Instructions for the ChIP-seq Data Analysis Class (SciLifeLab NGS course, Uppsala 27-28 January 2016)

Agata Smialowska

January 27, 2016

1 Introduction

REST (NRSF) is a transcriptional repressor that represses neuronal genes in non-neuronal tissues. It is a member of the Kruppel-type zinc finger transcription factor family. It represses transcription by binding a DNA sequence element called the neuron-restrictive silencer element (NRSE). The protein is also found in undifferentiated neuronal progenitor cells and it is thought that this repressor may act as a master negative regular of neurogenesis. In addition, REST has been implicated as tumour suppressor, as the function of REST is lost in breast, colon and small cell lung cancers.

2 Data and Methods

2.1 Data

Data you will use originates from the ENCODE project (www.encodeproject.org). It consists of duplicates of ChIP-seq of a transcription factor REST in several human cell lines and in vitro differentialed neural cells. The ChIP data contains matching input chromatin samples.The accession numbers are listed in the Table 1. Each sample accession number used in this exercise is listed in the Table 2, in the Appendix.

The reads are mapped to the human genome assembly version hg19.

No	Accession	Cell type	Description
1	ENCSR000BMN	HeLa	adenocarcinoma (Homo sapiens, adult 31 year female)
2	ENCSR000BOT	HepG2	hepatocellular carcinoma (Homo sapiens, child 15 year male)
3	ENCSR000BOZ	SK-N-SH	neuroblastoma (Homo sapiens, child 4 year female)
4	ENCSR000BTV	neural	in vitro differentiated (Homo sapiens, embryonic male)

Table 1: ENCODE accession numbers for data sets used in this exercise.

In the interest of time, all steps are performed using a scaled down data set (containing reads mapped to chromosomes 1, 2 and 3). However, for the post peak-calling QC and differential occupancy, the peaks on chromosomes 1, 2 and 3 are selected from the peaks called using the full data set.

2.2 Methods

After mapping the reads to the genome using a preferred short read aligner performing ***ungapped (global)*** alignment (bowtie in this example), the bam files are preprocessed to remove reads which could confound the subsequent analysis. The files you are working with are mapped by the ENCODE consortium, and only the reads with ***one best alignment*** are reported (in some sources these are referred to as "unique alignments" or "uniquely aligned reads"). This means that alignments of reads mapped to multiple locations in the genome are not present in the data. If other mapping strategy is used, such multi-mapping reads need to be removed from the data prior to analyses.

First, as an initial quality control (QC) step, you will compute (i) cross correlation metrics using phantompeakqualtools. These metrics and plots are used to assess the level of enrichment in ChIP-seq, and thus inform about general quality of the data set.

Next, bam files will be preprocessed to remove reads which: (ii) are duplicates or, (iii) map to blacklisted regions ("hyper-chippable" regions). These reads may result from experimental artefacts and their presence may interfere with downstream analyses.

Subsequently, post-alignment QC steps include calculation of (iv) cumulative enrichment using deepTools (bamFingerprint), another method to assess the enrichment in the data set; and (v) sample clustering using deepTools (bamCorrelate), a method to inspect similarities between libraries.

Next, you will generate coverage tracks using deepTools (bamCoverage) (vi); these can be viewed in a genome browser (vii).

Enriched regions aka peaks are identified using MACS2 (viii). After peak calling, you will perform several checks on identified peaks (ix), and perform another round of sample clustering (x).

The last four steps are optional; I recommend trying one or more out if time allows. They include: post-peak calling QC (R script), differential occupancy (R script), peak annotation (R script), and visualisation of the ChIP signal with respect to genomic features (deepTools). People familiar with R can perform these analyses in the R terminal (just type R in the command line; q() to quit R).

Please note that all the methods used in this exercise perform significantly better when used on complete (i.e. non-subsetted) data sets. Their accuracy most often scales with the number of mapped reads in each library, but so does the run time. As a

reference, for some of the steps, plots generated analysing the complete data set are also presented.

Last but not least, we have prepared intermediate files in case some step won't work; these will allow you to progress through the analysis. You will find them in /results.

