ECCB2020Tutorial Documentation

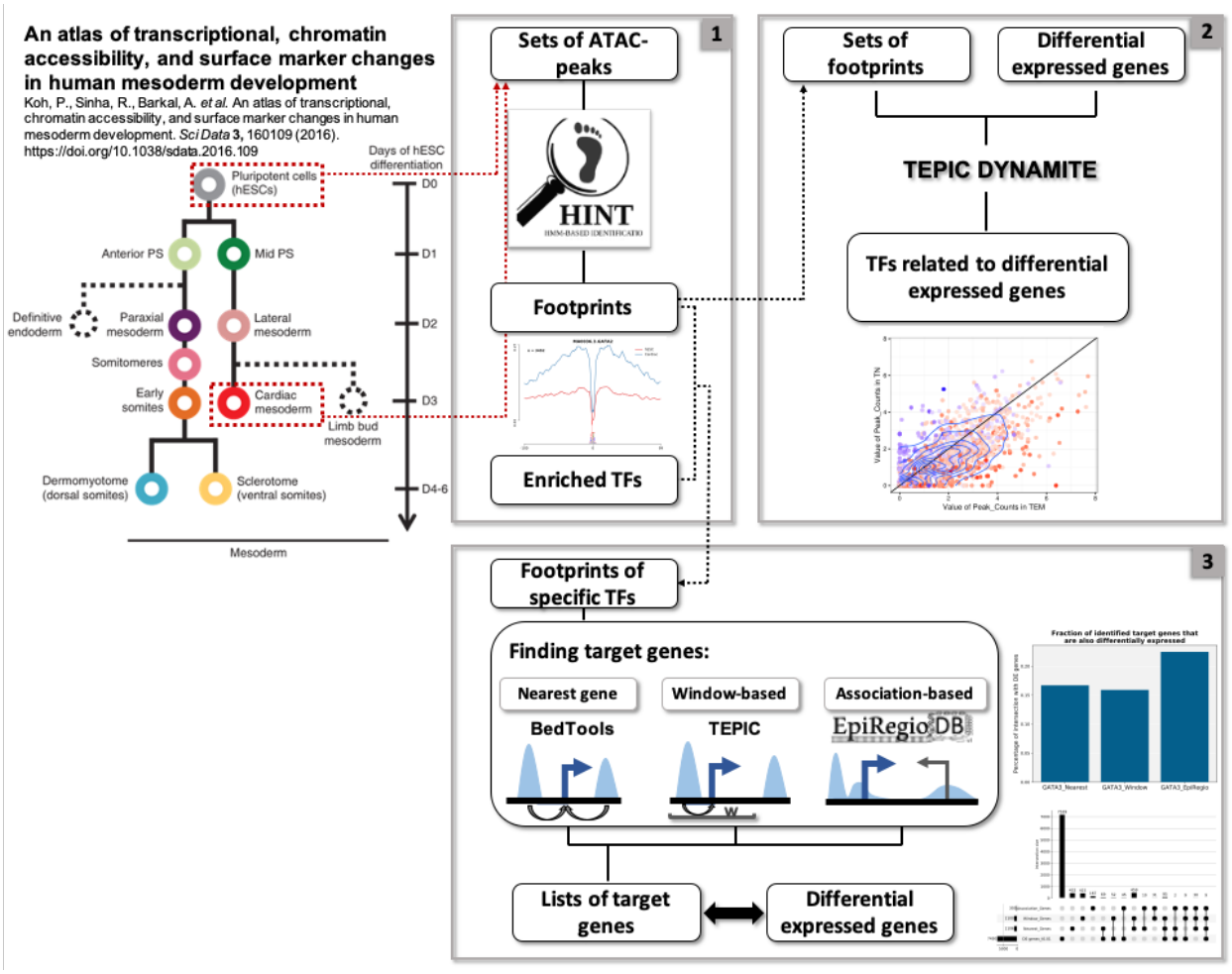
Release 1.0

Zhijian Li, Nina Baumgarten, Dennis Hecker, Sivarajan Karunanith

Sep 14, 2020

Contents

If you are taking part in our tutorial, then you find the information here that you need. The documentation is split up into 4 parts. Make sure to read and follow the **What you need part** with all necessary **Installations** before you join us virtually. Below you can see an illustration of the workflow we planned for you.



PDF of the flowchart

Contents:

CHAPTER 1

What you need

You need to have a **laptop/desktop** with the following software installed (see detailed instructions below)

- **Docker**
  - Having a docker user account is not necessary for the purposes of the tutorial.
- IGV

1.1 Download docker image

Please **download the docker image** we have prepared for the purposes of this tutorial **before** the tutorial. The size of docker image is **approximately 20GB**.

