-bedgraphtobigwig:377–h0b8a92_2**, convert **B-1.k.fixed.bedGraph** to **B-1.bw**
2. Save the file in **$dir_atac/results/genrich/**

Solution

```
docker run \
--mount $mnt_dir_atac \
quay.io/biocontainers/ucsc-bedgraphtobigwig:377--h0b8a92a_2 \
bedGraphToBigWig \
/mnt/results/genrich/B-1.k.fixed.bedGraph \
/mnt/ref_genome/Ssal_chrm_sizes.txt \
/mnt/results/genrich/B-1.bw
```

5.1.2 Launch IGV

The IGV browser is launched from the command line. Open a new terminal and run:

igv

5.1.3 Load Genome and Tracks

IGV loads its genomes from the Broad Institute, and these genomes use RefSeq accessions as chromosome names. Refseq accessions are different from the ones used by the submitter of the genome (which Ensembl uses), therefore we need to load this standard genome manually.

Task

1. In the IGV window, click *Genomes* -> *Load Genome from File* -> navigate to *train-aquafaang-bioinf/atac-seq/ref_genome/* and select
   -> **Salmo_salar.ICSASG_v2.dna.toplevel.fa**
2. In the IGV window, click *File* -> *Load from File* -> navigate to *train-aquafaang-bioinf/atac-seq/results/genrich/* and select
   -> *Salmo_salar.ICSASG_v2.103.chr.sorted.gtf* -> *B-1.bw*
   -> *B-1.narrowPeak*
   -> *L-1.bw*
   -> *L-1.narrowPeak*

If you want to go futher and get an idea about the differences between the replicates, simply load all other *.bw and *.narrowPeak files the same way.

5.1.4 Explore using IGV

We will investigate two genes, *Neurogenic differentiation factor* (**NDF1**) which is involved in neuron development, and *Hepatocyte nuclear factor 1-alpha* (**hnf1a**) which is involved in liver development.

NDF1

The genomic coordinates of NDF1 are *ssa25:27,019,821-27,030,126*. Copy and paste them in the seach box and click Go. Figure 1 shows the expected result. The sample B-1 has open chromatin upstream of the NDF1 gene (peak_59291), whereby the liver samples show closed chromatin.

