Epigenomic Assigenement

Rumiano Letizia

2025-06-08

Contents

1 EN-TEx ATAC-seq data: downstream analyses

1.1 Move to folder ATAC-seq, and create folders to store bigBed data files and peaks analyses files. Make sure the files are organized in a consistent way as done for ChIP-seq

```
#I moved to the correct folder
cd ~/epigenomics_uvic/ATAC-seq

# Create folders
mkdir -p data/bigBed.files
mkdir -p analyses/peaks.analysis
mkdir -p annotation
```

1.2 Retrieve from a newly generated metadata file ATAC-seq peaks (bigBed narrow, pseudoreplicated peaks, assembly GRCh38) for stomach and sigmoid_colon for the same donor used in the previous sections. Hint: have a look at what we did here. Make sure your md5sum values coincide with the ones provided by ENCODE.

```
# Download metadata file for ATAC-seq on stomach and sigmoid colon (GRCh38, released)
../bin/download.metadata.sh"https://www.encodeproject.org/metadata/?type=Experiment&replicates.
library.biosample.donor.uuid=d370683e-81e7-473f-8475-7716d027849b&status=released&assembly=
GRCh38&biosample_ontology.term_name=sigmoid+colon&biosample_ontology.term_name=stomach
&assay_title=ATAC-seq"

# Explore metadata
head -1 metadata.atac.tsv | awk 'BEGIN{FS=OFS="\t"}{for (i=1;i<=NF;i++) print $i, i}'

grep "bigBed" metadata.atac.tsv | grep "pseudoreplicated" | grep "narrow" |
grep -E "stomach|sigmoid colon"

### Download ATAC-seq peak files (bigBed)
wget -O data/bigBed.files/stomach.bigBed https://www.encodeproject.org/files/ENCFF762IFP
/@@download/ENCFF762IFP.bigBed

wget -O data/bigBed.files/sigmoid_colon.bigBed https://www.encodeproject.org/files/
ENCFF287UHP/@@download/ENCFF287UHP.bigBed

#Order them in an appropriate way
move files to corresponding folder: mv stomach.bigBed sigmoid_colon.bigBed data/bigBed

move to folder: cd data/bigBed

# Check md5sums
```

```
md5sum data/bigBed.files/stomach.bigBed
md5sum data/bigBed.files/sigmoid_colon.bigBed

#The following files were downloaded and verified using md5sum

Stomach:
File accession: ENCFF762IFP
URL: https://www.encodeproject.org/files/ENCFF762IFP/@@download/ENCFF762IFP.bigBed
Verified md5sum: f6a97407b6ba4697108e74451fb3eaf4

Sigmoid Colon:
File accession: ENCFF287UHP
URL: https://www.encodeproject.org/files/ENCFF287UHP/@@download/ENCFF287UHP.bigBed
Verified md5sum: 46f2ae76779da5be7de09b63d5c2ceb9
```

1.3 For each tissue, run an intersection analysis using BEDTools: report 1) the number of peaks that intersect promoter regions, 2) the number of peaks that fall outside gene coordinates (whole gene body, not just the promoter regions). Hint: have a look at what we did here and here.

```
#I went back to the main ATAC-seq directory.

#Created a new folder named "annotation" and moved into it.

#Copied the BED file containing promoter regions of protein-coding genes into this folder.

#Ultimately, calculated how many peaks overlap with the promoter regions.

#Converted bigBed to BED
bigBedToBed data/bigBed.files/stomach.bigBed analyses/peaks.analysis/stomach.bed

bigBedToBed data/bigBed.files/sigmoid_colon.bigBed analyses/peaks.analysis/sigmoid_colon.bed

#Copy annotation files
cp ../ChIP-seq/annotation/gencode.v24.protein.coding.gene.body.bed annotation/
cp ../ChIP-seq/annotation/gencode.v24.protein.coding.non.redundant.TSS.bed annotation/

#Find peaks overlapping promoters (TSS)
bedtools intersect -u \
    -a analyses/peaks.analysis/stomach.bed \
    -b annotation/gencode.v24.protein.coding.non.redundant.TSS.bed \
    > analyses/peaks.analysis/stomach_promoter_peaks.bed

bedtools intersect -u \
    -a analyses/peaks.analysis/sigmoid_colon.bed \
    -b annotation/gencode.v24.protein.coding.non.redundant.TSS.bed \
    > analyses/peaks.analysis/sigmoid_colon_promoter_peaks.bed

#Find peaks outside gene bodies
bedtools intersect -v \
    -a analyses/peaks.analysis/stomach.bed \
```

```
    -b annotation/gencode.v24.protein.coding.gene.body.bed \
    > analyses/peaks.analysis/stomach_non_gene_peaks.bed

bedtools intersect -v \
    -a analyses/peaks.analysis/sigmoid_colon.bed \
    -b annotation/gencode.v24.protein.coding.gene.body.bed \
    > analyses/peaks.analysis/sigmoid_colon_non_gene_peaks.bed
```

