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 [1, 2] 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 [1] 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 [3] 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 [4] data structures and methods, and it is available through Bioconductor repository since version 3.7.

The tool is organized in three main steps. A peak caller, which is a standard moving scan window that compares the reads counts signal within a sliding window, to the signal 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 represents 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.

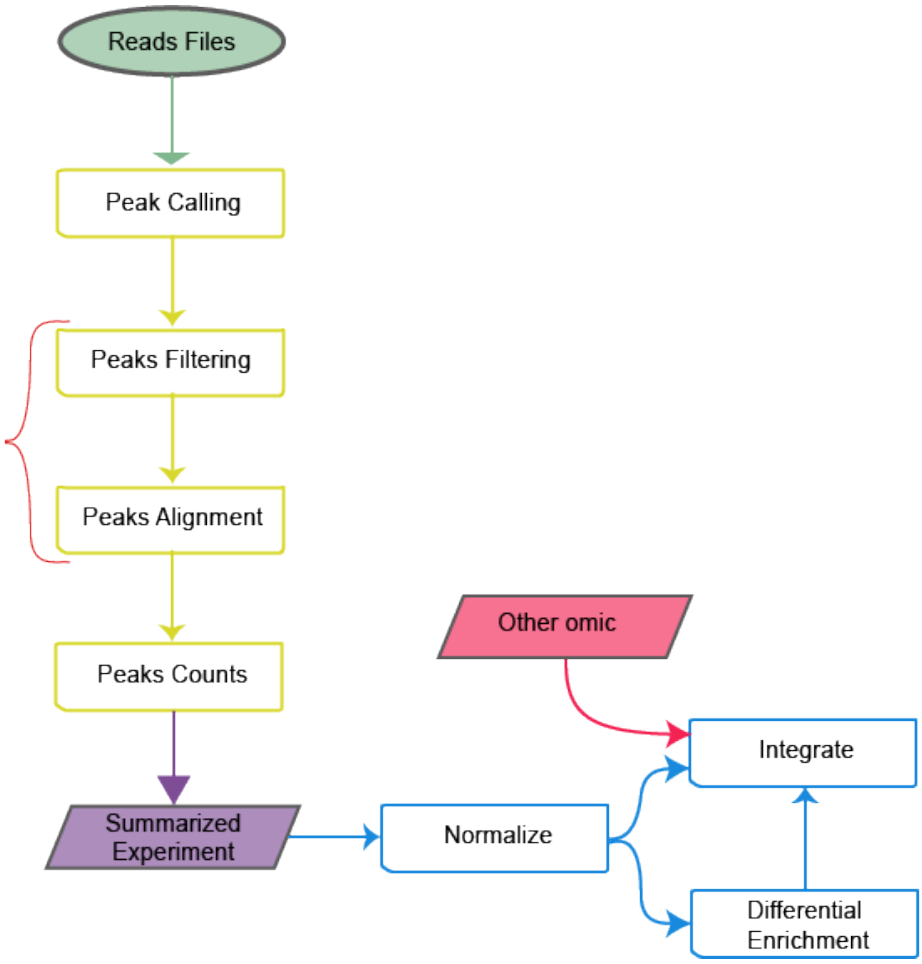

Figure 3.1.1: A differential enrichement flow representation. DEScan2 steps are highlighed in yellow.

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.

Furthermore, it offers some additional features described in 3.2.4.

3.2.1 Peak Caller

The Peak Caller (`findPeaks` function) takes as input a set of alignment files (BAM [5] 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* [6], 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 chromosome list of *GenomicRangesList*, where each element represents the file containing just the *GenomicRanges* of the specific chromosome (see section 3.2.4).

For each element of this data structure the algorithm firstly divides each chromosome as bins of `binSize` parameter length (default value 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 values are 5000bp and 10000bp) parameters, computing a matrix of n bins and p windows.

The coverages matrix is useful to merge contigous regions and to compute a score for each of them, applying a Maximum Likelihood Extimator (MLE), assuming a Poisson distribution.

[PUT THE POISSON DISTRIBUTION AND THE LIKELIHOOD]

[describe output as tsv]

3.2.2 Peak Filtering and Alignment

In order to filter out false positives peaks, we designed a method (`finalRegions`) which firstly filters out low score regions and then aligns the resulting regions

between the samples, using two different thresholds. One on the peaks's score and one 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 file. This is the data structure produced by the peak caller, but, we also developed a method to load peaks produced by other software like MACS [7], as described in section 3.2.4.

