Development of algorithms and next-generation sequencing data workflows for the analysis of gene regulatory networks

Dissertation
zur Erlangung des Doktorgrades
Dr. rer. nat.
der Fakultät für Mathematik und Informatik
der Georg-August-Universität Göttingen

im PhD Programme in Computer Science (PCS)
der Georg-August University School of Science (GAUSS)

vorgelegt von

Orr Shomroni
aus Rehovot, Israel

Göttingen, im 2017

Betreuungsausschuss:	Dr. Stefan Bonn, Deutsches Zentrum für Neurodegenerative Erkrankungen (DZNE), Göttingen
	Prof. Dr. Tim Beißbarth, Institut für Medizinische Statistik, Universitätsmedizin, Georg-August Universität, Göttingen
	Prof. Dr. Stephan Waack, Institut für Informatik, Georg-August Universität, Göttingen
Prüfungskommission:	
Referent:	Dr. Stefan Bonn, Deutsches Zentrum für Neurodegenerative Erkrankungen (DZNE), Göttingen
Korreferent:	Prof. Dr. Stephan Waack, Institut für Informatik, Georg-August Universität, Göttingen
Weitere Mitglieder der Prüfungskommission:	Prof. Dr. Burkhard Morgenstern, Institut für Mikrobiologie und Genetik Abtl. Bioinformatik, Georg-August Universität, Göttingen
	Prof. Dr. Carsten Damm, Institut für Informatik, Georg-August Universität, Göttingen
	Prof. Dr. Florentin Wörgötter, Physikalisches Institut Biophysik, Georg-August-Universität, Göttingen
	Prof. Dr. Tim Beißbarth, Institut für Medizinische Statistik, Universitätsmedizin, Georg-August Universität, Göttingen
Tag der mündlichen Prüfung:	der 2. März 2017

Contents

List of Figures **v**

List of Tables **vii**

List of Abbreviations **viii**

Summary **1**

1 Introduction **2**
- 1.1 DNA structure and function . . . 3
  - 1.1.1 Chromatin and DNA compression . . . 3
  - 1.1.2 Gene expression and functionality . . . 5
  - 1.1.3 Regulation of gene expression . . . 5
- 1.2 Studying gene regulation and expression with NGS . . . 7
  - 1.2.1 Mapping DNA-protein interactions: ChIP-seq . . . 7
  - 1.2.2 Profiling methylation patterns: MeDIP-seq . . . 8
  - 1.2.3 Gene and exon expression: RNA-seq . . . 9
- 1.3 Scope of the thesis . . . 9

2 Fragment size estimation and protein enrichment testing **11**
- 2.1 Features of fragment size estimation . . . 13
  - 2.1.1 Technical background . . . 13
  - 2.1.2 'Phantom' peak and mappability . . . 14
  - 2.1.3 Enrichment test . . . 15
- 2.2 *chequeR*: state-of-the-art package for fragment size estimation and enrichment testing . . . 16
  - 2.2.1 Indexed, random-access input files . . . 19
  - 2.2.2 Bit vectors, Boolean operations and C++ integration . . . 20
  - 2.2.3 Statistical validity . . . 21
  - 2.2.4 Informative region (sub)selection . . . 22
- 2.3 *chequeR* memory and runtime benchmark . . . 23
- 2.4 Overview . . . 26

3 Development of ChIP-seq analysis workflow **27**
- 3.1 Genome alignment . . . 28
  - 3.1.1 FastQ format . . . 29
  - 3.1.2 Bowtie2 usage and parameters . . . 29
    - 3.1.2.1 Reporting alignments bowtie2 vs. bowtie1 . . . 30

3.1.2.2 Read filtering with mapping quality . . . 31
3.2 Data visualization . . . 33
3.2.1 Replicate BAM merging . . . 33
3.3 The importance of quality control . . . 34
3.3.1 Determining sequencing quality: FastQC . . . 34
3.3.2 Alignment rate and genomic coverage . . . 35
3.3.3 ChIP-seq enrichment . . . 35
3.3.4 Saturation and pairwise correlations: *MEDIPS* . . . 36
3.3.5 Quality control overview . . . 37
3.4 Peak calling . . . 37
3.4.1 Broad vs. narrow peaks . . . 38
3.4.2 Peaks from technical or biological replicates . . . 39
3.4.2.1 Peak motif analysis . . . 40
3.4.2.2 Overlapping peaks with DEGs . . . 41
3.4.2.3 Peak calling on replicates — summary . . . 41
3.5 Differential occupation analysis . . . 42
3.6 Functional enrichment analysis . . . 43
3.7 Motif discovery . . . 44
3.8 Automation of ChIP-seq workflow . . . 44
3.9 Overview . . . 47

4 Development of MeDIP- and RNA-seq analysis workflows 53
4.1 MeDIP-seq workflow . . . 54
4.1.1 Alignment for MeDIP-seq data . . . 55
4.1.2 Identification of differentially methylated regions . . . 55
4.1.3 Automation of MeDIP-seq workflow . . . 56
4.2 RNA-seq workflow . . . 56
4.2.1 Alignment with STAR . . . 56
4.2.2 Differential expression for RNA-seq data . . . 57
4.2.2.1 Differentially expressed genes and exons . . . 57
4.2.3 Automation of RNA-seq workflow . . . 58
4.3 Overview . . . 58

5 Case Study 1: chromatin modifications regulating memory acquisition and maintenance 60
5.1 Study overview . . . 60
5.2 Histone modification data . . . 61
5.2.1 Data quality . . . 62
5.2.2 Marker-specific changes . . . 64
5.3 Methylated DNA data . . . 66
5.3.1 Data quality . . . 67
5.4 Overview . . . 68

6 Case Study 2: regulation of cilia development by p73 70
6.1 Experiment design . . . 71
6.2 Differential gene expression during ciliogenesis . . . 72
6.2.1 Data quality . . . 72

6.2.2 Ciliogenesis deregulation . . . 73
6.2.3 Differentially expressed genes and ciliogenic gene association . . . 74
6.3 Genes bound by p73 . . . 75
6.3.1 Data quality . . . 76
6.3.2 p73 motif discovery . . . 77
6.3.3 Annotation of p73 peaks . . . 77
6.4 Regulation of differentially expressed genes by p73 . . . 78
6.4.1 Homologous genes . . . 79
6.4.2 Overlaps between p73 peaks and motifs, differentially expressed genes and cilia-related genes . . . 79
6.4.3 Visualization of target genes . . . 80
6.5 Overview . . . 81

7 Discussion 84
7.1 *chequeR* . . . 84
7.1.1 Comparison of *chequeR* with other shift size estimators . . . 84
7.1.2 Further utilization of read cross-correlations . . . 85
7.2 NGS analysis workflows . . . 86
7.2.1 Realignment of multi-mapped reads . . . 86
7.2.2 Biological variability between replicates . . . 88

References 94

Acknowledgements 95

List of Figures

1.1 DNA bases . . . 3
1.2 DNA compressed structure . . . 4
1.3 Heterochromatin vs. euchromatin . . . 5
1.4 Central Dogma . . . 5
1.5 Alternative splicing . . . 6
1.6 Initiation of DNA transcription . . . 6
1.7 Chromatin immunoprecipitation sequencing . . . 8
1.8 Methylated DNA immunoprecipitation sequencing . . . 9
1.9 RNA sequencing . . . 10