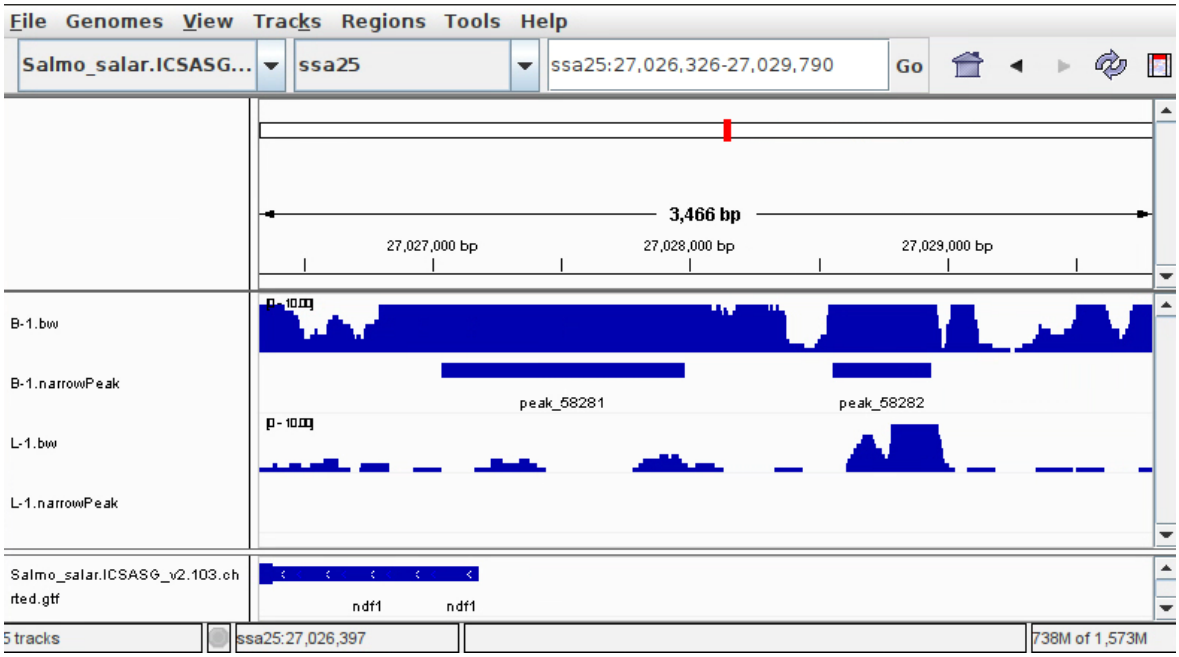

Figure 1: IGV showing area around the NDF1 gene

hnf1a

The genomic coordinates of hnf1a are *ssa20:19,522,855-19,530,228*. Copy and paste them in the search box and click Go. In figure 2, the roles are reversed - the liver sample shows open chromatin upstream of the hnf1a gene (peak_51067).

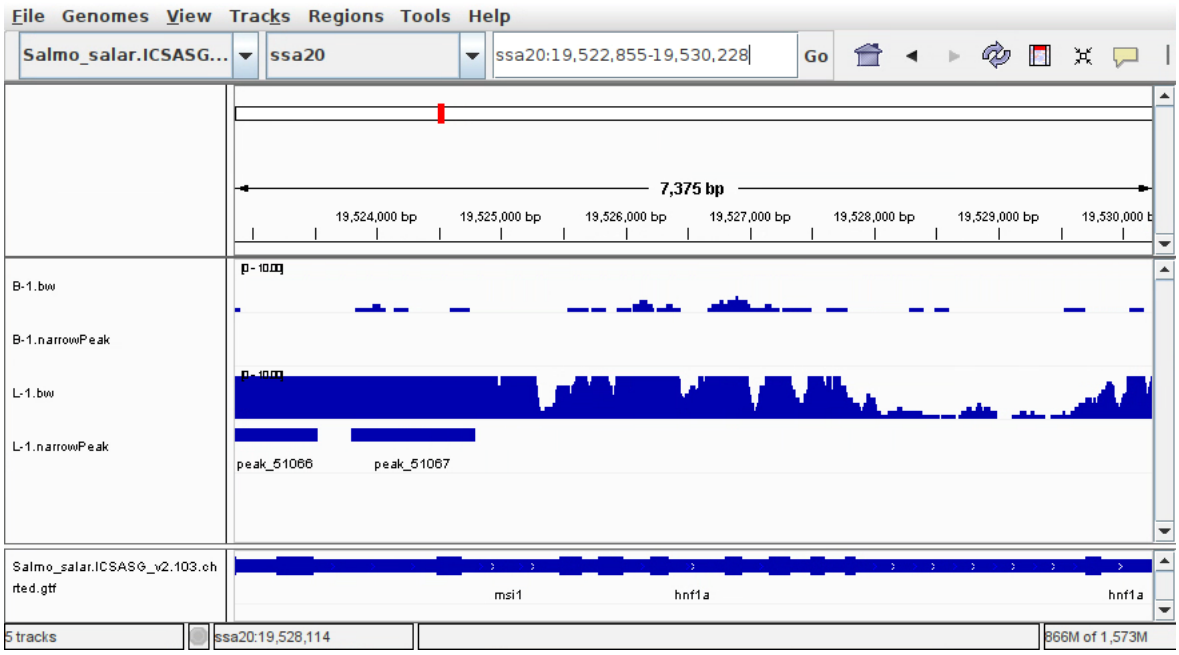

Figure 2: IGV showing area around the hnf1a gene

The overlap between the msi1 gene and hnf1a can easily be explained by having a look at the Ensembl genome browser (figure 3), the last intron of transcript 202 extends into hnf1.