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

Then, for aligning the peaks between the samples, it extends a 200bp window in both directions of remaining regions, computing the overlaps using the `findOverlapsOfPeaks` method (with `connectedPeaks` parameter set as `merge`), as defined in *ChIPpeakAnno* [8] R/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. On our experience, we suggest to set the samples threshold as a mutiple of the number of replicates of the conditions.

3.2.3 Counting Peaks

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

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, respectively, represent the regions and the samples.

In order to keep trace of all information associated to the regions, it produces a *SummarizedExperiment* [9] data structure, giving the possibility to retrieve the *GenomicRanges* of associated peaks and the count matrix, respectively, using `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, as RNA-Seq.

3.2.4 Additional Features

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

To give the possibility to use our pipeline with external peaks, the method `readFilesAsGRangesList` takes as input a directory containing 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/aligning 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`.

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, when common operations on them, between multiple samples, are needed. 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 with a cleaned *genomeInfo* assigned. `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 ATAC-Seq data

We illustrate the performances of DEScan2 using a dataset [10] 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.

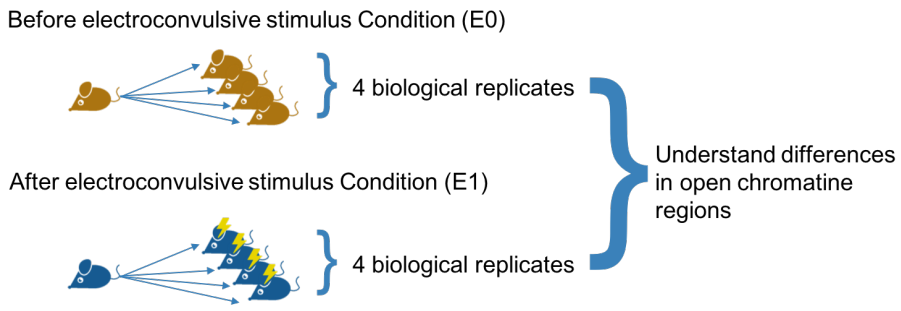

Figure 3.3.1: An illustration of our extraction of the [10] dataset.

We downloaded the data from Gene Expression Omnibus (GEO) database

[11, 12] with accession number GSE82015[1] and mapped raw data using *STAR* [13] 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 [10]. 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.

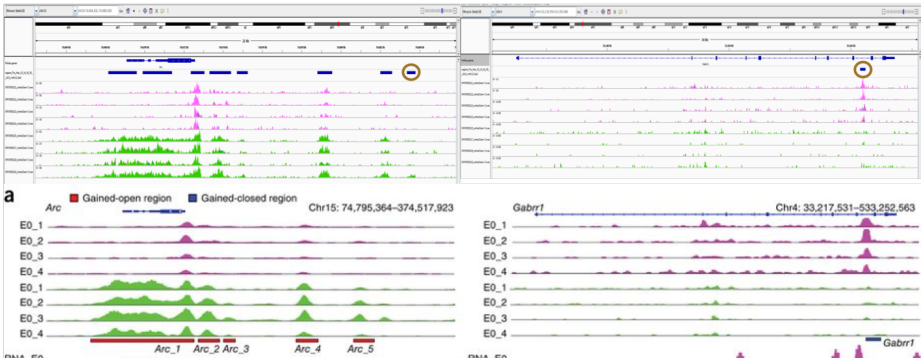

Figure 3.3.2: A comparison of DEScan2 detected peaks with validated peaks in article [10].

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.

