UNIVERSITÁ DEGLI STUDI DI SALERNO

DOTTORATO IN MANAGEMENT & INFORMATION TECHNOLOGY

CURRICULUM: INFORMATION SECURITY & INNOVATION SYSTEMS

COORDINATORE: Ch.mo. Prof. Antonelli Valerio

Ciclo XVII N.S.

Novel tools for reproducible Next Generation Sequencing data analysis and integration

Relatori

Ch.mo. Prof. Tagliaferri Roberto

Ch.mo. Prof. Angelini Claudia

Candidato

Righelli Dario

Matr. 8800800010

ANNO ACCADEMICO 2017/2018

How to reach a goal?
Without haste but without rest
Goethe

Add acknowledgements here

Write your abstract here

Contents

Chapter 1

Introduction

1.1 Biological Background

1.2 Sequencing Techniques

1.2.1 RNA-Seq

1.2.2 Atac-Seq

1.3 Computational Aspects

Chapter 2

TiCoRSe - Time Course RNA-Seq data analysis

2.1 Introduction

2.1.1 Time Course RNA-Seq

2.2 Methods

2.2.1 General Approach

2.2.2 Time Course Methods

2.2.3 Other Methods

2.2.4 Additional Features

2.3 Results

Chapter 3

DEScan2 - Differential Enriched Scan 2

Epigenetic, as shown in introduction (cite), is a pretty wide and complex field, and the sequencing technology to adopt depends on the biological question under investigation.

Some studies Koberstein2018, Auerbach2009 demonstrated the importance of genomewide chromatin accessibility of a broad spectrum of chromatin phenomena activation using sequencing techniques as *Atac-Seq*, *Sono-Seq*, etc. Even if there are some methods for the analysis of these omic data types, there still is a lack of them, in particular for an emerging omic as *Atac-Seq*.

To address this lack, we decided to create a useful instrument for analysing chromatine regions accessibility data (such as *Atac-Seq*, *Sono-Seq*). Very often the biological questions to be answered, as for the RNA-Seq, need the comparison of two or more different biological conditions. Starting from a set of already published Koberstein2018 scripts, we designed Differential Enriched Scan 2 (DEScan2), a software for helping the analysis of chromatin accession sequencing data.

3.1 Introduction

The DEScan2 is an R Ihaka1996 tool developed for detecting open chromatin regions signal in order to facilitate the differential enrichment of genomic regions between two or more biological conditions.

The package has been implemented using Bioconductor Gentleman2004 data structures and methods, and it is available on Bioconductor since version 3.7.