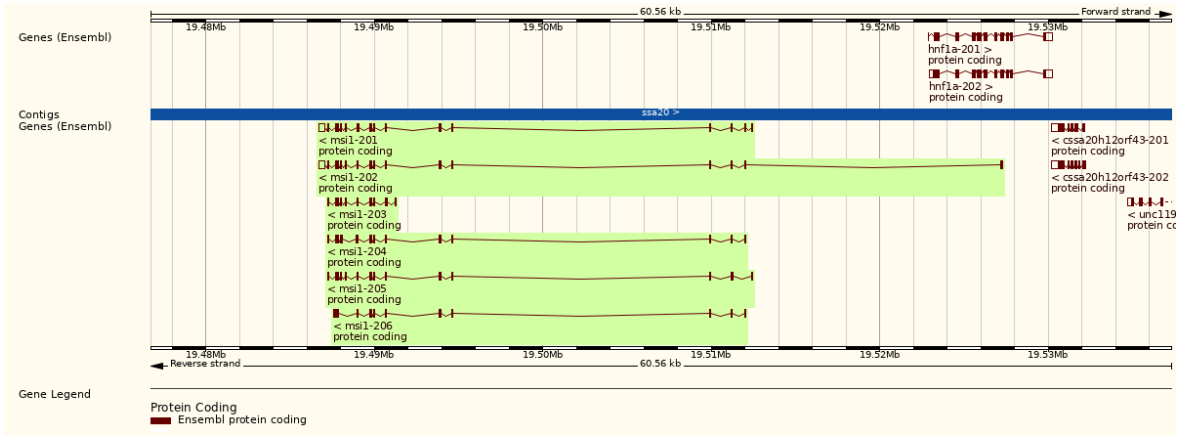

Figure 3: msi1 gene in the Ensembl genome browser

5.2 ATAQV

Ataqv is a toolkit for measuring and comparing a broad range of ATAC-seq quality metrics. It makes it easier to spot differences that might be caused by library prep or sequencing. We will investigate two features: high quality autosomal alignment (HQAA) fragment length distribution (FLD) and transcription start sites (TSS) enrichment.

5.2.1 Index BAM

To run ATAQV the BAM file needs to be indexed. Using samtools, index B-1.bam.

Task

1. Using **juettemann/samtools** index, convert **$dir_atac/data/B-1.bam** to **B-1.bw**
2. Name the output **B-1.bam.bai** and save the file in **$dir_atac/data**

Solution

```
docker run \
--mount $mnt_dir_atac \
juettemann/samtools index \
/mnt/data/B-1.bam /mntdata/B-1.bam.bai
```

5.2.2 Run ATAQV

The main program, ataqv, examines our aligned reads and reports some basic metrics.

Task

1. Using **quay.io/biocontainers/ataqv:1.2.1–py27h97b49fa_1 ataqv** run **ataqv** (Documentation: https://github.com/ParkerLab/ataqv)
2. ataqv needs information about our terminal, pass **-e TERM=xterm** to the container.
3. Input
    - organism name is Ssal
    - alignment file $dir_atac/data/B-1.bam
    - peak file is $dir_atac/results/genrich/B-1.narrowPeak
    - TSS file is $dir_atac//ref_genome/SsalTSS_NCBI.bed
    - autosomal reference is $dir_atac//ref_genome/Ssal_chrm.txt
4. Output
    - metrics file $dir_atac/results/ataqv/B-1.ataqv.json

Solution

```
docker run \
-e  TERM=xterm  \
--mount $mnt_dir_atac  \
quay.io/biocontainers/ataqv:1.2.1--py27h97b49fa_1 \
ataqv  Ssal /mnt/data/B-1.bam \
--peak-file /mnt/results/genrich/B-1.narrowPeak \
--tss-file /mnt/ref_genome/SsalTSS_NCBI.bed \
--autosomal-reference-file /mnt/ref_genome/Ssal_chrm.txt \
--metrics-file /mnt/results/ataqv/B-1.ataqv.json
```

5.2.3 Visualising results

The mkarv script that collects ataqv results and sets up a web application to visualize them.