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

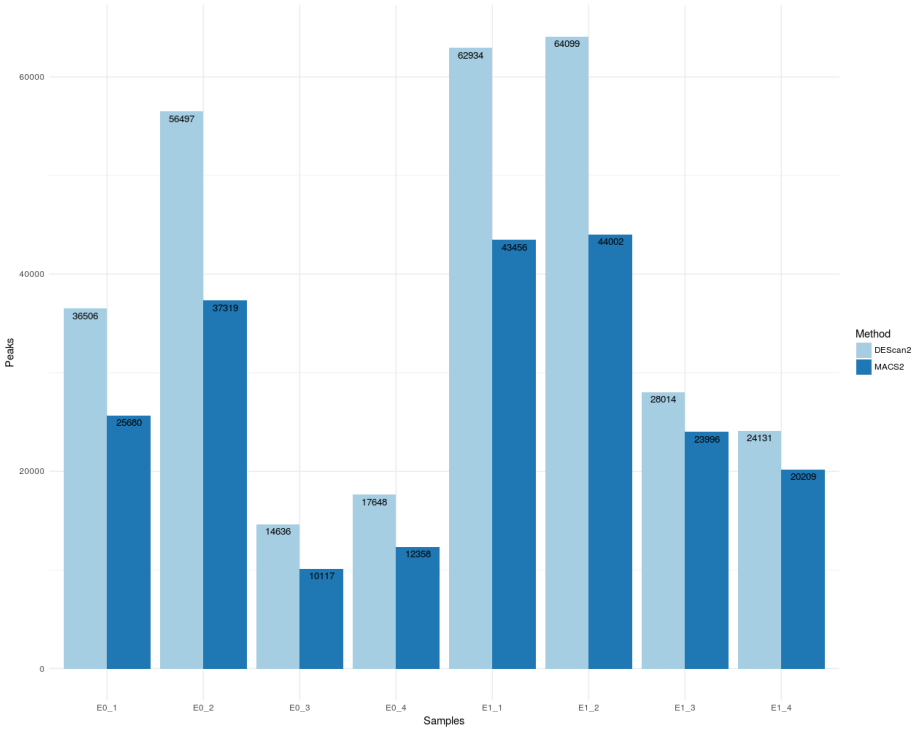

Figure 3.3.3: A comparison of DEScan2 and *MACS2* detected peaks for each sample in the dataset.

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.

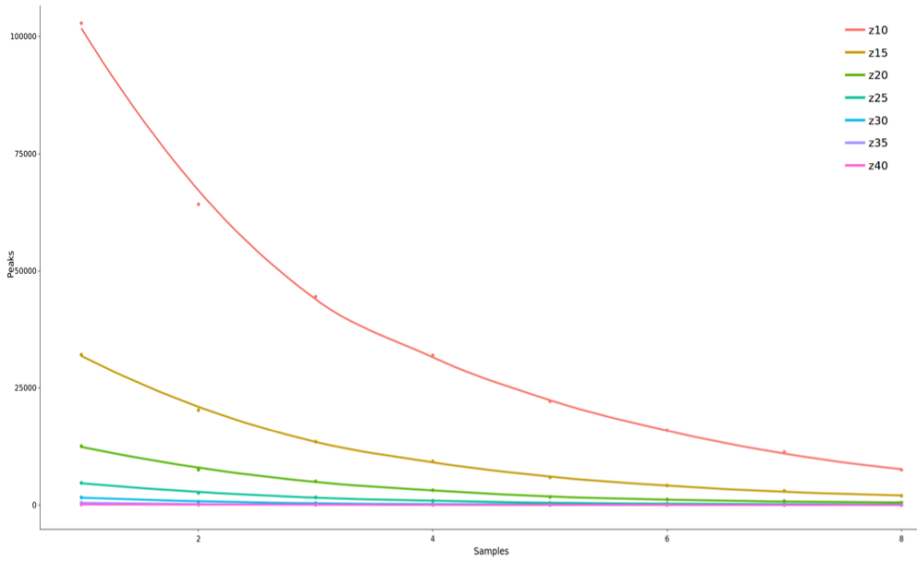

Figure 3.3.4: Filtering the detected regions with different thresholds on peak scores.

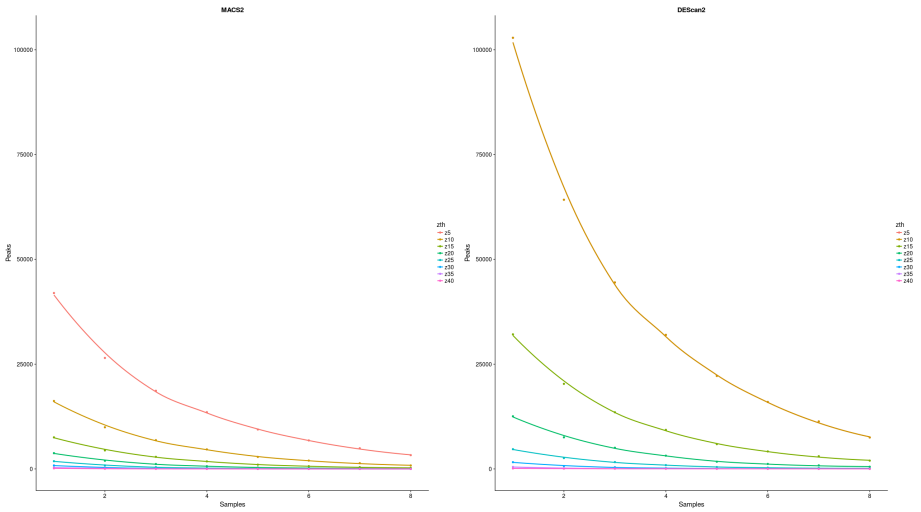

Figure 3.3.5: Filtering the detected regions with different thresholds on peak scores between *MACS2* and DEScan2.