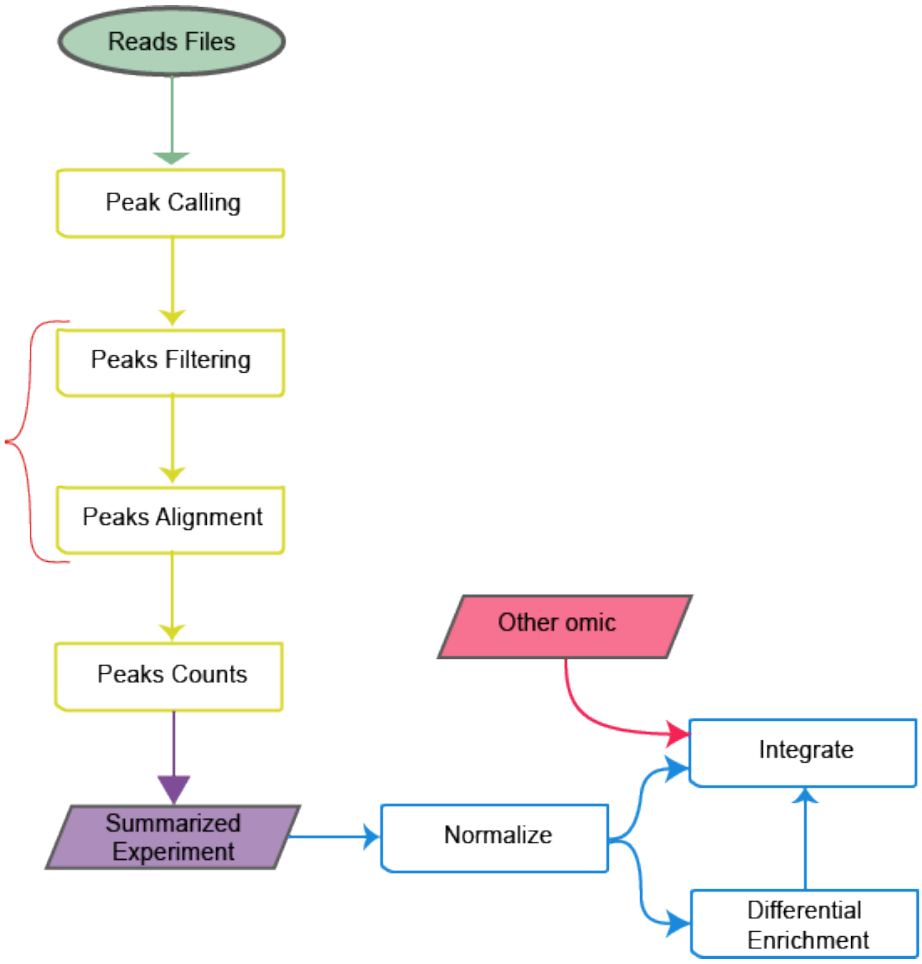

Figure 3.1.1: A differential enrichement flow representation. DEScan2 steps are highlighed in yellow.

The tool is organized in three main steps. A peak caller, which is a standard moving scan window that compares the counts within a sliding window, to the counts in a larger region outside the window. It uses a Maximum Likelihood Estimator on a Poisson Distribution, providing a final score for each detected

peak.

The filtering step is aimed to determine if a peak is a "true peak" on the basis of its replicability in other samples. This step is based on a double user-defined threshold, one on the peak's scores and one on the number of samples.

Finally, the third step produces a counts matrix where each column represents a sample and each row a peak. The value of each cell is the number of reads for the peak in the sample.

The so produced counts matrix, as illustrated in the figure 3.1.1, is useful both for doing differential enrichment between multiple conditions and for integrating the epigenomic data with other -omic data types.

3.2 Methods

The package is organized in three main steps, the peak caller in section 3.2.1, the filtering and alignment of the peaks in section 3.2.2 and the peak counting described in section 3.2.3.

It offers some additional features described in 3.2.4.

3.2.1 Peak Caller

The Peak Caller (`findPeaks` method) takes as input a set of alignment files (BAM Li2009 or BED format) with the code of the reference genome (i.e. *mm10* for Mus Musculus version 10) and several additional parameters, useful for the peak detection setup.

The alignement data are stored as GenomicRangesList Lawrence2013, where each element represents a file. In order to facilitate the parallelization of the computations over the chromosomes, the list is re-arranged as a list of chromosomes of GenomicRangesList, where each element represents a file containing just the GenomicRanges of the specific chromosome (see section 3.2.4.

On this data structure the algorithm firstly divides each chromosome as bins of `binSize` parameter length (default is 50bp) and then computes the reads coverage on the bins with moving scan windows, spanning from `minWin` to `maxWin`

parameters of `binSize` interval.

In order to be able to catch small and spread peaks the algorithm computes the coverage also using windows of two different lengths, that can be defined with `minCompWinWidth` and `maxCompWinWidth` (defaults are 5000 and 10000) parameters.

The so produced coverages are useful to compute a score for each detected region, applying a Maximum Likelihood Extimator (MLE) on the coverages between the sliding windows, assuming a Poisson distribution.

[PUT THE POISSON DISTRIBUTION AND THE LIKELIHOOD]

3.2.2 Peak Filtering and Alignment

In order to filter out false positives peaks, we designed a filtering method (`finalRegions`) based on two different thresholds. A first threshold on the peaks score and a second threshold on the number of samples.

The filtering step is designed to take as input a list of peaks as GenomicRangesList, where each element represents a chromosome. This is the data structure produced by the peak caller, but, we also developed a method to load peaks produced also by other software like MACS Zhang2008, as described in section 3.2.4.

Firstly, using the threshold on the peak's score (`zThreshold` parameter), the method filters out the peaks with a score lower than the user-defined threshold value.

Then it extends a 200bp window in both directions of the detected region, computing the overlaps between the samples using the `findOverlapsOfPeaks` method (with `connectedPeaks` parameter set as `merge`), defined by the ChIPpeakAnno Zhu2010 Bioconductor package.