You can use the following command:

```
docker run -it -P --name epigenomics -v ~/container-data:/data sivarajank/epigenomics_
↪tutorial:latest
```

Command for Windows users:

```
docker run -it -P --name epigenomics -v C:\Users\*username*\Documents\container-data:/
↪data sivarajank/epigenomics_tutorial:latest
```

Please do **not** forget to change your username.

The *name* parameter assigns a name to your docker image. If you used the above command it is *epigenomics (–name)*. If, for some reason, you exit or abort your docker image, you can use the following commands to re-login to the docker container. :

```
docker start epigenomics
docker attach epigenomics
```

To get out of the docker container, you can simply type *exit*.

If, for some reason, we ask you to update the docker container, you need to exit the docker container and execute:

```
docker pull sivarajank/epigenomics_tutorial:latest
```

After that you can execute the docker start and attach commands mentioned earlier.

1.2 How to access data inside the docker container

We recommend to access the data inside the container using the proposed method, for the purposes of this tutorial.

In the aforementioned *docker run* command you might have noticed the *-v ~/container-data:/data* parameter. This parameter enables you to access any data you have inside the **/data** directory of your container from your local machine at **~/container-data/**. The **/data** directory is mirrored to your local **~/container-data/** folder, meaning that everything you do (remove, edit, add etc.) in you local folder **~/container-data/** will also be changed in the **/data** folder and vice-versa. This has the advantage that we can create files like plots inside of Docker, write them to the **/data** directory and you can easily access them outside of Docker by looking into the **~/container-data/** folder.

We have shipped some preprocessed data as part of our container which can be found at **/root/EpigenomeAnalysisTutorial-2020**. Once you are inside the docker container, please run the following command so that you can access this data from your local machine to view in IGV or view the plots :

```
mv /root/EpigenomeAnalysisTutorial-2020 /data/
cd /data
```

Now, you can find the folder EpigenomeAnalysisTutorial-2020 in your local machine at **~/container-data/** folder. We will run all commands under this folder during the course of the tutorial. Please take care to keep this directory as you current directory.

```
cd EpigenomeAnalysisTutorial-2020
```

1.3 How to remove all the containers and images after the tutorial

If you want to **remove all docker content (images/containers)** from your system, you can run the following commands:

```
docker rm epigenomics
docker image prune -a
```

CHAPTER 2

Practical I - Detection of Open Chromatin regions & Footprint calling & Transcription factor prediction

In the first practical, we will perform basic analysis of ATAC-seq, footprint analysis and motif matching with HINT to identify cell specific binding sites from open chromatin data (ATAC-seq). Beforehand, be sure you have installed Docker and pulled the container which includes all tools and data for this tutorial.

2.1 Example data

We will analyse here chromatin (ATAC-seq) and gene expression data during human mesoderm development from Koh PW et al 2016. Here, we focus on regulatory changes between hESC cells and Cardiac mesoderm cells. For each condition, we have two ATAC-seq libraries corresponding to biological duplicates.

2.2 Step 1: Quality check, aligment and peak calling of ATAC-seq data

A first step in the analysis of ATAC-seq data are the so called low level analysis, which includes read trimming, quality check, aligment and peak calling. We have used for this the nfcore/atacseq, a rather complete pipeline for ATAC-seq data. Due to the computational time to run such pipeline, we have pre-computed results and we provide all important files under *./data/nf_core_atacseq*.

Among other things, the pipeline will generate important files, which will be used during this tutorial:

- quality check (QC) statistics: *./data/nf_core_atacseq/multiqc/narrowPeak/multiqc_report.html*
- alignment files: *./data/nf_core_atacseq/*
- genomic profiles (big wig): *./data/nf_core_atacseq/bigwig*
- peak calling results: *./data/nf_core_atacseq/macs/narrowPeak*
- differential peak calling results: *./data/nf_core_atacseq/macs/narrowPeak/consensus/deseq2/CardiacvshESC/*
- IGV session: *./data/nf_core_atacseq/IGV/igv_session.xml*

First, you can inspect the QC statistics, just go to *~/container-data/EpigenomeAnalysisTutorial-2020/data/nf_core_atacseq/multiqc/narrowPeak* and double click *multiqc_report.html*. Do the atac-seq libraries have any quality issue before and after read trimming?

Next, you can use then IGV to vizualise ATAC-seq signals and peaks particular loci. Open the previously mentioned IGV session and take a look at cardiac related genes, i.e. GATA4 or GATA6, or stem cell related genes, i.e. POU5F1 (OCT4).