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.

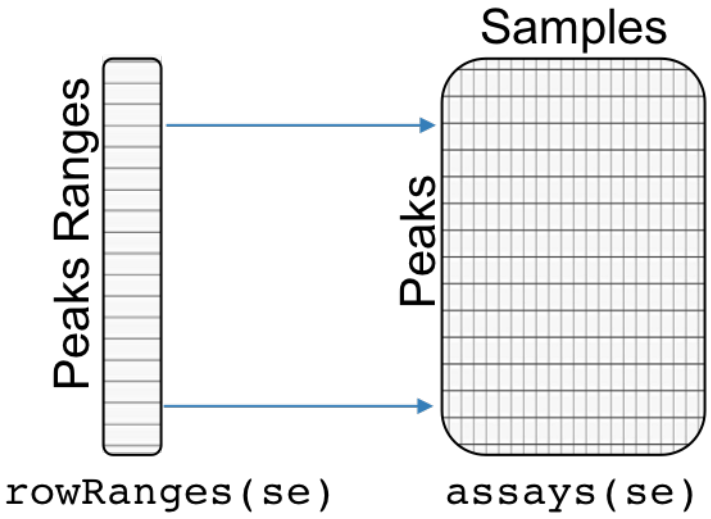

Figure 3.3.6: An illustration of the *SummarizedExperiment* data structure produced by DEScan2.

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**, 14, 15].

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.

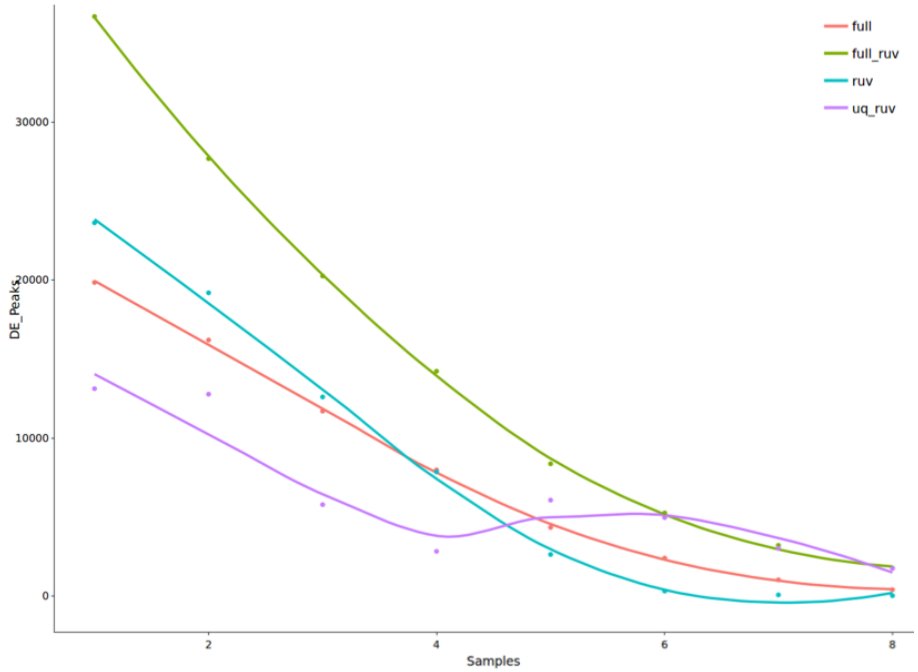

Figure 3.3.7: The figure shows the effects of different normalizations on the epigenomic differentially enriched regions.

To estimate the DERs, any of the RNA-Seq methods can be applied, such as *DESeq2*, *edgeR*, *NOISeq*, etc [16–18].

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.