3 Before you start...

First, you need to book a node on milou. We have reserved half a node for each student during this course. By now, you are probably already familiar with the procedure:

```
salloc -A g2016001 -t 08:00:00 -p core -n 8 --no-shell --reservation=g2016001_201601_27 &
```

NB! Do the node booking only once and make sure you do not have multiple reservations running at the same time, otherwise you will take away resources from other course participants!

To save time and minimise problems, you will use a bash script which will set few variables necessary for this class, and create symbolic links to data. To prepare your Uppmax session for the ChIP-seq class copy the file chipseq_setup.sh from the course director to your home directory:

```
cd ~/glob
cp  /proj/g2016001/labs/chipseq_agata/chipseq_setup.sh ./

source chipseq_setup.sh
```

You should see a directory named "chipseq":

```
cd chipseq
cd analysis
```

...and you are ready to start.

NB! Please do not change files in /proj/g2016001/labs/chipseq_agata/

Any change in that directory changes the content for everyone!

4 Quality Control, part 1: cross correlation

You will calculate cross correlation for REST ChIP-seq in HeLa cells. This section is performed using data subsetted to chromosomes 1, 2 and 3.

```
mkdir xcor
cd xcor

module load phantompeakqualtools/1.1

run_spp.R -c=../../data/ENCFF000PED.chr123.bam \
 -savp=hela1_xcor.pdf -out=xcor_metrics_hela.txt

module unload phantompeakqualtools/1.1
```

This step takes a few minutes, and phantompeakqualtools prints messages as it progresses through different stages of the analysis.

Inspect the resulting file xcor_metrics_hela.txt.

The metrics file is tabulated, and the fields are:

COL1: Filename

COL2: numReads: effective sequencing depth i.e. total number of mapped reads in input file

COL3: estFragLen: comma separated strand cross-correlation peak(s) in decreasing order of correlation. In almost all cases, the top (first) value in the list represents the predominant fragment length.

COL4: corr_estFragLen: comma separated strand cross-correlation value(s) in decreasing order (col3 follows the same order)

COL5: phantomPeak: Read length/phantom peak strand shift

COL6: corr_phantomPeak: Correlation value at phantom peak

COL7: argmin_corr: strand shift at which cross-correlation is lowest

COL8: min_corr: minimum value of cross-correlation

COL9: Normalized strand cross-correlation coefficient (NSC) = COL4 / COL8

COL10: Relative strand cross-correlation coefficient (RSC) = (COL4 - COL8) / (COL6 - COL8)

COL11: QualityTag: Quality tag based on thresholded RSC (codes: -2:veryLow; -1:Low; 0:Medium; 1:High; 2:veryHigh)

The columns to pay attention to are:

`COL3: gives the fragment length as estimated from the data;`

`COL9: NSC; NSC>1.1 (higher values indicate more enrichment; 1 = no enrichment)`

`COL10: RSC; RSC>0.8 (0 = no signal; <1 low quality ChIP; >1 high enrichment`

```
COL11: Quality tag based on thresholded RSC
(codes: -2:veryLow,-1:Low,0:Medium,1:High; 2:veryHigh)
```

For comparison, the cross correlation metrics computed for the entire data set using non-subsetted data are available by:

```
cat ../../results/xcor/xcor_metrics_REST.txt
```

In addition to inspecting the metrics file, it is always recommended to inspect the plots, as the shape of the cross correlation is more informative than just numbers.

Compare the plot hela1_xcor.pdf (cross correlation of the first replicate of REST ChIP in HeLa cells, using sub-setted data) with cross correlation computed using the non-subsetted data set presented on figures 1 - 3. Compare to the ChIP using the same antibody performed in HepG2 cells (figures 4 - 6).