Based on this idea, the filtering step is developed to filter out those peaks not present in at least a user-defined (`minCarriers` parameter) number of samples. In the light of this, the user can decide the minimum number of samples where each peak has to be detected. We suggest to set the threshold as a mutiple of the number of replicates of the conditions.

3.2.3 Peak Counts

The counting step (`countFinalRegions` method) is designed to take a *GenomicRanges* data structure as input, where for each peak additional features, as the score and the number of samples, are saved. Moreover, it requires also the path of the BAM/BED files where the reads are stored, in order to quantify the peaks given as input.

For each region the method counts the number of reads present in each sample. In so doing, it produces as result a matrix of the counts, where the rows and the columns represent, respectively, the regions and the samples.

In order to keep trace of all information associated to the regions, it produces a *SummarizedExperiment* SummExp data structure, giving the possibility to retrieve the *GenomicRanges* peaks associated data structure and the count matrix, respectively, with `rowRanges` and `assays` method.

The choice to produce a count matrix is guided by the versatility of this data structure, useful not only for the differential enrichment of the regions between multiple conditions, but also for integrating the epigenomic data with other -omics.

3.2.4 Additional Features

The package offers additional features for loading data (i.e. peaks) resulting from other sources, and for manipulating *GenomicRanges* data structure.

The method `readFilesAsGRangesList` takes as input a directory with BAM or BED data, to load in *GenomicRangesList* format. This data structure is useful to store genomic information, as peaks or mapped reads, produced by other software like *MACS2* or *STAR* and, in case of peaks, it is necessary during the DEScan2 filtering step. Additionally to `fileType` (BAM, BED, BED.zip) parameter specification it requires the genome code to use during the file processing. Moreover, when the input files represent peaks the `arePeaks` flag needs to be set to `TRUE`. In such a way the DEScan2 package can work also with data coming from other sources, preferred by the user.

Furthermore, DEScan2 provides several functionalities for GenomicRanges

data structure handling. One over the others (`fromSamplesToChrsGRangesList`) gives the possibility to split a GenomicRangesList by the chromosomes. This procedure could be useful for parallelizing the computations on the chromosomes, assigning a single chromosome to a single computing unit. Taken as input a GenomicRangesList organized by samples, this method returns a list of chromosomes, where each element has a GenomicRangesList of samples, containing only the regions associated to the single chromosome.

[Create figure to better explain the transformation]

Other useful utilities are `keepRelevantChrs`, that takes a GenomicRangesList and a list of chromosomes and return only the interested chromosomes. `saveGRangesAsTsv` that saves a tab separated value file starting from a GenomicRanges. `saveGRangesAsBed` that save a standard BED file format starting from a GenomicRanges data structure. `setGRangesGenomeInfo` which, starting from a genome code, sets a specific *genomeInfo* to a *GenomicRanges* object.

3.3 Case Study

Few words on epigenomic data

We illustrate the performances of DEScan2 using a dataset Su2017 that describes in vivo adult mouse dentate granule neurons before and after synchronous neuronal activation using Atac-Seq and RNA-Seq technologies (see sections 1.2.2 and 1.2.1 for a description of these sequencing techniques).

This dataset is organized in 62 samples of Atac-Seq and RNA-Seq, extracted at different time points, with four replicates at each time point. We chose to compare the differences btween the first two stages, time 0 (E0) and 1 hour after neuronal induction (E1), in order to show a possible Atac-Sec workflow for Differential Enrichment, and how to integrate this data type with RNA-Seq. A general illustration of our dataset is represented in figure 3.3.1.

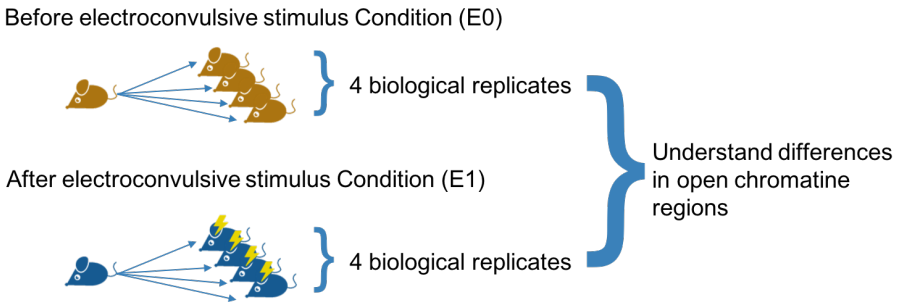

Figure 3.3.1: An illustration of our extraction of the Su2017 dataset.