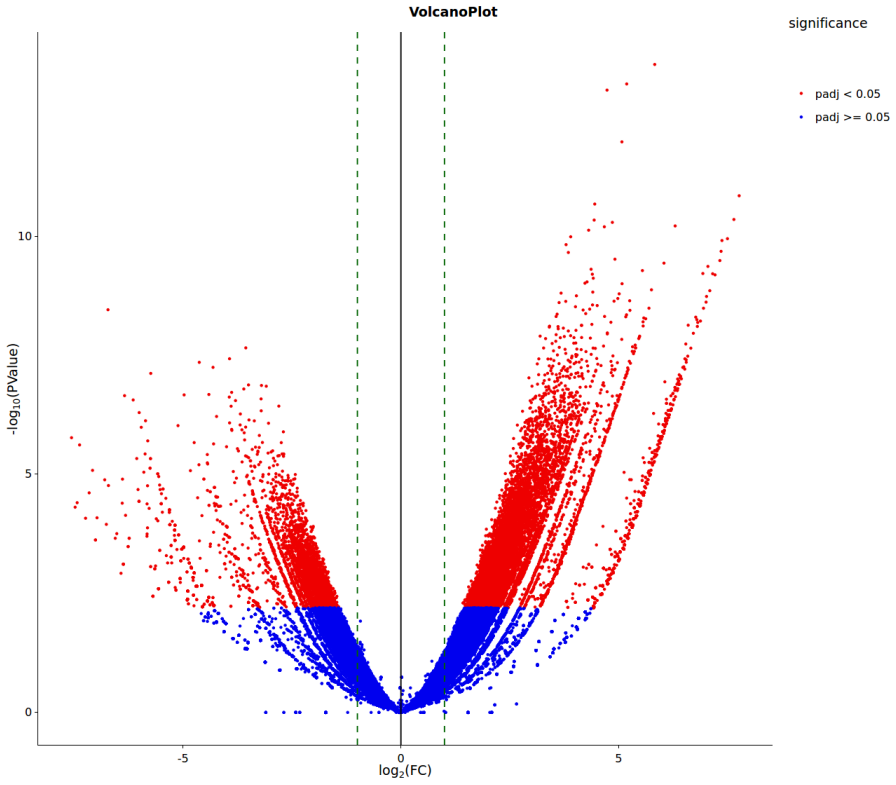

Figure 3.3.8: A volcano plot of Differential Enriched Regions. Blue dots represent the not significant DERs, while the red ones represent the significant DERs.

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* [19] 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.

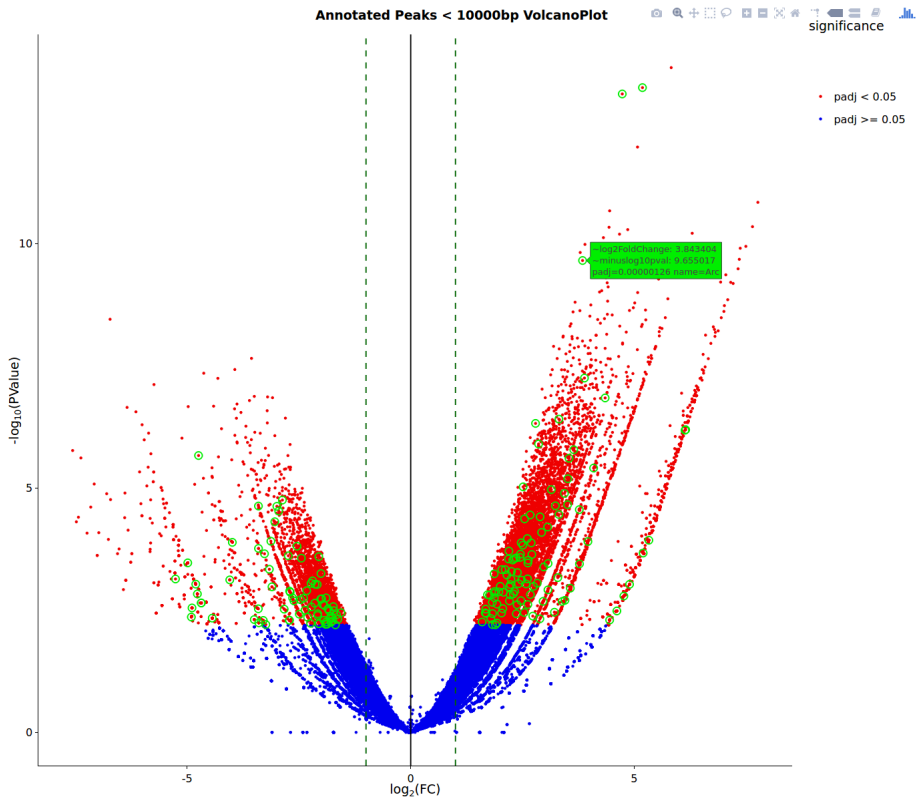

Figure 3.3.9: A volcano plot of DERs. Blue dots represent the not significant DERs, while the red ones represent the significant DERs. Green circles highlights the peaks with a Differentially Expressed Gene (DEG) annotated.