2.1 ChIP-seq tags . . . 12
2.2 ChIP-seq tag shifting . . . 12
2.3 Fragment size estimation . . . 13
2.4 'Phantom' peak . . . 14
2.5 Enrichment ChIP-seq samples . . . 17
2.6 *chequeR* workflow . . . 18
2.7 *chequeR* plot . . . 19
2.8 Benchmark chequeR human H4K20me1 . . . 24
2.9 Memory benchmark chequeR . . . 25
2.10 Runtime benchmark chequeR . . . 25

3.1 ChIP-seq analysis overview . . . 48
3.2 Custom genome browser . . . 49
3.3 FastQC per-base quality . . . 49
3.4 FastQC sequence quality . . . 50
3.5 Saturation correlation ChIP-seq samples . . . 50
3.6 PCA plot for ChIP-seq data . . . 51
3.7 Peaks of histone modifications and transcription factors in ChIP-seq data 51
3.8 Binding motifs of p53 transcription factor family . . . 51
3.9 Visualization differential occupation . . . 52
3.10 ChIP-seq scripts overview . . . 52

4.1 Overview of MeDIP-seq computational analysis . . . 53
4.2 Overview of RNA-seq computational analysis . . . 54

5.1 Experimental design of learning mice study . . . 61
5.2 Alignment rates of high-quality vs. uniquely-mapped reads . . . 62
5.3 Number of differential histone post-translational modification regions . . . 66

5.4 Overlap differential histone post-translational modifications with differentially expressed genes . . . 66
5.5 Alignment rates of high-quality vs. uniquely-mapped reads . . . 67
5.6 MeDIP-seq sample with minimal saturation . . . 68

6.1 Known graphical regulatory network of multiciliogenesis . . . 71
6.2 PCA plots for wild-type and p73 knock-out samples . . . 73
6.3 Heatmaps for top 20 enriched GO categories . . . 74
6.4 Models for p73 DNA binding . . . 76
6.5 Motif of TAp73 binding-site . . . 78
6.6 Enrichment test differentially expressed genes with p73 enrichment . . . 80
6.7 Overlap differentially expressed genes, p73 peaks, p73 binding sites and cilia-related genes . . . 81
6.8 Tracks showing expressions of transcription factors related to ciliogenesis . 82
6.9 Discovered graphical regulatory network of multiciliogenesis . . . 83

List of Tables

3.1 Bowtie parameters . . . 31
3.2 Bowtie benchmark . . . 32
3.3 MAPQ values . . . 33
3.4 Quality control example . . . 38
3.5 Motifs of peak sets from rules . . . 40
3.6 Peaks from different rules . . . 41
3.7 Peaks overlapping differentially expressed genes . . . 42

5.1 Genomic coverage of ChIP-seq samples . . . 63
5.2 Enrichment test statistics . . . 64
5.3 Saturation correlation of ChIP-seq samples . . . 64

6.1 Differentially expressed genes for RNA-seq data between wild-type and p73 knock-out at different time-points . . . 75
6.2 Quality TAp73 samples . . . 77
6.3 Contingency table differentially expressed genes . . . 79

Abbreviations

A	adenine
ACC	anterior cingulate cortex
BAM	binary alignment/map
BGZF	blocked GNU zip format
BiTS	batch isolation of tissue-specific chromatin
bp	base pair
C	cytosine
cDNA	complementary DNA
CFC	contextual fear conditioning
ChIP	chromatin immunoprecipitation
ChIP-seq	chromatin immunoprecipitation sequencing
CpG	5'-C-phosphate-G-3'
CRG	cilia-related gene
DE	differential expression
DEE	differentially expressed exon
DEG	differentially expressed gene
DHPTM	differential histone post-translational modification
DMR	differentially methylated region
DNA	deoxyribonucleic acid
DO	differential occupancy
EM	expectation maximization
ENCODE	Encyclopedia of DNA Elements
FDR	false discovery rate
FM	full-text minute

G	guanine
Gb	gigabyte
Gbp	giga bp
GLM	Gaussian linear model
GO	gene ontology
GRN	gene regulatory network
HOMER	hypergeometric optimization of motif enrichment
HPTM	histone post-translational modification
IDR	irreproducibility discovery rate
IGB	integrated genome browser
IGV	integrated genome viewer
IP	immunoprecipitation
KO	knock-out
MACS2	model-based analysis of ChIP-seq data version 2
MaSC	mappability-sensitive cross-correlation
Mb	megabyte
MedIP	methylated DNA immunoprecipitation
MedIP-seq	methylated DNA immunoprecipitation sequencing
miRNA	micro RNA
ML	maximum likelihood
mRNA	messenger RNA
MTEC	murine tracheal epithelial cells
NGS	next generation sequencing
NSC	normalized strand coefficient
PCA	principle component analysis
PCR	polymerase chain reaction
piRNA	*piwi*-interacting RNA
QC	quality control

RAM	random access memory
RNA	ribonucleic acid
RNA-seq	RNA-sequencing
RPKM	reads per kilobase of transcript per million
RPM	reads per million
RSC	relative strand correlation
RSEM	RNA-seq by expectation maximization
SAM	sequence alignment/map
SOLiD	sequencing by oligonucleotide ligation and detection
sRNA	small RNA
sRNA-seq	small RNA sequencing
T	thymine
TF	transcription factor
TSS	transcription start site
UCSC	University of California Santa Cruz
UTR	untranslated region
WebGestalt	web-based gene set analysis toolkit
WT	wild-type

Summary

Unraveling genetic and epigenetic mechanisms behind various biological processes is possible with next generation sequencing (NGS) methodologies, with a multitude of tools developed to analyze such data. Nevertheless, automated, robust and flexible workflows that analyze NGS data quickly and efficiently have been lacking. In addition, given that many NGS studies today involve integration of results from multiple resources in order to better understand complex biological mechanisms, the quick generation of primary results from separate NGS studies will allow researchers to focus on the result integration. As such, the development of such automated workflows is essential in order to analyse multiple datasets of the same type quickly and efficiently.

In addition to the implementation of analysis workflows, the lack of an efficient tool for fragment size estimation and enrichment testing of chromatin immunoprecipitation sequencing (ChIP-seq) data brought the necessity to develop such a tool, and so the *R* package *chequeR* was implemented and integrated into the ChIP-seq workflow.

The workflows developed for ChIP-seq, methylated DNA immunoprecipitation sequencing (MedIP-seq) and RNA-sequencing (RNA-seq) data were generated as automated scripts to integrate various analysis tools together in order to analyze datasets and return primary results. Having such workflows may allow users to generate said results with relative ease and use them in an integrative manner to establish regulatory networks between multiple genomic and epigenomic elements. This point is demonstrated in Chapters 5 and 6, where the former chapter discusses a study on the effect of short- and long-term memory on the epigenetic and genetic mechanisms in the mouse brain, while the latter chapter explains how the role of p73 in multiciliogenesis regulation was determined. With those workflows used in two particular case studies involving integration of various NGS data types, the importance of having reproducible, automated workflows to generate primary results quickly and simply, while allowing researchers to focus on the main integrative aspects of the studies, is displayed.