In the stomach sample, 44,749 peaks overlapped promoter regions, while 34,537 peaks were located outside annotated gene bodies. In the sigmoid colon sample, we observed 47,871 promoter-intersecting peaks and 37,035 peaks outside gene bodies. These results are consistent with the expected enrichment of chromatin accessibility at promoter regions and distal regulatory elements, reflecting tissue-specific gene regulation.

2 Distal regulatory activity.

2.1 Task 1: Create a folder regulatory_elements inside epigenomics_uvic. This will be the folder where you store all your subsequent results.

```
# Move to  main project directory
cd ~/epigenomics_uvic

# Create the new folder
mkdir p- regulatory_elements
```

2.2 Task 2: Distal regulatory regions are usually found to be flanked by both H3K27ac and H3K4me1. From your starting catalogue of open regions in each tissue, select those that overlap peaks of H3K27ac AND H3K4me1 in the corresponding tissue. You will get a list of candidate distal regulatory elements for each tissue. How many are they?

```
#I copied metadata from ChIP-seq. I integrated ChIP-seq data with my ATAC-seq open
#chromatin peaks to identify distal regulatory elements.

# H3K27ac
grep -F H3K27ac metadata-chip.tsv |\
grep -F "bigBed_narrowPeak" |\
grep -F "pseudoreplicated_peaks" |\
grep -F "GRCh38" |\
awk 'BEGIN{FS=OFS="\t"}{print $1, $11, $23}' |\
sort -k2,2 -k1,1r |\
sort -k2,2 -u > analyses/bigBed.peaks.H3K27ac.ids.txt

# H3K4me1
grep -F H3K4me1 metadata-chip.tsv |\
grep -F "bigBed_narrowPeak" |\
grep -F "pseudoreplicated_peaks" |\
grep -F "GRCh38" |\
```

```
awk 'BEGIN{FS=OFS="\t"}{print $1, $11, $23}' |\
sort -k2,2 -k1,1r |\
sort -k2,2 -u > analyses/bigBed.peaks.H3K4me1.ids.txt

# Download peak files for H3K27ac
cut -f1 analyses/bigBed.peaks.H3K27ac.ids.txt |\
while read filename; do
  wget -P data/bigBed.files
  "https://www.encodeproject.org/files/$filename/@@download/$filename.bigBed"
done

# Download peak files for H3K4me1
cut -f1 analyses/bigBed.peaks.H3K4me1.ids.txt |\
while read filename; do
  wget -P data/bigBed.files
  "https://www.encodeproject.org/files/$filename/@@download/$filename.bigBed"
done

#I created a folder and moved
mkdir -p analyses/bed.files
cd ../ATAC-seq/analysis

# H3K27ac
cut -f1 analyses/bigBed.peaks.H3K27ac.ids.txt |\
while read filename; do
  bigBedToBed data/bigBed.files/"$filename".bigBed analyses/bed.files/"$filename".bed
done

# H3K4me1
cut -f1 analyses/bigBed.peaks.H3K4me1.ids.txt |\
while read filename; do
  bigBedToBed data/bigBed.files/"$filename".bigBed analyses/bed.files/"$filename".bed
done

#Checked them
grep -Ff <(ls analyses/bed.files | sed 's/.bed//') metadata-chip.tsv | cut -f1,11,23

ENCFF872UHN sigmoid_colon   H3K27ac-human
ENCFF977LBD stomach H3K27ac-human
ENCFF724ZOF sigmoid_colon   H3K4me1-human
ENCFF844XRN stomach H3K4me1-human

#Code to find distal peaks
# Stomach distal peaks (outside gene bodies)
bedtools intersect -v \
  -a stomach.bed \
  -b annotation/gencode.v24.protein.coding.gene.body.bed \
  > stomach.distal.atac.peaks.bed

# Sigmoid colon distal peaks
bedtools intersect -v \
  -a sigmoid_colon.bed \
  -b annotation/gencode.v24.protein.coding.gene.body.bed \
```

```
  > sigmoid_colon.distal.atac.peaks.bed

#STOMACH

DISTAL REGULATORY
bedtools intersect \
  -a stomach.distal.atac.peaks.bed \
  -b analyses/bed.files/ENCFF977LBD.bed |\
bedtools intersect \
  -a - \
  -b analyses/bed.files/ENCFF844XRN.bed \
  > analyses/stomach.distal.regulatory.bed

echo "Stomach candidate distal regulatory elements:"
wc -l analyses/stomach.distal.regulatory.bed

root@32701b6d7716:/home/me/epigenomics_uvic/ATAC-seq/analysis
# echo "Stomach candidate distal regulatory elements:"
Stomach candidate distal regulatory elements:
root@32701b6d7716:/home/me/epigenomics_uvic/ATAC-seq/analysis
# wc -l analyses/stomach.distal.regulatory.bed
9034 analyses/stomach.distal.regulatory.bed

#SIGMOID DISTAL
bedtools intersect \
  -a sigmoid_colon.distal.atac.peaks.bed \
  -b analyses/bed.files/ENCFF872UHN.bed |\
bedtools intersect \
  -a - \
  -b analyses/bed.files/ENCFF724ZOF.bed \
  > analyses/sigmoid_colon.distal.regulatory.bed

echo "Sigmoid colon candidate distal regulatory elements:"
wc -l analyses/sigmoid_colon.distal.regulatory.bed

root@32701b6d7716:/home/me/epigenomics_uvic/ATAC-seq
# echo "Sigmoid colon candidate distal regulatory elements:"
Sigmoid colon candidate distal regulatory elements:
root@32701b6d7716:/home/me/epigenomics_uvic/ATAC-seq
# wc -l analyses/sigmoid_colon.distal.regulatory.bed
16492 analyses/sigmoid_colon.distal.regulatory.bed
```