If you have enabled X-forwarding on your local computer, this command opens pdf files directly on Uppmax:

```
evince hela1_xcor.pdf &
```

If this above command does not work for you (there may be problems, depending on the configuration of your local computer), you can copy the pdf files to your local computer and open using your preferred pdf viewer. Using terminal, cd to the desired destination directory and substitute the text in CAPS with appropriate names:

```
scp UPPMAX_LOGIN@milou.uppmax.uu.se:~/glob/chipseq/analysis/FOLDER_NAME/*pdf ./
```

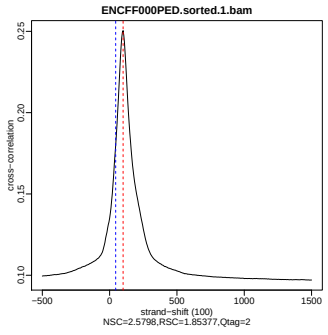

Figure 1: Cross correlation plot, HeLa, REST ChIP, replicate 1, QScore:2

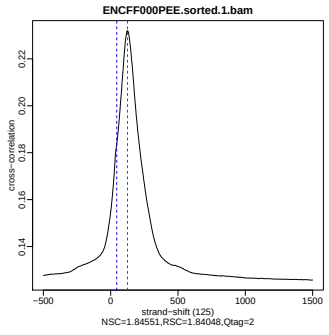

Figure 2: Cross correlation plot, HeLa, REST ChIP, replicate 2, QScore:2

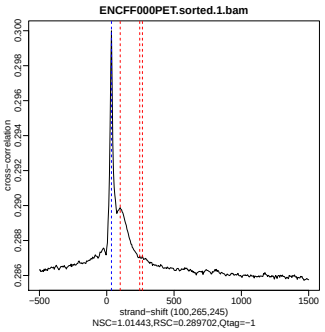

Figure 3: Cross correlation plot, HeLa, input, QScore:-1

How would you rate these particular data sets? Are all libraries of good quality?

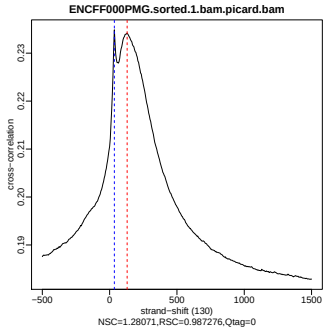

Figure 4: Cross correlation plot, HepG2, REST ChIP, replicate 1, QScore:0

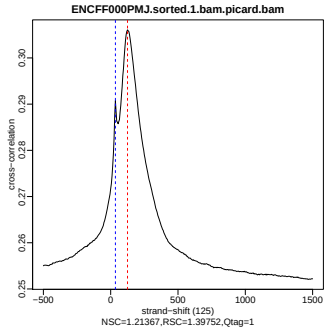

Figure 5: Cross correlation plot, HepG2, REST ChIP, replicate 2, QScore:1

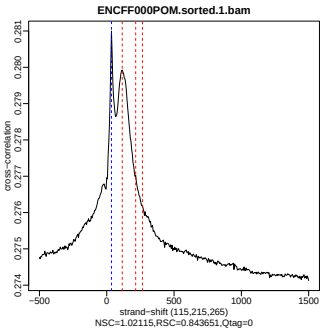

Figure 6: Cross correlation plot, HepG2, input, QScore:0

5 Alignment Preprocessing

This section is performed using data subsetted to chromosomes 1, 2 and 3. First, duplicated reads are marked using MarkDuplicates and removed from bam files using samtools. Marking as "duplicates" is based on their alignment location, not sequence.

```
cd ..
mkdir bam_preproc
cd bam_preproc

module load samtools/1.1
module load java/sun_jdk1.8.0_40
module load picard/1.141

java  -Xmx64G -jar $PICARD_HOME/picard.jar MarkDuplicates \
I=../../data/ENCFF000PED.chr123.bam O=ENCFF000PED.chr123.dup.bam \
M=dedup_metrics.txt VALIDATION_STRINGENCY=LENIENT \
REMOVE_DUPLICATES=false ASSUME_SORTED=true

samtools view -h -b -F 1804 -o ENCFF000PED.chr123.dup.rmdup.bam ENCFF000PED.chr123.dup.bam
```