The differential peaks file combined all peaks and here we can split it as hESC and Cardiac specific peaks by:

```
mkdir -p ./results/session1/diff_peaks
awk '{if ($5 < 0) print $0}' ./data/nf_core_atacseq/macs/narrowPeak/consensus/deseq2/
↪CardiacvshESC/CardiacvshESC.mRp.clN.deseq2.FDR0.05.results.bed > ./results/session1/
↪diff_peaks/hESC.bed
awk '{if ($5 > 0) print $0}' ./data/nf_core_atacseq/macs/narrowPeak/consensus/deseq2/
↪CardiacvshESC/CardiacvshESC.mRp.clN.deseq2.FDR0.05.results.bed > ./results/session1/
↪diff_peaks/Cardiac.bed
```

The above commands will check the sign in 5th column and output as hESC specific peak if values are negative.

If you are interested in running nf-core at a latter stage, you can check the script here.

2.3 Step 2: Footprint calling

Next, we will use HINT to find genomic regions (footprints) with cell specific TF binding sites. For this, HINT requires (1) a sorted bam file containing the aligned reads from the sequencing library (DNase-,ATAC- or histone ChIP-seq) (2) and a bed file including peaks detected in the same sequencing library provided in (1). These peak regions are used by HINT to reduce the search space.

HINT footprinting analysis is performed for each cell type independently (hESC and Cardiac) and do not consider replicate information. For this we will use bam files contatining reads from all replicates as well as condition specific consensus peaks.

We will only consider peaks inside chromosome 21 so that the whole analysis can be done in 30 minutes.

1. First, create a folder for results:

```
mkdir -p ./results/session1/hint_chr21
```

2. Select peaks from chromosome 21 (this step is only performed to reduce computing time for this practical exercise).

```
mkdir -p ./results/session1/hint_chr21/peaks
awk '$1 ~ /^chr(21)$/' ./data/nf_core_atacseq/macs/narrowPeak/hESC.mRp.clN_peaks.
↪narrowPeak > ./results/session1/hint_chr21/peaks/hESC.bed
awk '$1 ~ /^chr(21)$/' ./data/nf_core_atacseq/macs/narrowPeak/Cardiac.mRp.clN_peaks.
↪narrowPeak > ./results/session1/hint_chr21/peaks/Cardiac.bed
```

3. Next, we can execute HINT twice to find footprints specific to hESC and cardiac cells. This can be done as:

```
mkdir -p ./results/session1/hint_chr21/footprints
rgt-hint footprinting --atac-seq --paired-end --organism=hg38 --output-location=./
↪results/session1/hint_chr21/footprints --output-prefix=hESC ./data/nf_core_atacseq/
↪hESC.mRp.clN.sorted.bam ./results/session1/hint_chr21/peaks/hESC.bed
rgt-hint footprinting --atac-seq --paired-end --organism=hg38 --output-location=./
↪results/session1/hint_chr21/footprints --output-prefix=Cardiac ./data/nf_core_
↪atacseq/Cardiac.mRp.clN.sorted.bam ./results/session1/hint_chr21/peaks/Cardiac.bed
```

This will generate an output file, i.e `./results/session1/hint_chr21/footprints/hESC.bed`, containing the genomic locations of the footprints. HINT also produces a file with ending ".info", which has general statistics from the analysis as no. of footprints, total number of reads and so on. Input arguments indicate important information to HINT as genome version (–organism), chromatin protocol (–atac-seq) and type of read configuration (–paired-end). You can check more information on HINT here .

You can use the head command to check the information contained in footprints:

```
head ./results/session1/hint_chr21/footprints/hESC.bed
```

The first 3 columns define a genomic coordination and the 4th column contains name for each footprint. The 5th column contains the number of reads around predicted footprint and can be used as metric for ordering footprints, i.e. the more reads the more likely it is associated to an active binding site.

4. HINT performs footprinting analysis by considering reads at each genomic position after signal normalization and cleveage bias correction. You need to perform an extra command to generate such signals in order to visualize them in a genome browser:

```
mkdir -p ./results/session1/hint_chr21/tracks
rgt-hint tracks --bc --bigWig --organism=hg38 --output-location=./results/session1/
↪hint_chr21/tracks --output-prefix=hESC ./data/nf_core_atacseq/hESC.mRp.clN.sorted.
↪bam ./results/session1/hint_chr21/peaks/hESC.bed
rgt-hint tracks --bc --bigWig --organism=hg38 --output-location=./results/session1/
↪hint_chr21/tracks --output-prefix=Cardiac ./data/nf_core_atacseq/Cardiac.mRp.clN.
↪sorted.bam ./results/session1/hint_chr21/peaks/Cardiac.bed
```

You can load the newly generated bigwig files and fooptrints with IGV together with the signals and peaks detected by nf-core.