We downloaded the data from Gene Expression Omnibus (GEO) database Edgar2002, Barrett2013 with accession number GSE82015[1] and mapped raw data using *STAR* Dobin2013 with default parameter on Mus Musculus Genome ver.10 (mm10).

In order to detect the open chromatin regions we run our peak caller, cutting the genome in bins of 50bp and using running windows of minimum 50bp and maximum 1000bp. In such a way we are able to detect not just broad peak, but also smaller peaks.

To be confident with our results we compared the DEScan2 detected peaks with the same validated regions (Arc and Gabrr1) in the original work Su2017. The lower part of figure 3.3.2 shows the detected and validated regions (in blue and red) resulting differentially enriched between the E0 (in pink) and E1 (in green) conditions, while the upper part shows DEScan2 peaks (in blue), highlighting a capability to catch not only the same regions of the published ones, but also (gold circles) to be more careful in the smaller peaks detection.

[1] `https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE82015`

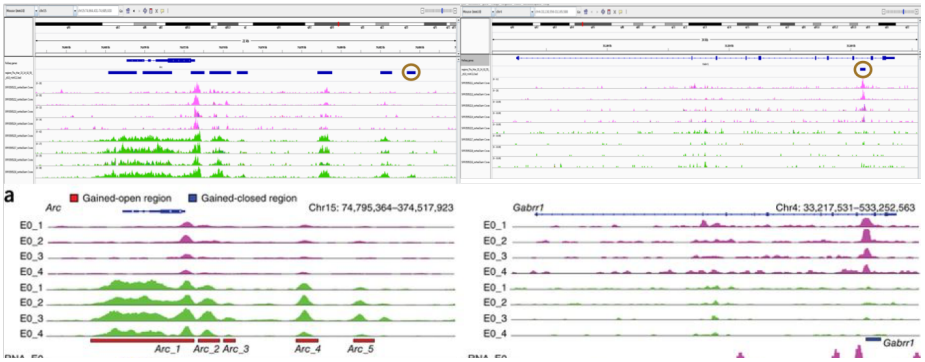

Figure 3.3.2: A comparison of DEScan2 detected peaks with validated peaks in article Su2017.

Moreover, we run *MACS2* on the same samples, and (as shown in figure 3.3.3) DEScan2 seems able to catch much more peaks than *MACS2* for each sample.

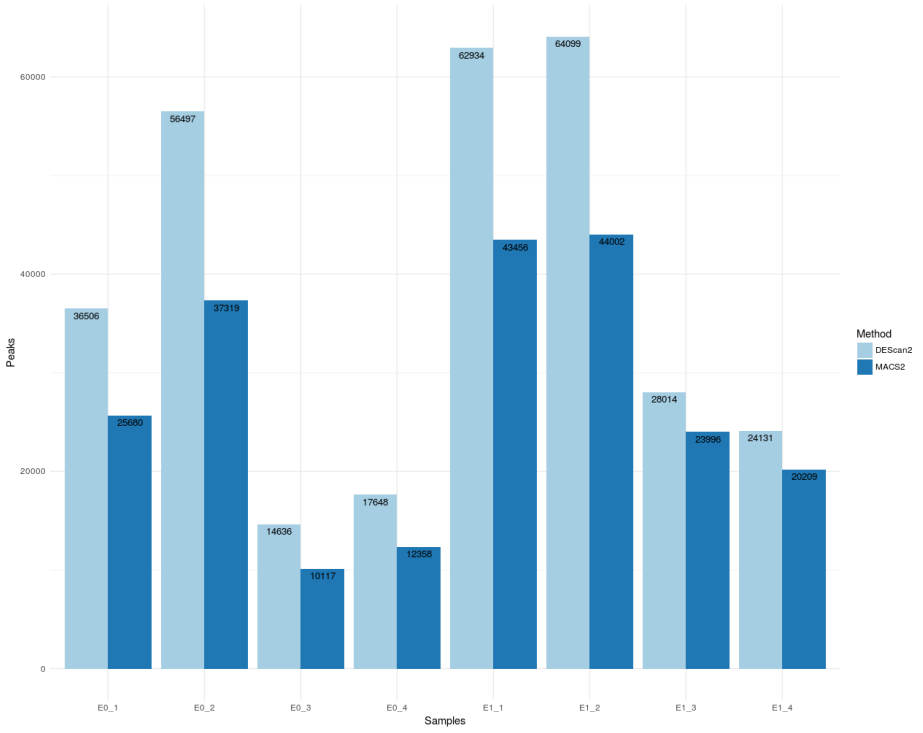

Figure 3.3.3: A comparison of DEScan2 and *MACS2* detected peaks for each sample in the dataset.