Chapter 1

Introduction

Humanity has been studying various organisms for centuries, including themselves, to better understand life and the mechanisms that create and maintain it, with recent developments in biological research allowing studying organisms on a microscopic scale. Such studies have uncovered the most fundamental building blocks of life, predominantly genes, ribonucleic acid (RNA) molecules and proteins, which interact together to provide organisms with life. However, the mechanisms by which those molecules interact with each other and themselves is so intricate, that determining those interactions one at a time would be highly time-consuming and virtually impossible. As such, next generation sequencing (NGS) techniques were developed to study such interactions in bulk, where different ones were used to study distinct kinds of interactions[1]. For example, chromatin immunoprecipitation sequencing (ChIP-seq) studies how a specific protein interacts with multiple genes to regulate their expressions.

Such technology allowed generating massive amounts of data, and required analyses to derive results and better understand the underlying biological mechanisms. For that purpose, a multitude of tools have been developed to analyze such data, with different tools covering various aspects of the analysis. Since specific NGS techniques are ubiquitous among different studies and require the same analysis, each NGS technique ought to have its own standardized workflow, which would provide robust, efficient and fast analysis. Therefore, the aim of this thesis was to create such efficient workflows for specific NGS techniques. Nevertheless, in order to develop robust workflows to analyse NGS data, it is necessary to understand the underlying biological concepts behind it. Hence, this chapter is dedicated to provide biological context to NGS techniques, which in turn will clarify the workflow development.

1.1 DNA structure and function

One particularly important building block of life is deoxyribonucleic acid (DNA), which is a double-stranded molecule consisting of four different bases that bind to each other in particular base pairs (bps) (adenine (A) to thymine (T) and cytosine (C) to guanine (G)) in order to hold the double-stranded structure together (Figure 1.1). Depending on the organism, DNA may extend to millions of bps, as they are used as templates to create various RNA and protein molecules required to initiate and sustain life (see Section 1.1.2). To make sure that maintenance is kept intact throughout the lifetime of an organism, its DNA is identical in cells that contain it. However, this begs the question of how different cell types exist. Furthermore, given that organisms grow and adapt to their environment, it is unclear how a static DNA sequence would allow such changes to occur. Both questions can be answered by acknowledging the role of regulation in DNA transcription, whereby DNA subunits called genes are transcribed into RNA molecules.

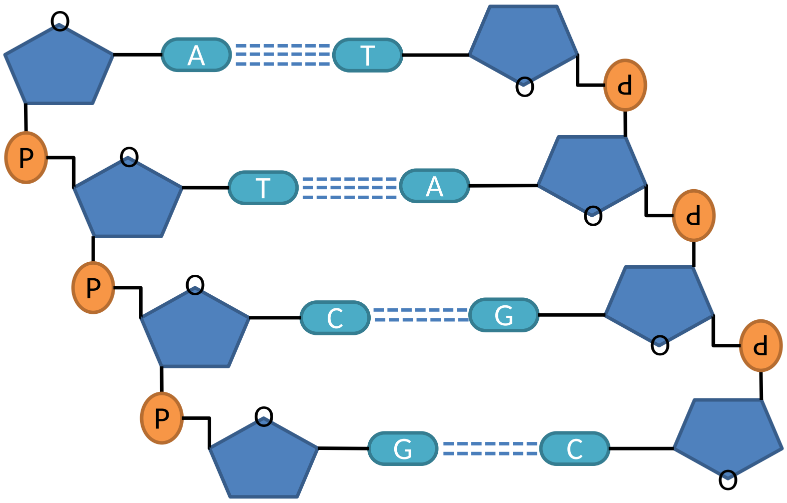


FIGURE 1.1: DNA consists of a deoxyribose backbone (blue pentagons) with phosphate groups connecting between them (orange circles). Strands are combined together by having hydrogen bonds between nucleotides A and T (three bonds) or C and G (two bonds)

1.1.1 Chromatin and DNA compression

Regulation of gene expression can be brought about through DNA compression. As mentioned in Section 1.1, DNA contains millions of bps in various organisms, which results in long sequences. For example, the human genome contains 3,547,762,741 bps[2], and with each bp having the length of 340 picometer ($3.4 * 10^{-10}$ meters), human DNA spans about 1.2 meters in its double-stranded form. Since human cells range in volume of about 500-4000 cubic micrometer ($5 - 40 * 10^{-16}$ cubic meter)[3], they are too small to contain such long DNA, which is why it needs to be folded. For this purpose, proteins known as histones create complexes that DNA wraps around. In brief, histones H2A, H2B, H3 and H4 form dimers amongst themselves and combine together as a histone complex. The DNA wraps around this complex to form a nucleosome, and multiple nucleosomes condense together to form chromatin. Finally, highly-condensed chromatin forms the chromosome, where the entire genetic material of an organism is spread over multiple chromosomes (Figure 1.2).

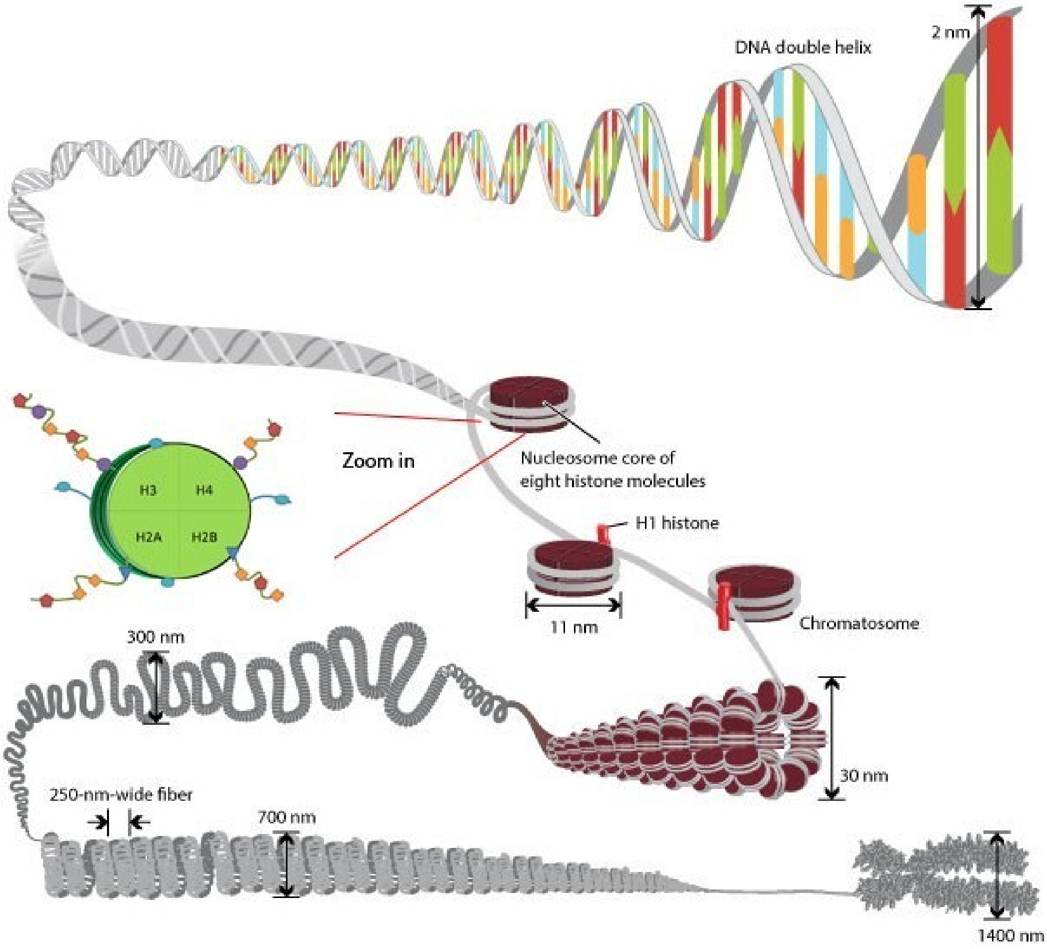
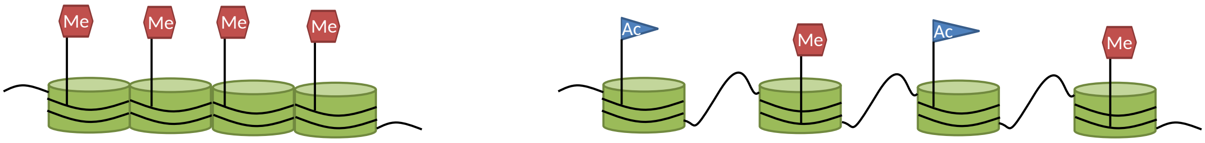