2.4 Step 3: TF binding site prediction

An important question when doing footprint analysis is to evaluate which TF motifs overlap with footprints and evaluate the ATAC-seq profiles around these motifs. RGT suite also offers a tool for finding motif predicted binding sites (MPBSs).

Execute the following commands to do motif matching inside footprints for chromosome 21:

```
mkdir -p ./results/session1/hint_chr21/motifmatching
rgt-motifanalysis matching --organism=hg38 --output-location=./results/session1/hint_
↪chr21/motifmatching --input-files ./results/session1/hint_chr21/footprints/hESC.bed␣
↪./results/session1/hint_chr21/footprints/Cardiac.bed
```

The above commands will generate bed files (i.e. Cardiac_mpbs.bed) containing MPBSs overlapping with distinct footprint regions. The 4th column contains the motif name and the 5th column the bit-score of the motif matching. Higher bit-score indicates higher agreement of the motif with the DNA sequence. HINT uses Jaspar database as default for motifs, but it allows users to use other databases or to define custom databases as well.

2.5 Step 4: Average footprint profiles and differential activity analysis

Finally, we use HINT to generate average ATAC-seq profiles around MPBSs. This analysis allows us to inspect the chromatin accessibility around the binding sites of a particular factor and indicates the TF activitiy, i.e. higher accessibility and clear footprints indicates higher TF activity. Moreover, by comparing the profiles from two ATAC-seq libraries (i.s. hESC vs Cardiac cells), we can get insights on changes in transcription factors with increase in activity (or binding) in two cells. For this, execute the following commands:

```
mkdir -p ./results/session1/hint_chr21/diff_footprints
rgt-hint differential --organism=hg38 --nc 4 --bc --no-lineplots --mpbs-files=./
↪results/session1/hint_chr21/motifmatching/hESC_mpbs.bed,./results/session1/hint_
↪chr21/motifmatching/Cardiac_mpbs.bed --reads-files=./data/nf_core_atacseq/hESC.mRp.
↪clN.sorted.bam,./data/nf_core_atacseq/Cardiac.mRp.clN.sorted.bam --conditions=hESC,
↪Cardiac --output-location=./results/session1/hint_chr21/diff_footprints
```

This command will read the motif matching files generated by step 3 and BAM files which contain the sequencing reads to perform the comparison. The option –nc allows parallel execution of the job and should be specified accordingly. Note that here we specify –bc to use the bias-corrected signal (currently only ATAC-seq is supported). Again, to reduce the running time, we here specify –no-lineplots to not generate the line plots, otherwise it will take too much time. You **should not** use it in your analysis after the tutorial.

The above analyses are based on chromosome 21 and the results are likely to be underpowered, we therefore provide the complete results using all chromsomes in *./results/session1/hint*. The script for this analysis is found here here.

The results include a txt file **differential_statistics.txt** under *./results/session1/hint/diff_footprints* and it contains the transcription factor (TF) activity dynamics between hESC and Cardiac. HINT performs a statistical test to detect TFs with a significant increase or decrease in activity. In addition, a folder called **Lineplots** can be found, which contains the ATAC-seq profile for each of the motifs found in the mpbs bed files.

Next, we use a R script to make a nicer visualization of the TF activity score:

```
Rscript scripts/session1/plot_diff.R -i ./results/session1/hint/diff_footprints/
↪differential_statistics.txt -o ./results/session1/hint/diff_footprints
```

The script will generate a divergent bar plot under *./results/session1/hint/diff_footprints* and two text files which include either Cardiac or hESC specific TFs. Note that it only considers TFs with significant change in activity (p-value < 0.05) and at least 1,000 binding sites for TF. Results rank several GATA TFs, which are well known to be related to cardiac cells, with increase in TF activity, while the well known ES cells factors SOX2:POU5F1 (OCT4) have the second highest decrease in TF activity.

You can check on the folder **Lineplots** for the average cleavage profiles of these factors and their corresponding DNA binding preference.

You should compare the motifs/profiles of Gata factors. Are they similar to one another? One caveat of sequence based analysis is that we might predict several TFs, which have a similar motif, equally.