While it is very important to detect good peaks with a peak caller, it seems to be more relevant to detect reliable regions. Indeed, during the filtering step, the number of peaks depends not only by the peak score, but also by the number of replicates designed in the experiment. The figure 3.3.4 puts in relation these two relevant information. On the x-axis is represented the number of replicates, while on the y-axis is traced the number of peaks, and each curve represents a different threshold on the peaks score, showing that higher are the thresholds on the scores and the number of replicates, lower is the number of the detected peaks. Highlighting a proportional inversion between the number of the peaks and the combination of the number of samples and the detected regions score.

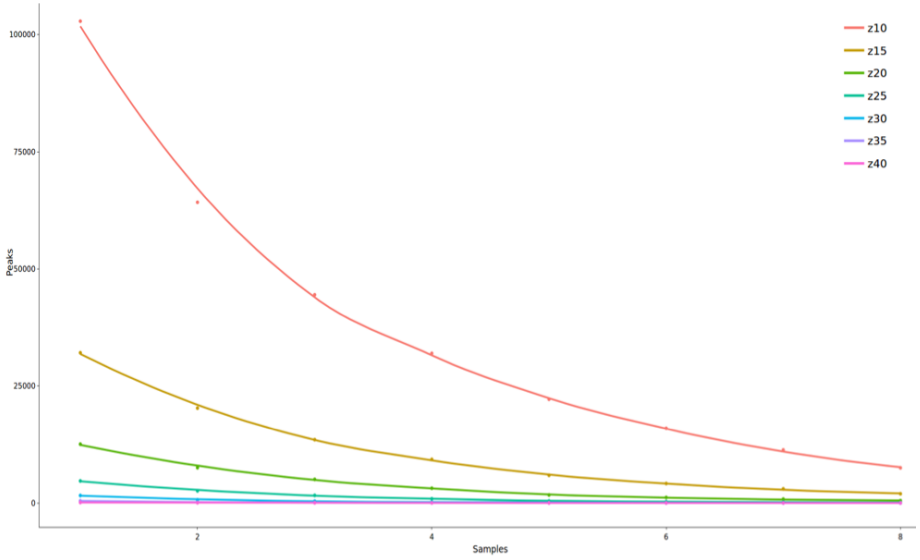

Figure 3.3.4: Filtering the detected regions with different thresholds on peak scores.

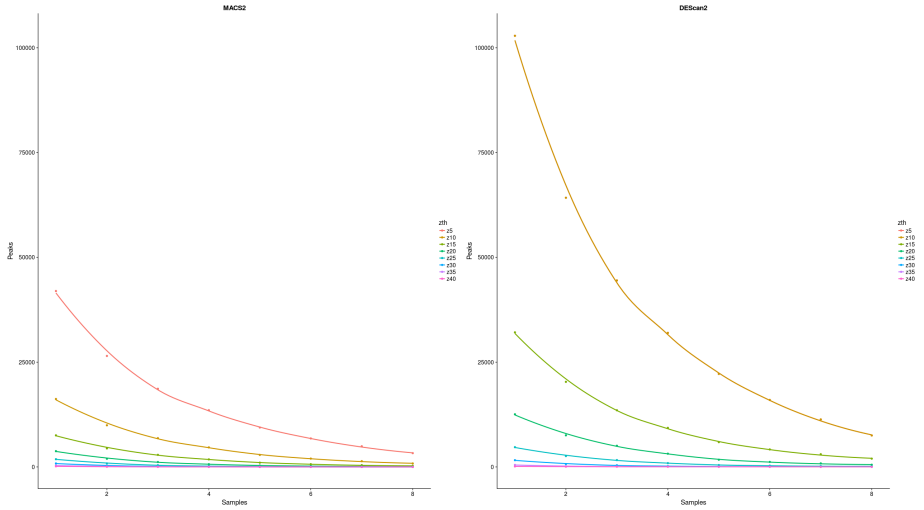

Figure 3.3.5: Filtering the detected regions with different thresholds on peak scores between *MACS2* and DEScan2.