Figure 1.2: The DNA structure begins as a double helix coiled around histone proteins to form a DNA-histone complex known as nucleosome. Nucleosomes cluster together to form chromatin, which is further compressed to form chromosomes. Nucleosome structure is showed with various histone post-translational modifications (HPTMs) (circles, diamonds and so on) that determine the chromatin structure. Figure taken from [4] and modified.

The HPTMs shown in Figure 1.2 determine the chromatin structure, which can be either condensed heterochromatin blocking gene transcription (Figure 1.3a), or lightly-packed euchromatin allowing potential gene transcription (Figure 1.3b). When HPTMs bind to specific histone locations, they change the chromatin confirmation in that area to allow gene expressions to become active or inactive, and those HPTMs can be targeted to study which chromatin regions are open or closed to determine the activity of nearby genes. To study the locations of HPTMs in different cells under various conditions, ChIP-seq can be used, which is discussed further in Section 1.2.1.



(A) Heterochromatin (B) Euchromatin

FIGURE 1.3: Chromatin can take different structures, where heterochromatin is a condensed form with methylation HPTMs (red hexagons), and euchromatin is an open chromatin structure with methylation and acetylation (blue triangles) HPTMs.

1.1.2 Gene expression and functionality

DNA contains a multitude of subunits called genes, where each gene is transcribed to a particular RNA molecule that may be further translated into a protein (Figure 1.4). In more detail, a gene consisting of introns and exons is spliced to form a combination of exons as a messenger RNA (mRNA) transcript, and that is translated into a protein (Figure 1.5). This process, known as the central dogma, was originally defined such that gene transcription and translation necessarily results in a protein. Nevertheless, it was later discovered that only mRNA is converted into proteins, while other types of RNA molecules, generally called non-coding RNAs (ncRNAs), stay as RNA in order to interact with DNA, proteins or other RNA molecules. Those ncRNAs have various functions, for example creating RNA-protein complexes to regulate gene expression. As such, the expression of particular genes is determined as the amount of RNA molecules produced from a certain gene rather than calculating protein amounts, and this can be done using the RNA-sequencing (RNA-seq) technique elaborated upon in Section 1.2.3.

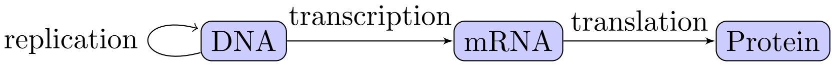


FIGURE 1.4: DNA replication is based on its own template, RNA molecules are transcribed from DNA templates, and proteins are translated from mRNA templates.

1.1.3 Regulation of gene expression

While the amount of RNA directly reflects on genes being transcribed, transcription itself is a complex process that involves multiple factors. One of those factors is the promoter region, which can be found upstream of every gene and contains particular regions where a protein complex must bind to initiate transcription. Specifically, the TATA box region within a promoter must be recognized by a particular transcription factor (TF), which is a protein that can activate or inhibit gene transcription. Another

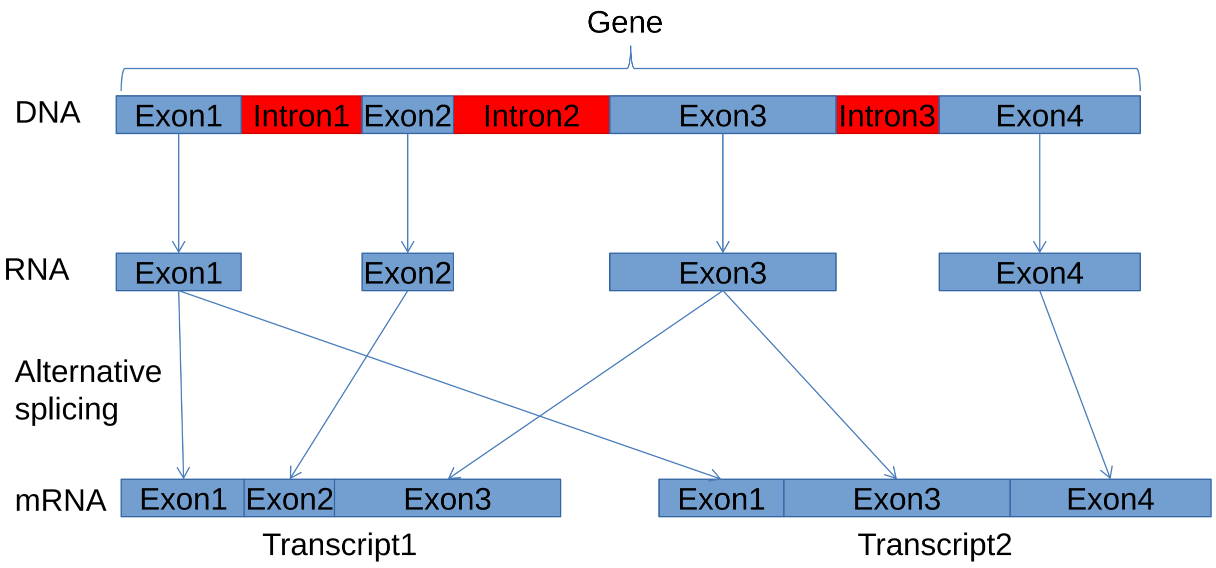


Figure 1.5: Particular species have DNA with intronic and exonic regions, and part of the transcription from DNA to RNA involves splicing introns out of the primary RNA and keeping particular exons in the mature RNA. With many exons, a single gene can be spliced differently to generate various isoforms, which can be translated into proteins with different structures and functions.

important factor for transcription is enhancers, which are distal regions from the gene that can be bound by activator molecules to induce transcription. Once the initial TF binds to the TATA box and the enhancers are bound by activators, other TFs bind to the first TF and create a TF complex, the enhancers region fold the DNA onto it, and RNA polymerase (Pol-II) is recruited to execute transcription (Figure 1.6). Since TFs are proteins binding to the DNA, such interactions are comparable to HPTMs on histones bound by DNA, and as such can be studied with ChIP-seq as well.