Finally, we will filter the motif matching results to only consider TFs enriched in a respective condition. You can do this with the following command:

```
mkdir -p ./results/session1/hint/diff_motifmatching
grep -f ./results/session1/hint/diff_footprints/Cardiac.txt ./results/session1/hint/
↪motifmatching/Cardiac_mpbs.bed > ./results/session1/hint/diff_motifmatching/Cardiac_
↪mpbs.bed
grep -f ./results/session1/hint/diff_footprints/hESC.txt ./results/session1/hint/
↪motifmatching/hESC_mpbs.bed > ./results/session1/hint/diff_motifmatching/hESC_mpbs.
↪bed
```

You can then open these files in IGV and inspect footprints and motif hits close to relevant genes as GATA6. Are you able to find any motif close to a gene? You can also zoom out of your IGV browser and check for potential enhancer regions. Also, you should zoom in the open chromatin region to see how the base pair signal provided by HINT-ATAC looks like.

CHAPTER 3

Practical II - Linking regions to genes and integration with gene expression data

In the last step of practical 1, we already identified enriched TFs, which overlap with the predicted footprint. In this part, we are going to infer TFs that might be related to gene expression differences between hESC and cardiac mesoderms. Therefore, we first link the footprints to potential target genes using TEPIC, and then apply DYNAMITE. The tool uses a logistic regression classifier to identify key regulators, which might explain changes in gene expression.

3.1 Step 1: Extracting TF motifs of TFs expressed in the cell types of interest

Ensure that you are in the directory

```
/data/EpigenomeAnalysisTutorial-2020/
```

and create a directory where we can write our results to:

```
mkdir results/session2/
```

DYNAMITE requires as input a list of known TF binding motifs. In this step, we want to extract the TF binding motifs of TFs which are expressed in pluripotent cells (hESCs), cardiac mesoderms (CM) or in both cell types. Therefore, we provide the script *extractPSEMsOfExpressedTFs.py*, which expects as input:

- gene expression values (TPM) from hESCs (precomputed),
- gene expression values (TPM) of CMs (precomputed),
- a file containing all available TF binding motifs as Position specific energy matrices (PSEMs),
- output file,
- a mapping from ensembl IDs to gene names and
- TPM threshold value. If the expression value of a TF is higher than the TPM threshold value, we consider this TF as expressed.

All required input files are located in the data directory (/data/EpigenomeAnalysisTutorial-2020/data/) of the docker container.

As mentioned before, the RNA-seq data is taken from An atlas of transcriptional, chromatin accessibility, and surface marker changes in human mesoderm development. We performed the quantificantion of the RNA-seq data using Salmon. Please run the script using the following command:

```
python scripts/session2/extractPSEMsOfExpressedTFs.py data/meanTPM_hESC.txt  data/
↪meanTPM_CM.txt  data/PSEM_JASPAR2020.txt results/session2/PSEMs_JASPAR2020_TPM_0.5.
↪txt data/ensemblID_GeneName.txt 0.5
```

3.2 Step 2: Intersect the footprint from HINT with differentially ATAC-peaks

As input regions for *DYNAMITE* we are going to use the footprint computed with *HINT*, which we intersect with the differential ATAC-peak. The remaining footprint are regions where we expect binding sites of key TFs and where we observe different chromatin accessibility between hESC and cardiac mesoderms.

Before we can perform the intersection step, we first have to extract the first 5 columns of the footprint output file from hint to create a file which can be handled by bedtools. Therefore we use awk. Please run

```
awk '{print $1 "\t" $2 "\t" $3 "\t" $4 "\t" $5}' results/session1/hint/footprints/
↪hESC.bed  > results/session2/footprints_hESC.bed
awk '{print $1 "\t" $2 "\t" $3 "\t" $4 "\t" $5}' results/session1/hint/footprints/
↪Cardiac.bed  > results/session2/footprints_CM.bed
```

To intersect the footprint of the hESC and the cardiac mesoderms with the differentially ATAC-peaks, we use bedtools intersect command. Two files are required as input. For each region in the first file (indicated by -a), an overlap with the second file (-b) is searched for.

```
bedtools intersect -a  results/session2/footprints_hESC.bed -b data/nf_core_atacseq/
↪macs/narrowPeak/consensus/deseq2/CardiacvshESC/CardiacvshESC.mRp.clN.deseq2.FDR0.05.
↪results.bed  > results/session2/footprints_DiffPeaks_hESC.bed
bedtools intersect -a  results/session2/footprints_CM.bed -b data/nf_core_atacseq/
↪macs/narrowPeak/consensus/deseq2/CardiacvshESC/CardiacvshESC.mRp.clN.deseq2.FDR0.05.
↪results.bed  > results/session2/footprints_DiffPeaks_CM.bed
```