The filtered-in regions can be processed by DEScan2 in order to obtain a count matrix with samples on the columns and peaks on the rows. This type of data structure is very versatile, because it enables to perform several operations, like the Differentially Enriched genomic Regions (DERs) and, if possible, the integration with other kind of omics, as RNA-Seq.

In order to preserve the information associated to the peaks, DEScan2 produces as output a *SummarizedExperiment* (figure 3.3.6) data structure, which enables to retrieve the count matrix with `assays` method, and to access the peaks information in *GenomicRanges* format with the `rowRanges` method.

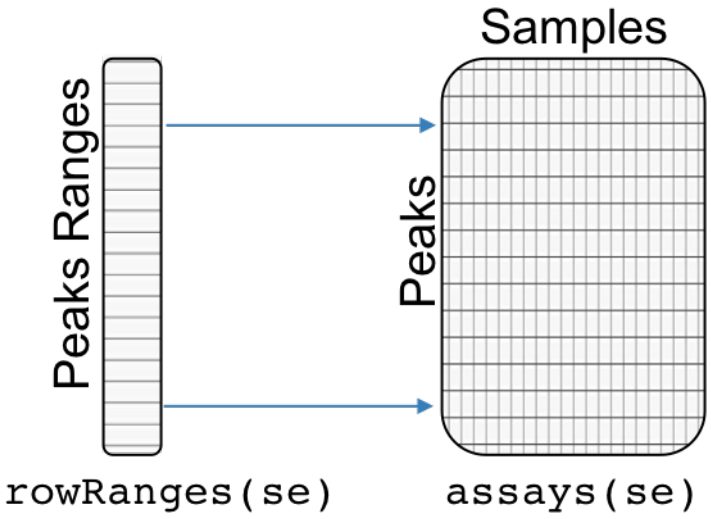

Figure 3.3.6: An illustration of the *SummarizedExperiment* data structure produced by DEScan2.

Before to proceed to detect DERs, it is a good standard to normalize the data, also because without any kind of normalization we are not able to detect any DER. The nature of the data, in count format, makes it possible to apply several well known RNA-Seq normalizations techniques, as *TMM*, *upper-quartile*, *full-quantile*, *RUV-Seq*, etc Risso2014, Robinson2010, Dillies2013.

While the *TMM* and *upper-quartile* normalizations modify the data in a way that makes it impossible to detect DERs, other kind of normalizations and combinantions of them give good results.

The figure 3.3.7 sintetizes this concept very well, highlighting a relation between the number of DERs and the minumum number of samples used for filtering the data during the DEScan2 filtering step.

The plot shows that *upper-quantile*, even if combined with *RUV-Seq* normalization, is not able to linearly detect a good amount of DERs, while *full-quantile*, when combined with *RUV-Seq* seems to affect the data in a way that overde-

tect the number of DERs. When looking at the *full-quantile* and *RUV-Seq* by themself seem to perform better than the other normalizations. The first one has a downhill almost linear, while the second one has a very fast downhill with a regrowth when the number of samples is higher.

Even if these normalization methods show good performances with this type of epignomic data, our investigations suggest that more testing is required, but maybe an ad-hoc normalization method for these data has to be developed.