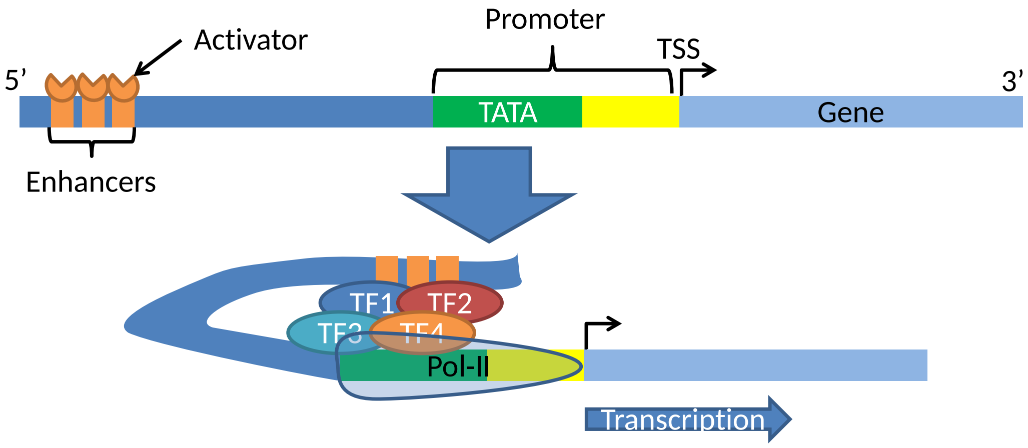


Figure 1.6: Transcription of DNA, conveying the TATA box where particular TFs recognize the promoter region, enhancer regions where activator molecules bind to fold the DNA and bond with the TF complex, and the RNA polymerase 2 (Pol-II) that performs the transcription itself. Once all involved TFs, activators and Pol-II are present, transcription progresses from the 5' end of the DNA to its 3' end to generate RNA

While TFs contribute to regulation of DNA transcription, other factors can regulate it as well. One example of transcription regulation is through 5'-C-phosphate-G-3' (CpG) methylation, where C bases in promoter regions are structurally modified into 5-methylcytosine. With C bases in the promoter changed, the TF complex cannot bind to it, thus inhibiting transcription. In order to study the appearance of 5-methylcytosine within DNA, it is necessary to target them specifically, which methylated DNA immunoprecipitation sequencing (MedIP-seq) does similarly to the protein-targeting done by ChIP-seq (see Section 1.2.2).

1.2 Studying gene regulation and expression with NGS

The first genome to be completely sequenced was single-stranded bacteriophage ϕX174 in 1977 using Sanger sequencing[5], which could be used to target the entire genome or specific regions and genes. Since the year 2000, however, new NGS technologies were developed to replace older Sanger sequencing. Those techniques are particularly advantageous to Sanger sequencing in their throughput and cost, with NGS techniques producing 100-1000 more sequences and costing 25-1000 times less than Sanger sequencing[6]. Therefore, with improved techniques for DNA and RNA sequencing, protein-DNA interactions, methylated DNA and RNA expression could be studied.

1.2.1 Mapping DNA-protein interactions: ChIP-seq

Proteins bound to DNA (TFs) or molecules bound to histones (HPTMs) are studied using ChIP-seq. The first step involves cross-linking between the protein of interest and DNA or the HPTM and the target histone. Cross-linked DNA is then sheared to produce DNA fragments, where some are protein-bound and others are not. This is followed by immunoprecipitation (IP), where an antibody is used to target the protein of interest such that only DNA fragments bound by it are recovered. The resulting fragments are unlinked from the protein and prepared for sequencing (Figure 1.7)[7]. The preparation involves amplifying the sequencing material using polymerase chain reaction (PCR) by separating the DNA fragment strands, using them as templates to copy them and repeat the process to obtain a large amount of DNA. From the resulting fragments, 5'-end sequences of length 24-100 bases are sequenced. Depending on the sequencing machine, the bases in each sequence are read into the machine and created as reads of A, C, G and T with sequencing quality for each base.

One important aspect in ChIP-seq is the shearing step, as it is complicated to standardize across experiments. Shearing, performed using sonication (sound energy to break

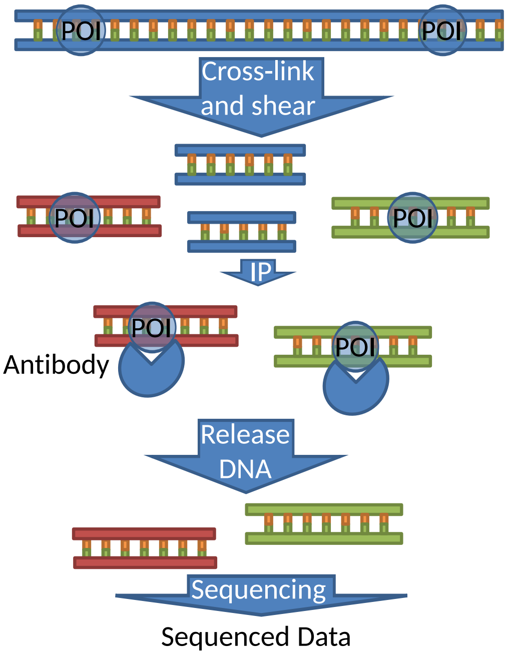


FIGURE 1.7: DNA is cross-linked with a protein of interest (POI, blue circles) and the DNA is sheared into pieces. This is followed by IP, whereby an antibody is used to target specific DNA fragments bound by the POI. Once all protein-bound DNA fragments are captured by the antibody, the POI is removed to return the DNA only, which can go through sequencing to generate the data to be analyzed.

connections in particles), breaks DNA into fragments of 100-800 bps. Other than generating fragments of different sizes, sonication is a stochastic process such that a protein binding to a fragment is not necessarily positioned at its center[8]. In order to mend this, the 5' end sequences being analyzed are established as normally distributed such that the protein is most probably bound in its center. This aspect is particularly important for shift size estimation (Chapter 2) and peak calling (Section 3.4).

1.2.2 Profiling methylation patterns: MeDIP-seq

Methylation of DNA is considered to be a particularly important mechanism in regulation of gene expressions, and as such the MedIP-seq technique was developed. While a previous approach utilized arrays to study methylated regions (methylated DNA immunoprecipitation (MedIP)-chip), it was with the introduction of high-throughput next-generation sequencing that it could be studied with higher depth. Similar to ChIP-seq, the MedIP-seq protocol involves sonicating DNA and targeting the methylation molecule bound to the DNA using an antibody, specifically 5-methylcytidine (5mC) antibody. Once the 5mC antibody is used to retrieve fragments with methylated DNA, they are amplified and sequenced as mentioned in Section 1.2.1[9](Figure 1.8).