Task

1. Using **quay.io/biocontainers/ataqv:1.2.1–py27h97b49fa_1 ataqv** run **mkarv** (Documentation: https://github.com/ParkerLab/ataqv)
2. Pass **-e TERM=xterm** to the container.
3. Input
    - results/mkarv/ataqv_good_html
    - whitespace separated list of all 8 B & L *.ataqv.json files, e.g. **results/ataqv/B-1.ataqv.json**

Solution

```
docker run \
-e  TERM=xterm  \
--mount $mnt_dir_atac  \
quay.io/biocontainers/ataqv:1.2.1--py27h97b49fa_1 \
mkarv   \
/mnt/results/mkarv/ataqv_good_html \
/mnt/results/ataqv/B-1.ataqv.json \
/mnt/results/ataqv/B-2.ataqv.json \
/mnt/results/ataqv/B-3.ataqv.json \
/mnt/results/ataqv/B-4.ataqv.json \
/mnt/results/ataqv/L-1.ataqv.json \
/mnt/results/ataqv/L-2.ataqv.json \
/mnt/results/ataqv/L-3.ataqv.json \
/mnt/results/ataqv/L-4.ataqv.json
```

Now we do the same for the spleen samples.

Task

1. Using **quay.io/biocontainers/ataqv:1.2.1–py27h97b49fa_1 ataqv** run **mkarv** (Documentation: https://github.com/ParkerLab/ataqv)
2. Pass **-e TERM=xterm** to the container.
3. Input
   - results/mkarv/ataqv_bad_html
   - whitespace separated list of all 4 SP *.ataqv.json files, e.g. **results/ataqv/SP-1.ataqv.json**

Solution

```
docker run  \
-e  TERM=xterm \

--mount $mnt_dir_atac \
quay.io/biocontainers/ataqv:1.2.1--py27h97b49fa_1  \
mkarv   \
/mnt/results/mkarv/ataqv_bad_html \
/mnt/results/ataqv/SP-1.ataqv.json \
/mnt/results/ataqv/SP-2.ataqv.json \
/mnt/results/ataqv/SP-3.ataqv.json \
/mnt/results/ataqv/SP-4.ataqv.json
```

In your webbrowser, navigate to directory $dir_atac/results/markv/ataqv_good_html and open the index html. Explore the different graphs and also the table tab.