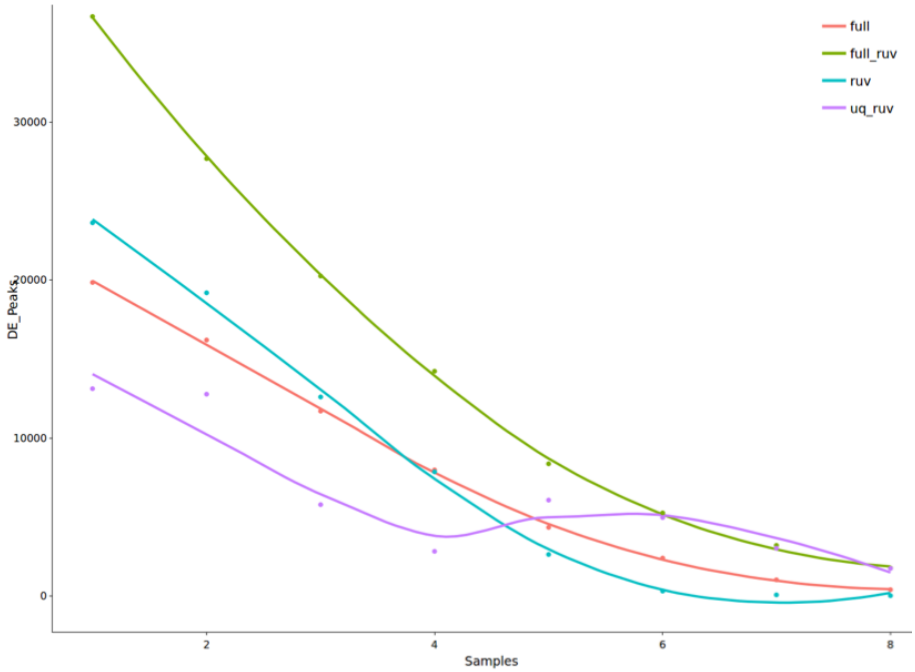

Figure 3.3.7: The figure shows the effects of different normalizations on the epigenomic differentially enriched regions.

To estimate the DERs, any of the RNA-Seq methods can be applied, such as *DESeq2*, *edgeR*, *NOISeq*, etc Robinson2009, McCarthy2012, Tarazona2012.

In this case, we decided to use *edgeR* package, because of its wide range of available statistical approaches and the possibility to better tune the design of

the experiment. Indeed, because we used the RUV-Seq normalized counts with `k` parameter set to 4, we modeled the experimental design with the `model.matrix` function, adding to our model not only the experimental conditions, but also the RUV-Seq estimated weights. Then we used the resulted design to estimate the dispersion and fit a Quasi-Likelihood test, as defined in edgeR.

The figure 3.3.8 shows a volcano plot of DERs between E0 and E1 conditions. Red dots highlights the regions with a False Discovery Rate (FDR) lower than 0.05, while blue dots highlight not significant regions.

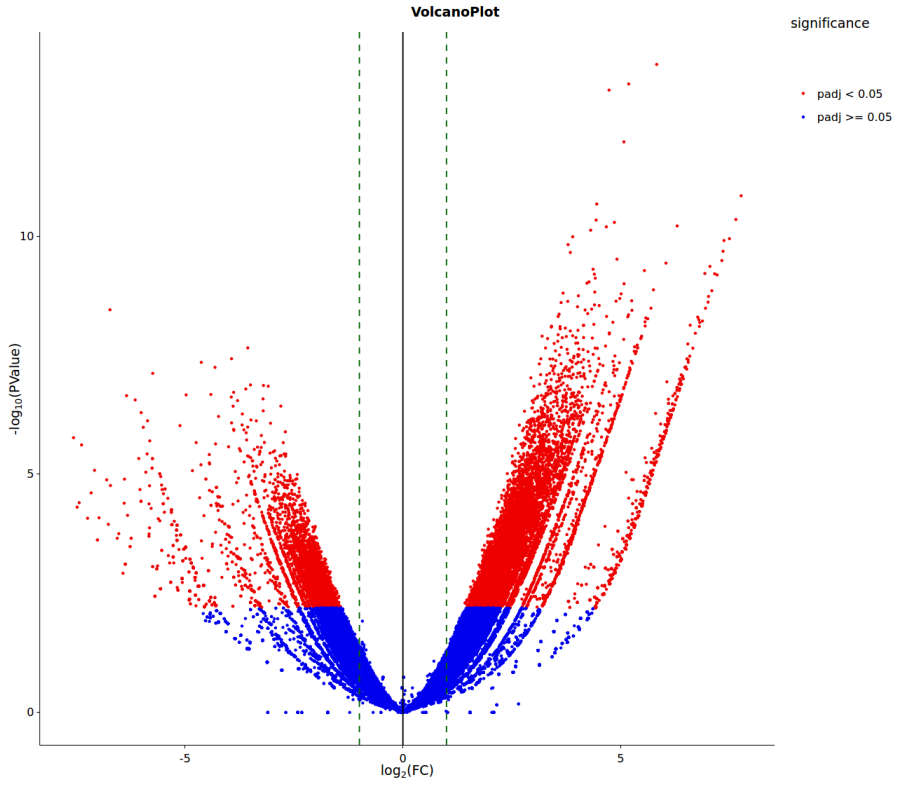

Figure 3.3.8: A volcano plot of Differential Enriched Regions. Blue dots represent the not significant DERs, while the red ones represent the significant DERs.