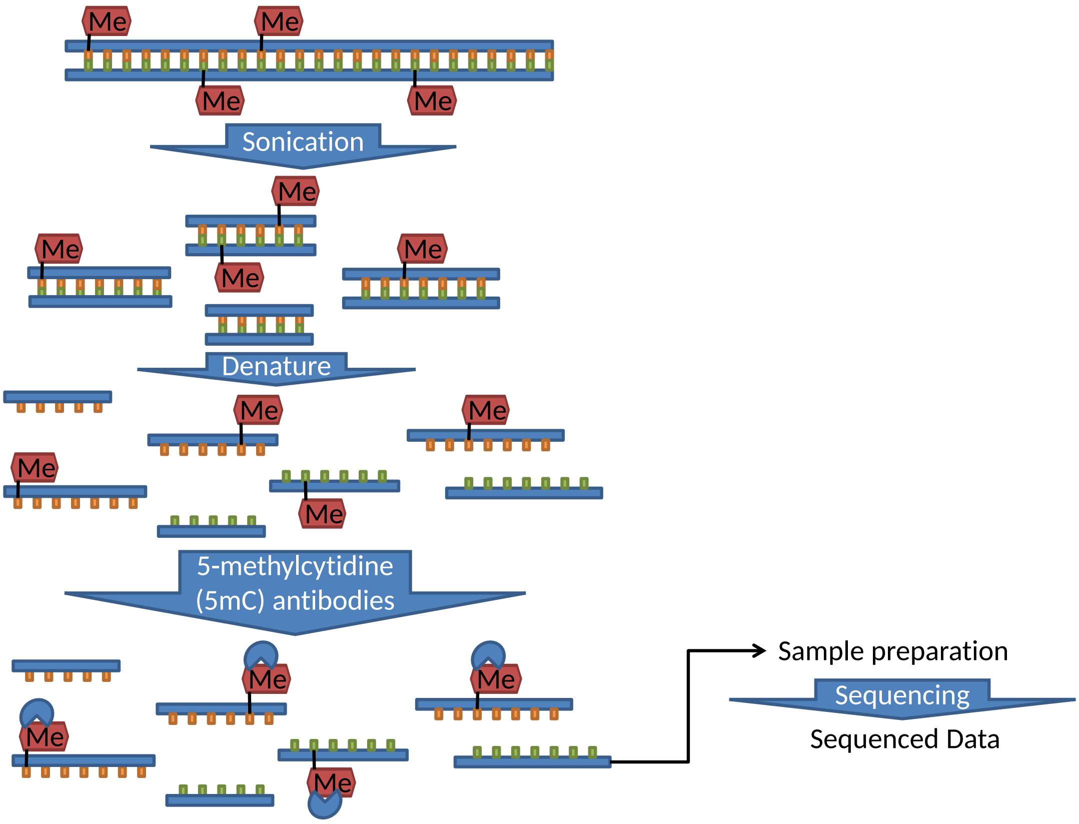


FIGURE 1.8: DNA is sonicated to shear it into pieces, denatured to obtain each strand separately and targeted by 5-methylcytidine (5mC) antibody to obtain only methylated fragments. The methylated fragments are retained and sequenced

1.2.3 Gene and exon expression: RNA-seq

Unlike ChIP-seq and MedIP-seq, RNA-seq is used to target RNA molecules. However, as discussed in Section 1.1.2, there are various types of RNA molecules, primarily coding (mRNA) and non-coding (ncRNA), and RNA-seq is specifically used to target mRNA. mRNAs molecules are known to contain poly-A tails to keep them stable when they are transported from the nucleus to the cytoplasm, while ncRNAs do not. Therefore, the first step of RNA-seq involves targeting mRNAs with poly-A tails using oligo(dT) beads. The resulting mRNA molecules are fragmented and reverse-transcribed to make complementary DNA (cDNA), which are then amplified and sequenced as mentioned in Section 1.2.1 (Figure 1.9)[10].

1.3 Scope of the thesis

Given the multitude of biological studies relying on NGS to study gene expression and regulation under different conditions, the necessity of stable, automated workflows to allow analyzing the data is essential. With this in mind, this thesis focuses on generating

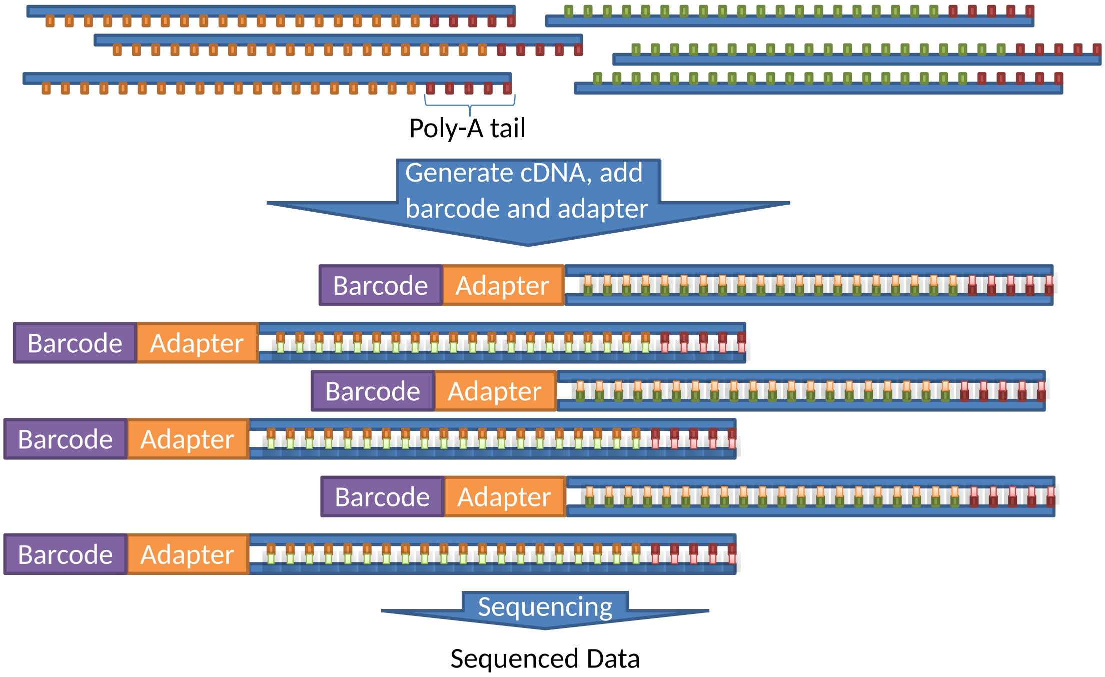


Figure 1.9: RNA-seq used to obtain RNA sequences by targeting their poly-A tail, generating cDNA from them, ligating a barcode and adapter at their 5'-end and sending the resulting sequences to be sequenced

specific workflows and tools to be used by them, and to illustrate how those workflows are used in specific case scenarios. Within the aim of this thesis, different chapters deal with particular aspects of it:

- Chapter 2: Development of a fragment size estimation and enrichment testing tool for ChIP-seq data.
- Chapter 3: Automated workflow designed for ChIP-seq data.
- Chapter 4: Workflows developed for MedIP-seq and RNA-seq data.
- Chapter 5: Case study demonstrating how the developed workflows were used to study the effect of short- and long-term memory on the epigenetic and genetic mechanisms in the mouse brain.
- Chapter 6: Case study illustrating how p73 regulates the development of cilia in mice lungs.