Second, reads mapped to blacklisted regions

(`https://sites.google.com/site/anshulkundaje/projects/blacklists`) are removed.

```
bamutils filter ENCFF000PED.chr123.dup.rmdup.bam \
ENCFF000PED.chr123.dup.rmdup.filt.bam \
-excludebed ../../hg19/wgEncodeDacMapabilityConsensusExcludable.bed nostrand
```

Finally, the preprocessed bam files are sorted and indexed:

```
samtools sort -T sort_tempdir -o ENCFF000PED.chr123.dup.rmdup.filt.sort.bam \
ENCFF000PED.chr123.dup.rmdup.filt.bam

samtools index ENCFF000PED.chr123.dup.rmdup.filt.sort.bam

module unload samtools/1.1
module unload java/sun_jdk1.8.0_40
module unload picard/1.141
```

6 Quality Control, part 2: cumulative enrichment

The cumulative enrichment (aka bamFingerprint) will be computed for the HeLa REST ChIP and corresponding input samples. This section is performed using data subsetted to chromosomes 1, 2 and 3. You will use the preprocessed file you have just created, and two other libraries from the same data set, prepared earlier.

```
module load deepTools/1.5.11

bamFingerprint --bamfiles ENCFF000PED.chr123.dup.rmdup.filt.sort.bam  \
../../data/bam/hela/ENCFF000PEE.preproc.chr123.bam \
../../data/bam/hela/ENCFF000PET.preproc.chr123.bam \
--fragmentLength 120  \
--labels HeLa_rep1 HeLa_rep2 HeLa_input \
--plotFile HeLa.fingerprint.pdf
```

Ignore the warnings: "The index file is older than the data file". The index is older because of the order the files were copied to the course directory.

Inspect the plot; does it indicate good data quality? For comparison, similar plots generated for other samples used in this exercise are presented in figures 7 and 8.

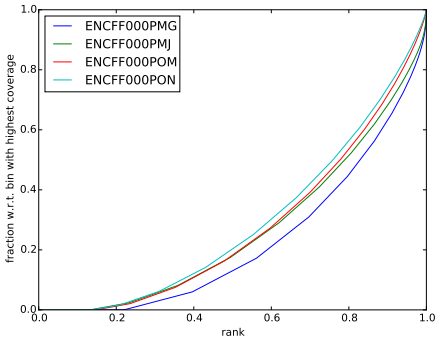

Figure 7: Cumulative enrichment for REST ChIP and corresponding inputs in HepG2 cells

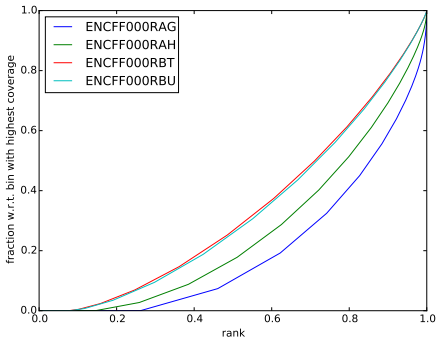

Figure 8: Cumulative enrichment for REST ChIP and corresponding inputs in SK-N-SH cells

Are the cumulative enrichment plots in agreement with the cross-correlation metrics computed earlier?

7 Sample clustering

To assess overall similarity between libraries from different samples and data sets, you will compute sample clustering heatmaps using bamCorrelate from deepTools. This section is performed using data subsetted to chromosomes 1, 2 and 3.