Next task is to integrate the obtained results with other omic data types, as RNA-Seq. Because of the low number of the samples, the easiest way to integrate the data is to annotate the DERs with differentially expressed genes resulting from the analysis of RNA-Seq.

For the Differential expression of the RNA-Seq data we firstly quantified the signal with `featureCounts` methods available in the *Rsubread* Liao2013 Bioconductor package. Then we filtered lowly expressed genes with the *proportion* test as implemented in *NOISeq* package, and applied the `noiseq` method for differential expression.

We selected the significant DEGs with a probability higher than 0.95, and used these genes to annotate the peaks with `annotatePeakInBatch` method of ChIPpeakAnno. Figure 3.3.9 illustrates with green circles the peaks with an annotated gene with distance lower than 10000bp from the gene TSS. Realizing the plot with the *plotly* library it's possible to enhance the names of the genes with a tip window.

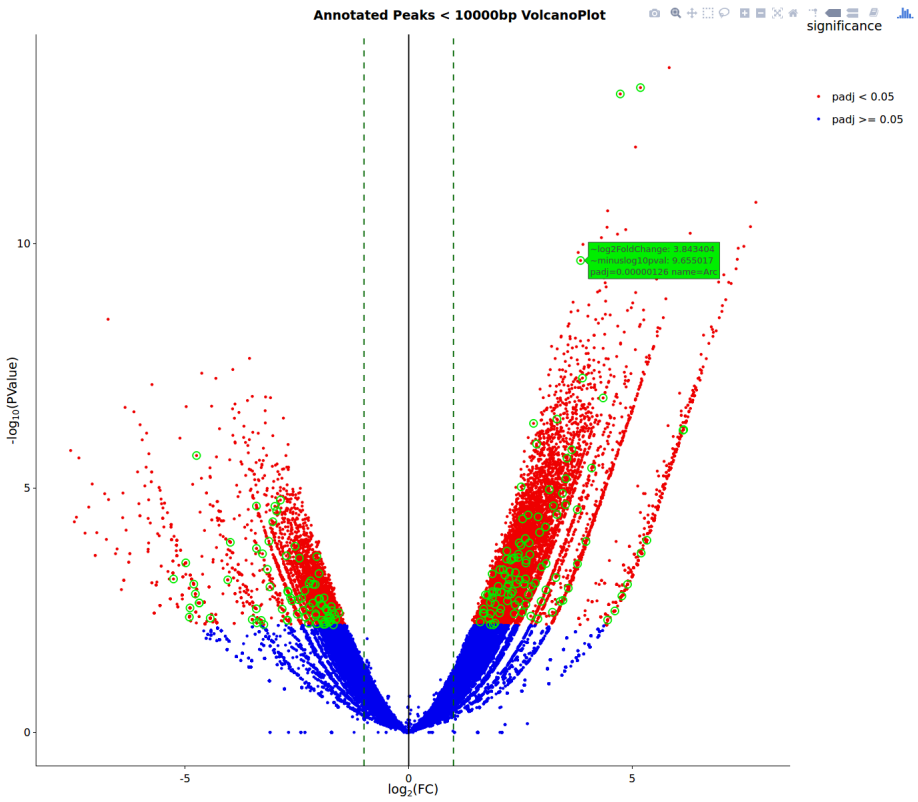

Figure 3.3.9: A volcano plot of DERs. Blue dots represent the not significant DERs, while the red ones represent the significant DERs. Green circles highlights the peaks with a Differentially Expressed Gene (DEG) annotated.

Chapter 4

IntegrHO - Integration of High-Throughtput Omics data

4.1 Introduction

4.2 Methods

4.2.1 Single Omic Approach

4.2.2 Multi Omic Approach

Low Level Itegration

High Level Itegration

4.3 Implementation Aspects

4.4 Reproducible Computational Research

4.5 Results

Chapter 5

Conclusions & Future Works

Appendices

.1 R Language

.2 R Markdown Language

Chapter 6

Bibliography

Acronyms

DEG Differentially Expressed Gene. 30, 31

DER Differentially Enriched genomic Region. 16, 26–31

DEScan2 Differential Enriched Scan 2. 15–17, 20–24, 26, 27, 47

GEO Gene Expression Omnibus. 22

MLE Maximum Likelihood Extimator. 19

mm10 Mus Musculus Genome ver.10. 22

List of Figures

List of Tables