Chapter 2

Fragment size estimation and protein enrichment testing

Interactions between proteins and DNA, primarily chromatin and TF binding, can be studied by extracting DNA fragments bound by target proteins, unlinking the protein from the DNA, followed by sequencing a 5'-end sequence of length 24-100 bps from that fragment (Figure 2.1). Determining the genomic loci of those fragments allows to determine the location a target protein binds to, which can eventually be associated with the expression of a nearby gene. To be more specific, ChIP-seq can be used with either paired-end design, whereby 5'-end tags on both positive and negative strands of each fragment are sequenced, or single-end design, whereby 5'-end tags on either strand of the fragment are sequenced. Among both designs, single-end design is more cost-effective and faster, as it does not require association of tags from the same fragments. Nevertheless, with the paired-end design establishing such a connection, the fragment length is known for each pair of reads, thus allowing a simple estimation of the average fragment length to be used for particular analyses. When utilizing a single-end design for ChIP-seq experiments, however, a fragment size estimation (FSE) algorithm is necessary to ascertain the average fragment length from the distances between positive- and negative-strand tags, and in combination with the read genomic locations, find where the target protein binds to DNA.

Mathematically speaking, given DNA fragments of average length L bps, both fragment ends, corresponding to tag starting positions, are $L/2$ bps away from the protein binding site at the fragment center. Therefore, once the genomic positions of the tags are retrieved, shifting them $L/2$ bases downstream will allow to find the approximate protein

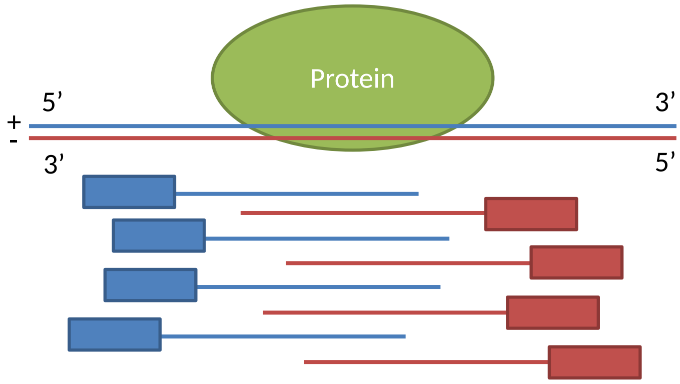


FIGURE 2.1: Positive- or negative-strand sequences appear on either side of the protein. 5’ ends (squares) of the selected fragments are sequenced to form positive- and negative-strand tags that are computationally analyzed.

binding site (Figure 2.2). In order to estimate L, tags on both strands need to be cross-correlated for different shifts (from no shift 0 to a certain shift $> L$), and the shift with highest cross-correlation is the average fragment size, i.e. L.

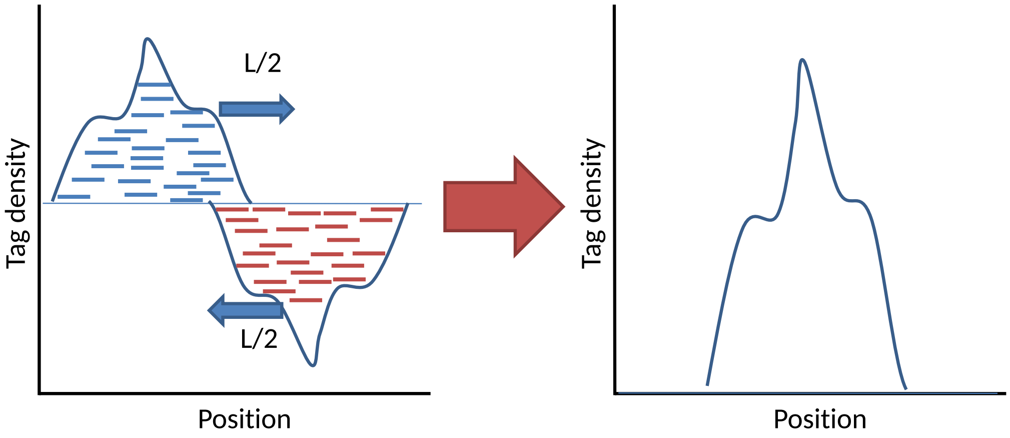


FIGURE 2.2: Tags appear on both DNA strands, where each tag is associated with a particular positive-strand (blue) or negative-strand (red) fragment. With average fragment size of L, shifting positive- and negative-strand tags downstream by $L/2$ allows to find the protein-binding location on fragment centers.

Estimating the fragment size between positive- and negative-strand reads is crucial for ChIP-seq analysis, as it influences ChIP-seq enrichment testing (Section 2.1.3), peak calling (Section 3.4) and downstream analyses. This chapter elaborates upon the manner by which fragment size is estimated and what problems are involved in it. The chapter also discusses the development of *chequeR*, a state-of-the-art *R* package used to estimate the fragment size within ChIP-seq samples, as well as calculate sample IP enrichment. This package has been published as part of a paper by Pena *et. al.*[11] and is currently available for download from Figshare[12].

2.1 Features of fragment size estimation

FSE is expected to consistently find the shift with maximum cross-correlation between positive- and negative-strand reads. However, samples with weak or no IP would have a lack of clear peaks indicating where the target protein binds, leading to a failed FSE. To assess the IP signal quality, some signal-to-noise and signal-to-background ratios were established to evaluate the quality of ChIP-seq data. In an attempt to find the correct fragment size despite low IP enrichment, FSE was incorporated with mappability, which focuses on uniquely-mapped regions to remove the effect of multi-mapped reads on FSE.

2.1.1 Technical background

In order to understand FSE deeper, its mathematical background is illustrated. First, positions $b \in 1, 2, ..., B$ in the reference genome with size B are used as indices for two binary vectors. Each position is mapped to functions $f, g : 1, 2, ..., B \to [0, 1]$, where $f(b)$ and $g(b)$ indicate whether a read start at position b on the positive or negative strand, respectively, is present (1) or not (0). Second, the vector comparison is done between $f(b)$ and $g(b + R - 1 - d)$, with R being read length and d being a particular shift of the negative-strand reads (Figure 2.3). Given this data representation, regular Pearson cross-correlation can be calculated for a particular shift d:

$$r_d(f,g) = \frac{1}{B-d}\sum_{b=1}^{B-d}\frac{[f(b)-\mu_f][g(b+d)-\mu_g]}{\sqrt{V_f V_g}} \approx \frac{\frac{1}{B-d}\left(\sum_{b=1}^{B-d} f(b)g(b+d)\right)-\mu_f\mu_g}{\sqrt{V_f V_g}} \tag{2.1}$$

where μ_f and μ_g represent means of functions f and g, respectively, and V_f and V_g represent variances of functions f and g, respectively. Finally, the estimated fragment size is defined as the shift size d with maximum correlation score, i.e. $d^* = \text{argmax}_d r_d(f, g)$.