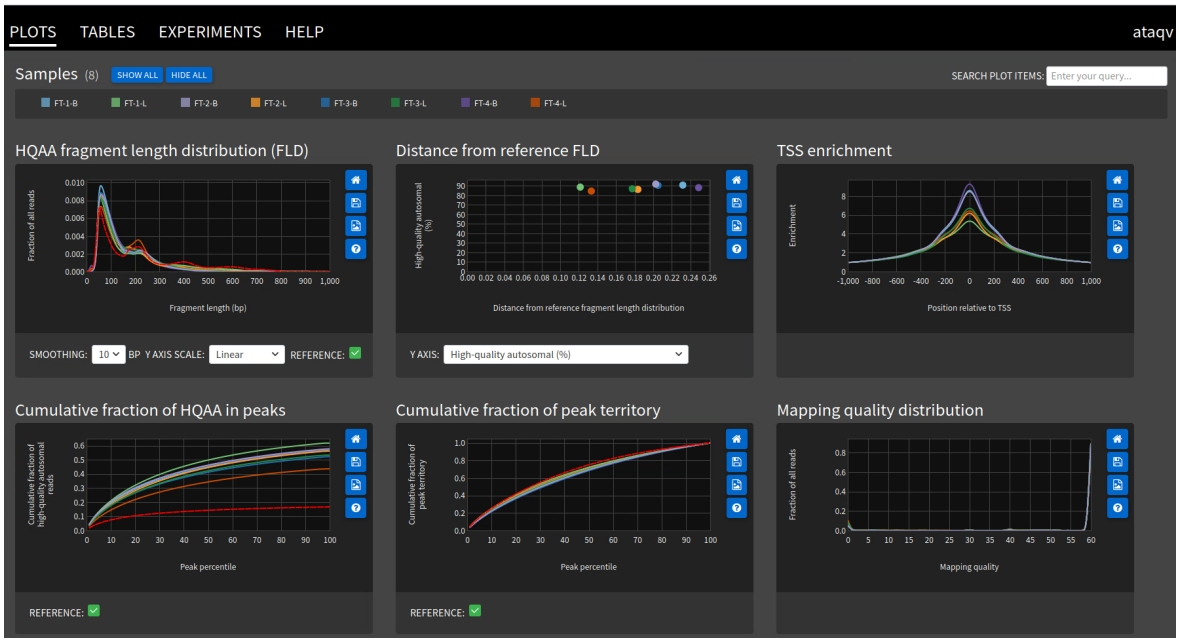

Figure 4: Result of a successful ATAC-seq experiment visualised using ataqv

Now do the same for spleen sample results ($dir_atac/results/markv/ataqv_good_html/index.html). What differences do you see?

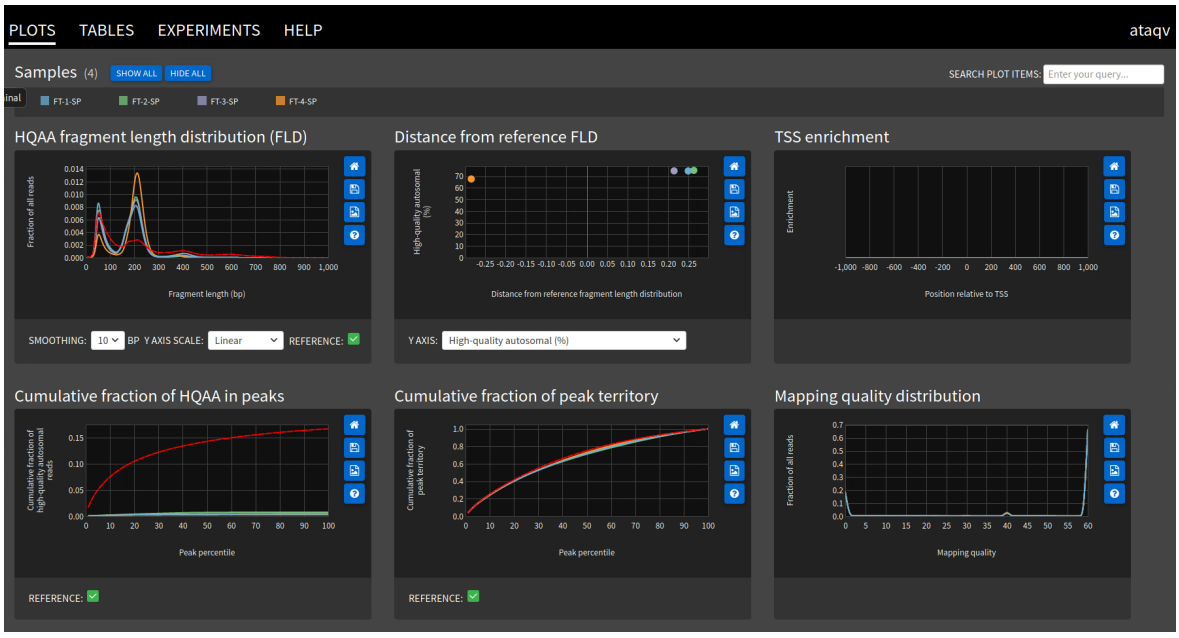

Figure 5: Result of an unsuccessful ATAC-seq experiment visualised using ataqv