In conclusion, Stomach: 9,034 candidate distal regulatory elements. Sigmoid colon: 16,492 candidate distal regulatory elements.

2.3 Task 3: Focus on regulatory elements that are located on chromosome 1 (hint: to parse a file based on the value of a specific column, have a look at what we did here), and generate a file regulatory.elements.starts.tsv that contains the name of the regulatory region (i.e. the name of the original ATAC-seq peak) and the start (5') coordinate of the region.

```
# Combine both stomach and sigmoid regulatory elements, select only chr1,
#and extract name + 5' start

# Create regulatory.elements.starts.tsv with tissue label and no intermediate files
{
  awk 'BEGIN{FS=OFS="\t"} $1=="chr1" {
      name = $4;
      strand = $6;
      start = (strand == "-") ? $3 : $2;
      print name, start, "stomach";
  }' analyses/stomach.distal.regulatory.bed

  awk 'BEGIN{FS=OFS="\t"} $1=="chr1" {
      name = $4;
      strand = $6;
      start = (strand == "-") ? $3 : $2;
      print name, start, "sigmoid_colon";
  }' analyses/sigmoid_colon.distal.regulatory.bed
} > regulatory.elements.starts.tsv


cut -f3 regulatory.elements.starts.tsv | sort | uniq -c

root@32701b6d7716:/home/me/epigenomics_uvic/ATAC-seq/analysis
# cut -f3 regulatory.elements.starts.tsv | sort | uniq -c
   1743 sigmoid_colon
   1114 stomach
```

2.4 Task 4: Focus on protein-coding genes located on chromosome 1. From the BED file of gene body coordinates that you generated here, prepare a tab-separated file called gene.starts.tsv which will store the name of the gene in the first column, and the start coordinate of the gene on the second column (REMEMBER: for genes located on the minus strand, the start coordinate will be at the 3').

```
#To complete Task 4, I started from the suggested awk command:
awk 'BEGIN{FS=OFS="\t"}{if ($6=="+"){start=$2} else {start=$3}; print $4, start}'

#However, this command processes all genes, regardless of chromosome.
#Since the task specifies to focus on protein-coding genes located on chromosome 1,
#I adapted the command by adding a filter to include only entries on chr1, which uses
#the BED file of protein-coding gene bodies (gencode.v24.protein.coding.gene.body.bed)
#ultimately redirecting the output to a file called gene.starts.tsv.
```

```
awk 'BEGIN{FS=OFS="\t"} $1=="chr1" {
  if ($6 == "+") { start = $2 } else { start = $3 }
  print $4, start
}' annotation/gencode.v24.protein.coding.gene.body.bed > gene.starts.tsv
```