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

1. Koberstein, J. N. *et al.* Learning-dependent chromatin remodeling highlights noncoding regulatory regions linked to autism. *Science Signaling.* ISSN: 19379145. doi:10.1126/scisignal.aan6500 (2018).
2. Auerbach, R. K. *et al.* Mapping accessible chromatin regions using Sono-Seq. *Proceedings of the National Academy of Sciences.* ISSN: 0027-8424. doi:10.1073/pnas.0905443106 (2009).
3. Ihaka, R. & Gentleman, R. R: A Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics.* ISSN: 15372715. doi:10.1080/10618600.1996.10474713. arXiv: arXiv:1011.1669v3 (1996).
4. Gentleman, R. *et al.* Bioconductor: open software development for computational biology and bioinformatics. *Genome Biology.* ISSN: 1465-6914. doi:10.1186/gb-2004-5-10-r80 (2004).
5. Li, H. *et al.* The Sequence Alignment/Map format and SAMtools. *Bioinformatics.* ISSN: 13674803. doi:10.1093/bioinformatics/btp352. arXiv: 1006.1266v2 (2009).

6. Lawrence, M. *et al.* Software for Computing and Annotating Genomic Ranges. *PLoS Computational Biology* **9** (ed Prlic, A.) e1003118. ISSN: 1553-7358 (2013).

7. Zhang, Y. *et al.* Model-based analysis of ChIP-Seq (MACS). *Genome Biology.* ISSN: 14747596. doi:10.1186/gb-2008-9-9-r137 (2008).

8. Zhu, L. J. *et al.* ChIPpeakAnno: A Bioconductor package to annotate ChIP-seq and ChIP-chip data. *BMC Bioinformatics.* ISSN: 14712105. doi:10.1186/1471-2105-11-237 (2010).

9. Morgan M, Obenchain V, Hester J, P. H. SummarizedExperiment: SummarizedExperiment container. doi:https://doi.org/doi:10.18129/B9.bioc.SummarizedExperiment (2018).

10. Su, Y. *et al.* Neuronal activity modifies the chromatin accessibility landscape in the adult brain. *Nature Neuroscience.* ISSN: 15461726. doi:10.1038/nn.4494 (2017).

11. Edgar, R. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Research.* ISSN: 13624962. doi:10.1093/nar/30.1.207 (2002).

12. Barrett, T. *et al.* NCBI GEO: Archive for functional genomics data sets - Update. *Nucleic Acids Research.* ISSN: 03051048. doi:10.1093/nar/gks1193 (2013).

13. Dobin, A. *et al.* STAR: Ultrafast universal RNA-seq aligner. *Bioinformatics.* ISSN: 13674803. doi:10.1093/bioinformatics/bts635. arXiv: 1201.0052 (2013).

14. Robinson, M. D. & Oshlack, A. A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology.* ISSN: 14747596. doi:10.1186/gb-2010-11-3-r25. arXiv: PMC2864565 (2010).

15. Dillies, M. A. *et al.* A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. *Briefings in Bioinformatics.* ISSN: 14675463. doi:10.1093/bib/bbs046 (2013).

16. Robinson, M. D., McCarthy, D. J. & Smyth, G. K. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics (Oxford, England)* **26,** 139–140. ISSN: <null> (2009).

17. McCarthy, D. J., Chen, Y. & Smyth, G. K. Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation. *Nucleic Acids Research* **40,** 4288–4297. ISSN: 03051048 (2012).

18. Tarazona, S., García, F., Ferrer, A., Dopazo, J. & Conesa, A. NOIseq: a RNA-seq differential expression method robust for sequencing depth biases. *EMBnet.journal* **17,** 18. ISSN: 2226-6089 (2012).

19. Liao, Y., Smyth, G. K. & Shi, W. The Subread aligner: Fast, accurate and scalable read mapping by seed-and-vote. *Nucleic Acids Research.* ISSN: 03051048. doi:10.1093/nar/gkt214 (2013).

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