First, to avoid very long paths in the command line, link the directories with the preprocessed bam files:

```
ln -s /proj/g2016001/labs/chipseq_agata/data/bam/hela ./hela
ln -s /proj/g2016001/labs/chipseq_agata/data/bam/hepg2 ./hepg2
ln -s /proj/g2016001/labs/chipseq_agata/data/bam/sknsh ./sknsh
ln -s /proj/g2016001/labs/chipseq_agata/data/bam/neural ./neural
```

```
bamCorrelate bins --bamfiles hela/ENCFF000PED.preproc.chr123.bam \
hela/ENCFF000PEE.preproc.chr123.bam hela/ENCFF000PET.preproc.chr123.bam \
hepg2/ENCFF000PMG.preproc.chr123.bam hepg2/ENCFF000PMJ.preproc.chr123.bam \
hepg2/ENCFF000POM.preproc.chr123.bam hepg2/ENCFF000PON.preproc.chr123.bam \
neural/ENCFF000OWM.preproc.chr123.bam neural/ENCFF000OWQ.preproc.chr123.bam \
neural/ENCFF000OXB.preproc.chr123.bam neural/ENCFF000OXE.preproc.chr123.bam \
sknsh/ENCFF000RAG.preproc.chr123.bam sknsh/ENCFF000RAH.preproc.chr123.bam \
sknsh/ENCFF000RBT.preproc.chr123.bam sknsh/ENCFF000RBU.preproc.chr123.bam \
--binSize 5000 --corMethod spearman \
--labels hela_1 hela_2 hela_i hepg2_1 hepg2_2 hepg2_i1 hepg2_i2 neural_1 \
neural_2 neural_i1 neural_i2 sknsh_1 sknsh_2 sknsh_i1 sknsh_i2 \
-o correlation_bins5k.pdf \
--outFileCorMatrix corr_matrix_bins5k.txt --numberOfProcessors "max"
```

Inspect the resulting pdf.

8 Computing read coverage

In this section you will compute the read coverage normalised to 1x coverage using tool bamCoverage from deepTools. This procedure is useful for comparing libraries sequenced to a different depth when viewing them in a genome browser such as IGV. This section is performed using data subsetted to chromosomes 1, 2 and 3; hence the effective genome size used is 690470000 (6.9e8) (for hg19 the effective genome size is 2451960000, 2.45e9 (see `http://www.nature.com/nbt/journal/v27/n1/fig_tab/nbt.1518_T1.html`)).

```
bamCoverage --bam ENCFF000PED.chr123.dup.rmdup.filt.sort.bam \
```

```
--outFileName ENCFF000PED.bedgraph \
--normalizeTo1x 690470000 --fragmentLength 120 --binSize 50 \
--missingDataAsZero yes --outFileFormat bedgraph --ignoreForNormalization "chrX, chrY, chrM"

module unload deepTools/1.5.11
```

9 Peak Calling

You will identify peaks in the ChIPseq data using MACS2. MACS2 is one of the most popular peak callers, and it performs very well on data sets with good enrichment of transcription factors ChIP. Peaks should be called on each replicate separately (do not pool the replicates).

As before, this section is performed using data subsetted to chromosomes 1, 2 and 3; hence the effective genome size used is 690470000 (6.9e8). Again, to avoid long paths in the command line, you will link the necessary files. You will call peaks only in one ChIP-seq library; the rest of the work is already done, and the peaks are in the directory /results/peaks_macs (for HeLa cells only) and /results/peaks_bed.

```
cd ..
mkdir peak_calling


ln -s /proj/g2016001/labs/chipseq_agata/data/bam/hela/ENCFF000PED.preproc.chr123.bam peak_calling/ENCFF000PED.preproc.bam
ln -s /proj/g2016001/labs/chipseq_agata/data/bam/hela/ENCFF000PET.preproc.chr123.bam peak_calling/ENCFF000PET.preproc.bam


cd peak_calling
```

9.1 Peak Calling

There are several parameters which affect peak calling itself, as well as result reporting. It is important to understand them to modify the command to the needs of your data set.

Meaning of the parameters:

-t, -c, -f, -n denote treatment, control, file format and output file names, respectively

-g is the genome size, in this case it is already encoded in MACS: -g hs = -g 2.7e9; -g mm = -g 1.87e9; -g ce = -g 9e7; -g dm = -g 1.2e8.

-q 0.01 is the q value (FDR) cutoff for reporting peaks

In our case, you use -g 6.9e8 because for peak calling you use data from chromosomes 1, 2 and 3 only (total size is 6.9e8 bp).