Since the footprints are rather short, we need to extend the region by some base pairs, such that all regions are longer than the longest TF binding motif (>21bp). This is necessary to probably compute the TF binding affinity per footprint region during the *DYNAMITE* analysis. Please run the following two commands:

```
awk '{print $1 "\t" $2 -10 "\t" $3 +10 "\t" $4 "\t" $5}' results/session2/footprints_
↪DiffPeaks_CM.bed  >results/session2/footprints_DiffPeaks_CM_extended.bed
awk '{print $1 "\t" $2 -10 "\t" $3 +10 "\t" $4 "\t" $5}' results/session2/footprints_
↪DiffPeaks_hESC.bed >results/session2/footprints_DiffPeaks_hESC_extended.bed
```

3.3 Step 3: Deriving candidate transcriptional regulators using *DYNAMITE*

During a *DYNAMITE* analysis, two main computational tasks are undertaken:

1. We calculate TF binding affinities for TFs expressed in hESC, cardiac mesoderms or both cell types and aggregate those to gene-TF scores using *TEPIC*. TF affinities are a quantitative measure of TF binding to a distinct genomic region.
2. A logistic regression classifier is learned that uses changes in TF gene scores between two samples to predict which genes are up/down- regulated between them. Investigating the features of the model allows the inference of potentially interesting regulators that are correlated to the observed expression changes.

Please check the documentation for details on the method.

We provide a script that automatically performs steps (1) and (2) as well as necessary data processing and formatting steps (See DYNAMITE documentation for details). All files used in this step are available in `/data/EpigenomeAnalysisTutorial-2020/data/`.

Note that we precomputed the differential gene expression estimates using DESeq2. Computing those is neither part of the actual tutorial nor of the *DYNAMITE* workflow.

1. Ensure that you are in the directory `/data/EpigenomeAnalysisTutorial-2020/`, otherwise *cd* to that directory.

2. Generate an output folder for the resulting files:

```
mkdir results/session2/DYNAMITE/
```

3. To run the *DYNAMITE* script go to the *DYNAMITE* folder in the *TEPIC* repository.

```
cd  /root/TEPIC-2.2/MachineLearningPipelines/DYNAMITE/
```

We provide a configuration files for the *DYNAMITE* analyses:

1. DYNAMITE-hESCvsCM-Top800DEGs.cfg

The configuration files list all parameters that are needed for a run of *DYNAMITE*. To help you customize these files for later usage, we explain the essential parameters here:

- open_regions_Group1: One or more files containing candidate transcription factor binding sites for samples belonging to group 1
- open_regions_Group2: One or more files containing candidate transcription factor binding sites for samples belonging to group 2
- differential_Gene_Expression_Data: Differential gene expression data denoted with log2 fold changes
- outputDirectory: Directory to write the results to
- referenceGenome: Path to the reference genome that should be used
- chrPrefix: Flag indicating whether the reference genome uses a chr prefix
- pwm: Path to the PSEMs that should be used
- cores_TEPIC: Number of cores that are used in the TEPIC analysis
- geneAnnotation: Gene annotation file that should be used
- window: Size of the window around a genes TSS that is screened for TF binding sites
- decay: Flag indicating whether TEPIC should be using exponential decay to downweight far away regions while computing gene-TF scores
- peakFeatures: Flag indicating whether TEPIC should compute features based on peaks, e.g. peak count, peak length, or signal intensity within a peak

In the scope of the tutorial, you do not have to change any of those. A full description of all parameters is provided here.

4. Run the individual pairwise comparisons for LSK vs B:

```
bash runDYNAMITE.sh  DYNAMITE-hESCvsCM-Top800DEGs.cfg
```

The result of the analysis will be stored in `/data/EpigenomeAnalysisTutorial-2020/results/session2/DYNAMITE/`. There are three subfolders for each comparison:

1. Affinities
2. IntegratedData
3. Learning_Results

The folder *Affinities* contains TF affinities calculated in the provided regions for both groups, gene TF scores for both groups, and a metadata file that lists all settings used for the TF annotation with *TEPIC* (subfolders *group1* and *group2*). The subfolder *mean* contains the mean gene TF scores computed for group1 and group2. This is needed if you analyze more than one biological replicate per group. The folder *ratio* contains the gene TF score ratios computed between the gene TF scores of group1 and group2.

The folder *IntegratedData* encloses matrices that are composed of (1) gene TF score ratios and (2) a measure of differential gene expression. In the folder *Log2* the differential gene expression is represented as the log2 expression ratio between group1 and group2. In the folder *Binary*, the differential gene expression is shown in a binary way. Here, a 1 means a gene is upregulated in group 1 compared to group 2, whereas a 0 means it is down-regulated in group1. The binary format is used as input for the classification.