2.5 Task 5: Download or copy this python script inside the epigenomics_uvic/bin folder. This script takes as input two distinct arguments: 1) –input corresponds to the file gene.starts.tsv (i.e. the file you generated in Task 4); 2)start corresponds to the 5' coordinate of a regulatory element.

```
#Through the use of nano function I edited the script as it follows:

#!/usr/bin/env python

#############
# LIBRARIES *
#############

import sys
from optparse import OptionParser

#################
# OPTION PARSING *
#################

parser = OptionParser()
parser.add_option("-i", "--input", dest="input")
parser.add_option("-s", "--start", dest="start")
options, args = parser.parse_args()

open_input = open(options.input)
enhancer_start = int(options.start)

#########
# BEGIN *
#########

x = 1000000  # set maximum distance to 1 Mb
selectedGene = ""  # initialize the gene as empty
selectedGeneStart = 0  # initialize as empty

for line in open_input.readlines():
    gene, y = line.strip().split('\t')  # split the line into two columns
    position = int(y)
    distance = abs(position - enhancer_start)

    if distance < x:
        x = distance
        selectedGene = gene
        selectedGeneStart = position
```

```
print("\t".join([selectedGene, str(selectedGeneStart), str(x)]))

#The results I got are the following:
root@32701b6d7716:/home/me/epigenomics_uvic/ATAC-seq/analysis
# python ../bin/get.distance.py --input gene.starts.tsv --start 980000
ENSG00000187642.9   982093  2093
```

2.6 Task 6. For each regulatory element contained in the file regulatory.elements.starts.tsv, retrieve the closest gene and the distance to the closest gene using the python script you created above.

```
#To perform this task I  added the start position to find the closest gene.
#Lastly, I saved the results in a file, along with the regulatory element,
#tissue and the distance.

cat regulatory.elements.starts.tsv | while read element start tissue; do
  result=$(python ../bin/get.distance.py --input gene.starts.tsv --start $start)
  echo -e "$element\t$start\t$tissue\t$result"
done > regulatoryElements.genes.distances.tsv

#These are the results:
root@32701b6d7716:/home/me/epigenomics_uvic/ATAC-seq/analysis
# column -t regulatoryElements.genes.distances.tsv | head
Peak_25860   1068237    stomach        ENSG00000237330.2   1074307    6070
Peak_68978   1068237    stomach        ENSG00000237330.2   1074307    6070
Peak_19319   1068516    stomach        ENSG00000237330.2   1074307    5791
Peak_24518   1068516    stomach        ENSG00000237330.2   1074307    5791
Peak_24039   1079493    stomach        ENSG00000237330.2   1074307    5186
Peak_25265   1079493    stomach        ENSG00000237330.2   1074307    5186
Peak_38063   1079493    stomach        ENSG00000237330.2   1074307    5186
Peak_32494   1124797    stomach        ENSG00000131591.17  1116361    8436
Peak_106662  1125097    stomach        ENSG00000131591.17  1116361    8736
Peak_106662  1125490    stomach        ENSG00000131591.17  1116361    9129

root@32701b6d7716:/home/me/epigenomics_uvic/ATAC-seq/analysis
# wc -l regulatoryElements.genes.distances.tsv
2857 regulatoryElements.genes.distances.tsv
```

2.7 Task 7: Use R to compute the mean and the median of the distances stored in regulatoryElements.genes.distances.tsv.

```
#I launched R in my environment via R command

#I used this R script
# Load the data
data <- read.table("regulatoryElements.genes.distances.tsv", header = FALSE, sep = "\t")

# Inspect the structure (optional)
```

```
# head(data)

# The distance is in the last column (column 6)
distances <- data[[6]]

# Compute statistics
mean_distance <- mean(distances)
median_distance <- median(distances)

# Print results
cat("Mean distance:", mean_distance, "\n")
cat("Median distance:", median_distance, "\n")
> # Print results
> cat("Mean distance:", mean_distance, "\n")
Mean distance: 63304.1
> cat("Median distance:", median_distance, "\n")
Median distance: 31908
>
```