You will see the progress of the analysis printed in the terminal (MACS prints messages as it progresses through different stages of the process). This step will take more than

10 minutes.

```
module load MACS/2.1.0

macs2 callpeak -t ENCFF000PED.preproc.bam -c ENCFF000PET.preproc.bam \
-f BAM -g 6.9e8 -n hela_1_REST.chr123.macs2 -q 0.01

module unload MACS/2.1.0
module unload python/2.7.6
```

The output of a MACS2 run consists of several files. You can inspect their contents using

```
head -n 50 filename
```

You will use the narrowPeak files in the subsequent parts. These files are in bed format (`https://genome.ucsc.edu/FAQ/FAQformat.html#format1`), which is one of the popular file format used in genomics. It is used to store information on genomic ranges (such as ChIP-seq peaks, gene annotation, TSS etc). Bed files can be also used for visualisation in genome browsers, including the popular UCSC Genome Browser `https://genome.ucsc.edu/cgi-bin/hgTracks`.

Files used in this step are derived from the *.narrowPeak files by selecting relevant columns, for example:

```
cut -f 1-3 hela_1_REST.chr123.macs2_peaks.narrowPeak >hela_1_chr123_peaks.bed
```

Peaks detected on chromosomes 1, 2 and 3, are present in directory /results/peaks_bed. These peaks were detected using the non-subsetted data, and therefore there may be differences between the peaks present in the prepared hela_1_peaks.bed file compared to the peaks you have detected. Use these pre-made peak bed files instead of the file you have just created. You can check how many peaks were detected in each library by listing number of lines in each file:

```
wc -l ../../results/peaks_bed/*.bed
```

By checking for overlaps in the peak lists from different libraries you can detect peaks common to both libraries. This will give you an idea which peaks are reproducible between replicates. Select two replicates of the same condition, for example:

```
module load  BEDTools/2.25.0

cp /proj/g2016001/labs/chipseq_agata/results/peaks_bed/*.bed ./

bedtools intersect -a hela_1_peaks.bed -b hela_2_peaks.bed -f 0.50 -r > peaks_hela.bed

wc -l peaks_hela.bed
```

You will need to create files with peaks common to replicates for the cell types you wish to compare.

Bedtools is a suite of utilities developed for manipulation of bed files; `http://bedtools.`

readthedocs.org/en/latest/.

In the command used here, the arguments are

-a, -b : two files to be intersected

-f 0.50 : fraction of the overlap between features in each file to be reported as an overlap

-r : reciprocal overlap fraction required

You can inspect which peaks were reproducibly found in two different cell lines, for example:

```
bedtools intersect -a peaks_hepg2.bed -b peaks_hela.bed -f 0.50 -r > peaks_hepg2_hela.bed

wc -l peaks_hepg2_hela.bed

module unload  BEDTools/2.25.0
```

Now you will generate a merged list of all peaks detected in the experiment, which will be needed in the next step. In this list, all overlapping intervals are merged into one, and a list of non-overlapping intervals is created. You will use another suite of tools for manipulation of bed files, BEDOPS http://bedops.readthedocs.org/en/latest/. Files used in this step are derived from the *.narrowPeak files by selecting relevant columns, for example:

These files are already prepared and present in the directory peak_calling (your current directory, if you follow the directory structure in the exercise).

```
module load BEDOPS/2.4.3

bedops -m hela_1_peaks.bed hela_2_peaks.bed hepg2_1_peaks.bed hepg2_2_peaks.bed \
neural_1_peaks.bed neural_2_peaks.bed sknsh_1_peaks.bed sknsh_2_peaks.bed \
>rest_peaks.bed

wc -l rest_peaks.bed

module unload BEDOPS/2.4.3
```

In case things go wrong at this stage, you can find the merged list of all peaks in the /results directory. Link the file to your current directory

```
ls -s ../../results/rest_peaks.bed ./rest_peaks.bed
```

9.2 Quality Control After Peak Calling

This section is performed using data subsetted to chromosomes 1, 2 and 3. For the subsequent parts, you will use peaks which were called in the non-subsetted data set, and peaks located on chromosomes 1, 2 and 3 were selected.