The folder *Learning_Results* comprises the results of the logistic regression classifier. The following files should be produced if all R dependencies are available:

1. Performance_overview.txt
2. Confusion-Matrix_<1..6>_Integrated_Data_For_Classification.txt
3. Regression_Coefficients_Cross_Validation_Integrated_Data_For_Classification.txt
4. Regression_Coefficients_Entire_Data_Set_Integrated_Data_For_Classification.txt
5. Performance_Barplots.pdf
6. Regression_Coefficients_Cross_Validation_Heatmap_Integrated_Data_For_Classification.svg
7. Regression_Coefficients_Entire_Data_SetIntegrated_Data_For_Classification.pdf
8. Misclassification_Lambda_<1..6>_Integrated_Data_For_Classification.svg

The file *Performance_overview.txt* contains accuracy on Test and Training data sets as well as F1 measures. These values are visualized in *Performance_Barplots.pdf*. As the name suggests, the files *Confusion-Matrix_<1..6>_Integrated_Data_For_Classification.txt* contain the confusion matrix computed on the test data sets. They show model performance by reporting True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN) in the following layout:

Observed/Predicted	Positive	Negative
Positive	TP	FN
Negative	FP	TN

The heatmap *Regression_Coefficients_Cross_Validation_Heatmap_Integrated_Data_For_Classification.svg* shows the regression coefficients of all selected features in the outer cross validation. This is very well suited to find features that are stably selected in all outer cross validation folds. The raw data used to generate the figure is stored in *Regression_Coefficients_Cross_Validation_Integrated_Data_For_Classification.txt*. The stronger a regression coefficient, the more important it is in the model.

In addition to the heatmap showing the regression coefficients during the outer cross validation, we also show the regression coefficients learned on the full data set: *Regression_Coefficients_Entire_Data_SetIntegrated_Data_For_Classification.pdf* and *Regression_Coefficients_Entire_Data_Set_Integrated_Data_For_Classification.txt*.

The figures *Misclassification_Lambda_<1..6>_Integrated_Data_For_Classification.svg* are of technical nature. They show the relationship between the misclassification error and the lambda parameter of the logistic regression function.

CHAPTER 4

Practical III - Target gene identification of candidate regions

In the previous steps we looked at footprints of TF and how we can evaluate their difference in activity in between hESC and cardiac mesoderm. This enables us to select TFs of interest and further explore their role in the regulatory machinery that drives the differentiation from hESC to cardiac mesoderm cells. In this step we want to look into different approaches for identifying target genes of genomic regions with regulatory potential. We will use the nearest gene, window-based and association-based approach and compare their results. In the following commands we will use GATA6 as a placeholder, replace it with the TF you are interested in.

4.1 Step 1: Fetching candidate regions

Ensure that you are in the correct directory:

```
cd /data/EpigenomeAnalysisTutorial-2020/
```

First of all, we need a directory where we can write our results to:

```
mkdir results/session3
```

As we do not have any candidate regions yet, we take the regions where HINT annotated a binding site of our selected TF and intersect them with differential ATAC-peaks. This will leave us with regions where we expect a binding of our TF and where the cell types differ in their chromatin accessibility. Thus, we can assume that these regions play a role in the differences in gene expression. The differential peaks were already called by the nf-core preprocessing pipeline. Execute the following command to receive a bed-file with our candidate regions:

```
python3 scripts/session3/FootprintTFFilter.py -m results/session1/hint/motifmatching/
↪Cardiac_mpbs.bed -p data/nf_core_atacseq/macs/narrowPeak/consensus/deseq2/
↪CardiacvshESC/CardiacvshESC.mRp.clN.deseq2.FDR0.05.results.bed -tf GATA6 -o results/
↪session3/GATA6_Targets
```

- **-m**: path to the motif hits, exchange the file name to hESC_mpbs.bed when looking at a TF that scores higher in hESC
- **-p**: path to the differential ATAC-peaks

- **-tf**: name of the TF of interest
- **-o**: Folder to create for the ouput file

A new folder will be created under *results/session3/GATA6_Targets*. It will contain the file *GATA6_Hits.bed*. This is our candidate region file which we will need for the next steps.

4.2 Step 2.1: Target genes - Nearest gene

For linking the nearest gene to a region we make use of the BedTools command closest, or more precisely the Python implementation of it. For each candidate region we will find the closest TSS. We created a file with the 5' TSS of genes in which *closest* will search in. To get the list with the IDs of all closest genes execute:

```
python3 scripts/session3/NearestGeneFinder.py -f results/session3/GATA6_Targets/GATA6_
↪Hits.bed -t data/5TSS_GRCh38p13.txt -o results/session3/GATA6_Targets/Nearest_Genes.
↪txt
```

- **-f**: path to the bed-file of the candidate regions
- **-t**: path to a file with the 5' TSS of genes
- **-o**: output path

You will find the file *Nearest_Genes.txt* in the *results/session3/GATA6_Targets* folder.

