**# Task 5 ChIP\_seq**

**#Previous**

cut -f-2 analyses/bigBed.peaks.ids.txt |\

while read filename tissue; do

echo "$tissue"

bedtools intersect -a data/bed.files/"$filename".bed -b ../ChIP-seq/annotation/gencode.v24.protein.coding.gene.body.bed -v |\

sort -u -k1,1 -k2,2 -k3,3 > data/bed.files/ATAC\_seq\_peaks\_outside\_genes\_"$tissue".bed

done

**# Task 1: Create a folder *regulatory\_elements* inside *epigenomics\_uvic*.**

**# This will be the folder where you store all your subsequent results.**

mkdir regulatory\_elements

cd regulatory\_elements

**# 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?**

mkdir analyses

**# 1. Search files to download**

for mod in H3K27ac H3K4me1

do

echo $mod

grep -F $mod ../ChIP-seq/metadata.tsv |\

grep -F "bigBed\_narrowPeak" |\

grep -F "pseudoreplicated\_peaks" |\

grep -F "GRCh38" |\

awk 'BEGIN{FS=OFS="\t"}{print $1, $11}' |\

sort -k2,2 -k1,1r |\

sort -k2,2 -u > analyses/bigBed.$mod.peaks.ids.txt

done

**# 2. download bigBed files**

mkdir data

mkdir data/H3K27ac

mkdir data/H3K4me1

for mod in H3K27ac H3K4me1

do

cut -f1 analyses/bigBed.$mod.peaks.ids.txt |\

while read filename; do

echo $filename

wget -P data/$mod "https://www.encodeproject.org/files/$filename/@@download/$filename.bigBed"

done

done

**# 3. test files for integrity**

for mod in H3K27ac H3K4me1

do

echo $mod

../bin/selectRows.sh <(cut -f1 analyses/bigBed.$mod.peaks.ids.txt) ../ChIP-seq/metadata.tsv | cut -f1,46 > data/$mod/md5sum.txt

cat data/$mod/md5sum.txt |\

while read filename original\_md5sum; do

md5sum data/$mod/"$filename".bigBed |\

awk -v filename="$filename" -v original\_md5sum="$original\_md5sum" 'BEGIN{FS=" ";OFS="\t"}{print filename, original\_md5sum, $1}'

done > tmp

mv tmp data/$mod/md5sum.txt

awk '$2!=$3' data/$mod/md5sum.txt

done

**# 4. convert downloads files to appropriate bed files**

for mod in H3K27ac H3K4me1

do

echo $mod

cut -f1 analyses/bigBed.$mod.peaks.ids.txt |\

while read filename; do

bigBedToBed data/$mod/"$filename".bigBed data/$mod/"$filename".bed

done

done

**# 5. Intersect gene peaks with H3K27ac and H3K4me1 peaks**

cut -f-2 analyses/bigBed.H3K27ac.peaks.ids.txt |\

while read filename tissue; do

echo "$tissue"

bedtools intersect -a ../ATAC-seq/data/bed.files/ATAC\_seq\_peaks\_outside\_genes\_"$tissue".bed -b data/H3K27ac/"$filename".bed -u > data/common\_peaks\_H3K27ac\_"$tissue".bed

done

cut -f-2 analyses/bigBed.H3K4me1.peaks.ids.txt |\

while read filename tissue; do

echo "$tissue"

bedtools intersect -a data/common\_peaks\_H3K27ac\_"$tissue".bed -b data/H3K4me1/"$filename".bed -u > data/common\_peaks\_H3K27ac\_H3K4me1\_"$tissue".bed

done

wc -l data/common\_peaks\_H3K27ac\_H3K4me1\_\*.bed

---------------------------------------------------------------------------------------

**Results:**

**# 8627 candidate distal regulatory elements for sigmoid\_colon**

**# 5148 candidate distal regulatory elements for stomach**

**# 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.**

mkdir regulatory\_elements\_starts

for tissue in sigmoid\_colon stomach

do

echo $tissue

grep -w chr1 data/common\_peaks\_H3K27ac\_H3K4me1\_"$tissue".bed | awk 'BEGIN{FS=OFS="\t"}{print $4, $2}' > regulatory\_elements\_starts/regulatory.elements.starts."$tissue".tsv

done

**# 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'). Use the command below as a starting point:**

mkdir gene\_starts

grep -w chr1 ../ChIP-seq/annotation/gencode.v24.protein.coding.gene.body.bed |

awk 'BEGIN{FS=OFS="\t"}{if ($6=="+"){start=$2} else {start=$3}; print $4, start}' > gene\_starts/gene.starts.tsv

**# Task 5: Download or copy this python script inside the epigenomics\_uvic/bin folder.**

**# Have a look at the help page of this script to understand how it works:**

touch ../bin/get.distance.py

nano ../bin/get.distance.py

**# Changes on the python script:**

**# for line in open\_input.readlines(): # for every line in the input file**

**# gene, position = line.strip().split('\t') # here, it splits the line into two columns based on a tab**

**# position = int(position) # definition of a variable named position that keeps the integer corresponding to the start of the gene**

**# distance = abs(position - enhancer\_start) # it estimates the absolute value of the difference between position and enhancer\_start**

**#**

**# if distance < x: # if this absolute value is lower than x**

**# x = distance # assign this distance value to the x**

**# selectedGene = gene # save gene as *selectedGene***

**# selectedGeneStart = position # save position as *selectedGeneStart***

**Have a look at the options of get.distance.py file:**

python ../bin/get.distance.py -h

**# Usage: get.distance.py [options]**

**#**

**# Options:**

**# -h, --help show this help message and exit**

**# -i INPUT, --input=INPUT**

**# -s START, --start=START**

python ../bin/get.distance.py --input gene\_starts/gene.starts.tsv --start 980000

# ENSG00000187642.9 **982093 2093**

**# 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.**

**# Use the command below as a starting point:**

mkdir regulatory\_elements\_distances

for tissue in sigmoid\_colon stomach

do

echo $tissue

cat regulatory\_elements\_starts/regulatory.elements.starts."$tissue".tsv | while read element start; do

python ../bin/get.distance.py --input gene\_starts/gene.starts.tsv --start $start;

done > regulatory\_elements\_distances/regulatoryElements.genes.distances."$tissue".tsv

done

**# Task 7: Use R to compute the mean and the median of the distances stored in**

**# regulatoryElements.genes.distances.tsv.**

colon<-read.csv("regulatory\_elements\_distances/regulatoryElements.genes.distances.sigmoid\_colon.tsv", header=F, sep="\t")

**#Use unlist & as .vector**

dist\_c<-as.vector(unlist(colon[3]))

---------------------------------------------------

**#First result:**

**mean(dist\_c)**

**# 73067.4**

**median(dist\_c)**

**# 36045**

stomach<-read.csv("regulatory\_elements\_distances/regulatoryElements.genes.distances.stomach.tsv", header=F, sep="\t")

**#Use unlist & as .vector**

dist\_s<-as.vector(unlist(stomach[3]))

-------------------------------------------------

**#Second result:**

**mean(dist\_s)**

**# 47013.77**

**median(dist\_s)**

**# 27773.5**