9.2.1 Clustering of libraries based on the reads mapped in peaks

You will use bamCorrelate, the same tool as for clustering of samples based on reads mapped genome-wide. This time you will provide a list of intervals where reads are counted; this list is a merged list of all peaks detected in all libraries. You will begin by making a new directory and copying the data there.

```
cd ..
mkdir plots
cd plots

cp /proj/g2016001/labs/chipseq_agata/data/bam/hela ./helacp
cp /proj/g2016001/labs/chipseq_agata/data/bam/hepg2 ./hepg2
cp /proj/g2016001/labs/chipseq_agata/data/bam/sknsh ./sknsh
cp /proj/g2016001/labs/chipseq_agata/data/bam/neural ./neural
cp ../../results/rest_peaks.bed ./rest_peaks.bed

module load deepTools/1.5.11

bamCorrelate BED-file --BED rest_peaks.bed \
 --bamfiles hela/ENCFF000PED.preproc.chr123.bam \
hela/ENCFF000PEE.preproc.chr123.bam hela/ENCFF000PET.preproc.chr123.bam \
hepg2/ENCFF000PMG.preproc.chr123.bam hepg2/ENCFF000PMJ.preproc.chr123.bam \
hepg2/ENCFF000POM.preproc.chr123.bam hepg2/ENCFF000PON.preproc.chr123.bam \
neural/ENCFF000OWM.preproc.chr123.bam neural/ENCFF000OWQ.preproc.chr123.bam \
neural/ENCFF000OXB.preproc.chr123.bam neural/ENCFF000OXE.preproc.chr123.bam \
sknsh/ENCFF000RAG.preproc.chr123.bam sknsh/ENCFF000RAH.preproc.chr123.bam \
sknsh/ENCFF000RBT.preproc.chr123.bam sknsh/ENCFF000RBU.preproc.chr123.bam \
 --corMethod spearman \
 --labels hela_1 hela_2 hela_i hepg2_2 hepg2_1 hepg2_i1 hepg2_i2 neural_1 \
neural_2 neural_i1 neural_i2 sknsh_1 sknsh_2 sknsh_i1 sknsh_i2 \
-o correlation_peaks.pdf --outFileCorMatrix corr_matrix_peaks.txt --numberOfProcessors "max"

module unload deepTools/1.5.11
```

In this heatmap different to the previously generated one? Why?

9.3 Visualisation mapped reads, coverage profiles and peaks in a genome browser

This part of the exercise is best performed locally on your own computer. It requires installation of Interactive Genome Browser, IGV (`https://www.broadinstitute.org/igv/`). You can install IGV on your computer and view the following files (remember setting the reference genome to hg19):

(REST ChIP in HeLa cells, replicate 1)

`../analysis/bam_preproc/ENCFF000PED.preproc.bam`

`../analysis/bam_preproc/ENCFF000PED.bedgraph`

`../analysis/peak_calling/hela_1_REST.chr123.macs2_peaks.narrowPeak`

(input in HeLa cells)

`../../data/bam/hela/ENCFF000PET.preproc.chr123.bam`

`../../results/coverage/ENCFF000PET.preproc.coverage.norm1x.bedgraph`

In IGV, in menu "File": "Load from file", select files you want to visualise. Load the selected tracks. Click on chromosome 1, 2 or 3, zoom in and explore! Go to one of the locations you found interesting (for example, of the REST binding peaks detected in both HeLa samples), available in the file peaks_hela.bed, which you generated earlier.

Is the read distribution in the peaks (bam file tracks) consistent with the expected bimodal distribution? Which genes are associated with the detected peaks?

10 Additional analyses

The following steps can be performed either by executing the scripts provided with the exercise from the command line, or by typing in all commands in an R console. The latter is recommended for people who have some previous exposure to the R environment. Post-peak calling QC is performed using the R / Bioconductor package ChIPQC, and many steps are redundant with the already performed ones; it is an alternative to the already presented workflow (a much slower alternative, as R is not designed to handle such large data sets); this step is optional. However, this procedure also creates an object used for the Differential Occupancy and Peak Annotation sections.