4.3 Step 2.2: Target genes - Window-based approach

The window-based approach looks in a defined window around the TSS of a gene and all candidate regions that are located inside of this window are linked to this gene. Call:

```
python3 scripts/session3/WindowGenesFinder.py -f results/session3/GATA6_Targets/GATA6_
↪Hits.bed -t data/5TSS_GRCh38p13.txt -w 50000 -o results/session3/GATA6_Targets/
↪Window_Genes.txt
```

- **-f**: path to the bed-file of the candidate regions
- **-t**: path to a file with the 5' TSS of genes
- **-w**: window size
- **-o**: output path

The script creates intervals of size **-w** centered around the TSS of each gene and then intersects them with our candidate regions. You will find the identified target genes in *results/session3/GATA6_Targets/Window_Genes.txt*.

4.4 Step 2.3: Target genes - Association-based approach

Both of the other methods are based on coordinates. Although they can yield good results in some cases, they are not able to capture long-ranged enhancer-gene interactions. Association-based methods can overcome this issue by combining data like ATAC-seq or DNase-seq for annotation of regulatory elements (REMs)/enhancers with gene expression data. We will make use of the webserver EpiRegio, which incorporates the results of the tool STITCHIT. STITCHIT interprets differences in DNase-signal to explain changes in gene expression among samples of different cell and tissue types. We will use EpiRegio's *Region Query* which will return all annotated regulatory elements and their target genes that overlap with our candidate regions. As required overlap we choose 50%, meaning that at least

half of the length of our candidate region has to overlap with a REM. But instead of using the website (feel free to try it out as well), we will call EpiRegio's REST API via the Python package Requests. Requests allows to make HTTP queries and we can directly continue working with the results. Call the following script:

```
python3 scripts/session3/EpiRegio_Request.py -f results/session3/GATA6_Targets/GATA6_
↪Hits.bed -ov 50 -o results/session3/GATA6_Targets/Association_Genes.txt
```

- **-f**: path to the bed-file with the candidate regions
- **-ov**: overlap as percentage of the length of the candidate regions
- **-o**: output path

For more details on STITCHIT have a look at the preprint.The publication on EpiRegio can be found here.

4.5 Step 3: Intersecting the identified target genes

Now we have three lists of target genes for our candidate regions from different approaches. To compare them, we will create an Upset plot, displaying the intersection with the list of differentially expressed genes which were called by DESeq2 (FDR 0.01). To create the plot use the command:

```
python3 scripts/session3/UpSetPlot_DEGenes.py -f results/session3/GATA6_Targets/
↪Nearest_Genes.txt results/session3/GATA6_Targets/Window_Genes.txt results/session3/
↪GATA6_Targets/Association_Genes.txt -g data/DESeq2_result_file_CM_hESC.tabular -t 0.
↪01 -s 0 -o results/session3/GATA6_Targets/
```

- **-f**: files of gene lists from the different approaches, separated by whitespace
- **-g**: path to the result file of DESeq2
- **-t**: threshold for the adjusted p-value
- **-s**: whether to split the DE genes into negative and positive change (1) or not (0)
- **-o**: ouput path for the files

In addition to the Upset plot, the script will also create a bar plot which depicts the percentage of target genes that are differentially expressed (DE) for all approaches. Further, you will find four new gene ID files. For each approach we filter the target genes for differentially expressed genes and write them into a new file (*... DEGenes_intersection*). The fourth file */Users/dennis/Dev/ECCB20Tutorial/GATA2_TargetGenes/ApproachesMerged_DEGenes_intersection.txt* merges the target genes of all approaches and filters for the DE genes. These files can be used to paste the IDs to functional enrichment analysis tools like gProfiler.

4.6 Step 4: All steps in one

All of the steps above can also be performed by calling the script *TF_to_UpSet_series.sh*:

```
bash ./scripts/session3/TF_to_UpSet_series.sh -m results/session1/hint/motifmatching/
↪Cardiac_mpbs.bed -t "GATA3 GATA6"
```

- **-m**: path to the motif hits, exchange the file name to hESC_mpbs.bed when looking at a TF that scores higher in hESC
- **-t**: TF(s) of interest

This will call all scripts needed one after another, create the output folder and write the files into it. It is adapted to the folder structure of our Docker image, so be sure to edit all paths when you want to call it in a different environment. Like in the example, you can call the script with multiple TFs, don't forget the quotation marks when doing so.