Finally, you can visualise the distribution of the ChIP signal with respect to genomic features using deepTools.

10.1 Alternative Quality Control Workflow in R

You will start in directory /analysis/R. The file REST_samples.txt contains information on files location, and the paths are given with respect to /analysis/R; if you choose to start in another directory, please modify the paths in REST_samples.txt. This script takes a while to run.

```
cd ../analysis/R
Rscript chipqc.R
```

10.2 Differential Occupancy and Peak Annotation in R

Please note that normally three biological replicates are required for statistical analysis of factor occupancy. There are only two replicates each in the ENCODE data sets used in this class - hence you use duplicates for demonstration sake.

```
cd ../analysis/R
Rscript diffbind_annot.R
```

People familiar with R can modify and execute the commands from within the R terminal, selecting different contrasts of interest, for example.

10.3 Signal visualisation using deepTools

You will visualise ChIP signal in relation to annotated TSS on chromosomes 1, 2 and 3. A description of all visualisation options is given at `https://github.com/fidelram/deepTools/wiki/Visualizations`. Create a separate directory in /analysis; cd to it. Check if all the paths to create links are correct for the location of your directory.

First you will compute the matrix of values using computeMatrix. This program takes bigWig files (`https://genome.ucsc.edu/goldenpath/help/bigWig.html`) as input; you will need to convert bedgraph to bigWig using UCSC utilities:

```
cp ../../hg19/chrom.sizes.hg19  chrom.sizes.hg19
cp ../bam_preproc/ENCFF000PED.bedgraph ENCFF000PED.bedgraph

module load ucsc-utilities/v287

bedGraphToBigWig ENCFF000PED.bedgraph chrom.sizes.hg19 hela_1.bw

module unload ucsc-utilities/v287
```

You are now ready to compute the matrix of scores for visualisation. You will need a bed file with positions of TSS; you copy it to your current directory.

```
cp /proj/g2016001/labs/chipseq_agata/hg19/refGene_hg19_TSS_chr123.bed  refGene_hg19_TSS_chr123.bed

module load deepTools/1.5.11

computeMatrix reference-point --regionsFileName refGene_hg19_TSS_chr123.bed \
 --scoreFileName hela_1.bw --outFileName hela_1_matrix.out \
 --outFileNameData hela_1_profile.tab --outFileSortedRegions hela_1_sorted_regions.bed \
 --referencePoint TSS --beforeRegionStartLength 2000 --afterRegionStartLength 2000 \
 --sortRegions descend --minThreshold 0 --numberOfProcessors "max" --skipZeros
```

Having the matrix of scores ready, you can now plot the binding profile around TSS and the heatmap:

```
heatmapper --matrixFile hela_1_matrix.out --outFileName hela.vis1.pdf --colorMap RdPu

module unload deepTools/1.5.11
```

11 Concluding remarks

The workflow presented in this exercise is similar to a typical one used for analysis of ChIP-seq data. There are more types of analyses you can do, which were not discussed here. One typical task is to identify short sequence motifs enriched in the regions bound by the assayed factor (peaks). There are several tools available, and I recommend testing at least two tools for your data.

Homer: `http://homer.salk.edu/homer/`

GEM: `http://groups.csail.mit.edu/cgs/gem/`

RSAT: `http://floresta.eead.csic.es/rsat/peak-motifs_form.cgi`

MEME: `http://meme-suite.org/`

12 Appendix

No	Accession	Cell type	Replicate	Input
1	ENCFF000PED	HeLa	1	ENCFF000PET
2	ENCFF000PEE	HeLa	2	ENCFF000PET
3	ENCFF000PMG	HepG2	1	ENCFF000POM
4	ENCFF000PMJ	HepG2	2	ENCFF000PON
5	ENCFF000OWQ	neural	1	ENCFF000OXB
6	ENCFF000OWM	neural	2	ENCFF000OXE
7	ENCFF000RAG	SK-N-SH	1	ENCFF000RBT
8	ENCFF000RAH	SK-N-SH	2	ENCFF000RBU

Table 2: ENCODE accession numbers for samples used in this exercise.