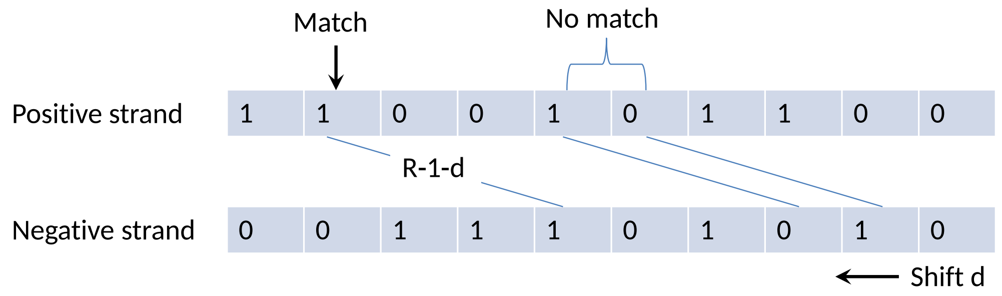


FIGURE 2.3: FSE through cross-correlation between reads on positive and negative strands. 1's and 0's represent positions where read starts are either present or not, respectively, R represents read length and d represents a particular shift size.

2.1.2 'Phantom' peak and mappability

When plotting the cross-correlations r_d as a function of d, ChIP-seq samples with a strong IP signal are expected to have a clear cross-correlation peak at the 'true' fragment size $L > R$. Nevertheless, samples with weak IP would have another peak around read length R, and in some cases it might even be bigger than the 'true' fragment peak, causing the fragment size analysis to return read length as the fragment size (Figure 2.4). This secondary peak around the read length, known as the 'phantom' peak, has recently been traced back to read mappability, which corresponds to the uniqueness of a read aligning to the target genome[13]. Multi-mapped reads are those which map to multiple locations in the genome (also known as unmappable regions), and as such their 'true' genomic locations are ambiguous. During the step of mapping reads to a target genome, such multi-mapped reads are usually removed, resulting in artificially created 0's at unmappable regions. With many such artificial 0 regions, FSE does not distinguish between 0's in unmappable or mappable regions, resulting in strong correlation between 0's in unmappable regions and other 0's at positions that are R bases away on the other strand, resulting in incorrectly estimating the fragment size as R, i.e. the read length.

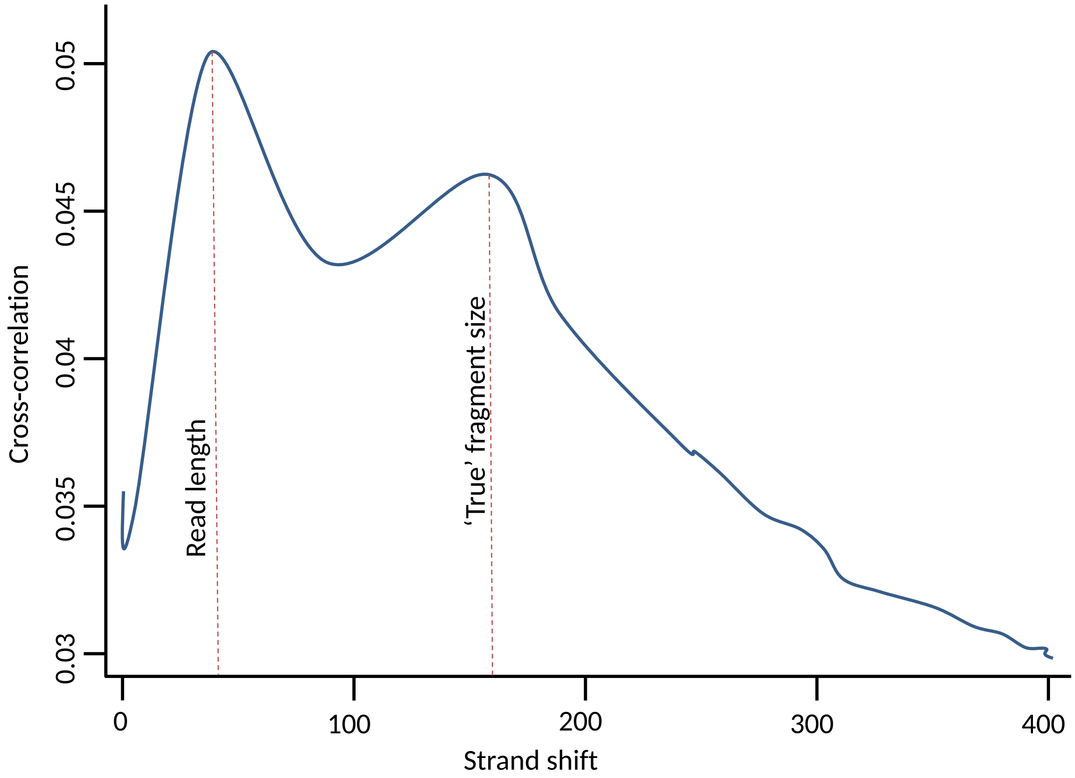


FIGURE 2.4: Plot of the cross-correlation as a function of shifts between positive- and negative-strand reads, indicating the common appearance of the 'phantom' and 'true' fragment size peaks in low quality data.

Previous FSE algorithms have various approaches to deal with this 'phantom' peak, predominantly masking it[14, 15, 16] or calculating local cross-correlation maxima as alternative fragment sizes[17]. However, these approaches have shown to not always return the exact fragment size, as the 'phantom' peak presence distorts cross-correlation values for all shifts, resulting in an incorrect fragment size[13]. As such, a better approach would be to prevent the 'phantom' peak from forming altogether. This can be done by avoiding unmappable regions with potential multi-mapped reads and using uniquely-mappable ones only[13]. Mappability is defined per base for sequences of particular length: given position b and length k, if N is defined as the number of times DNA sequence from position b to $b+k-1$ maps to the genome, mappability score is defined as $\frac{1}{N}$. Therefore, given a sample from a particular organism with reads of length k, the mappability file should contain bases with score of 1, i.e. start positions of reads of length k which map once to the genome[18].

From a technical perspective, the cross-correlation analysis should consider uniquely-mappable positions on both DNA strands. This can be mathematically expressed as $M_R^d = M_R \cap (M_R + R - 1 - d)$, representing the intersection between mappable positions on the positive (M_R) and negative strands ($M_R + R - 1 - d$) for particular shift d. Therefore, the doubly-mappable cross-correlation would be calculated as follows:

$$\rho_d(f,g) = \frac{1}{|M_R^d|} \sum_{b \in M_R^d} \frac{[f(b) - \mu_f^d][g(b+d) - \mu_g^d]}{\sqrt{V_f^d V_g^d}} \approx \frac{\frac{1}{|M_R^d|}\left(\sum_{b \in M_R^d} f(b)g(b+d)\right) - \mu_f^d \mu_g^d}{\sqrt{V_f^d V_g^d}} \tag{2.2}$$

where $|M_R^d|$ is the number of doubly-mappable positions[13].

With the mappability principle utilized to find the 'true' fragment size even in low-quality data, it can be used to calculate quality metrics and evaluate the enrichment of ChIP-seq samples.

2.1.3 Enrichment test

Since 2012, several quality metrics based on the regular Pearson cross-correlation were established to assess the signal-to-noise ratios for various ChIP-seq experiments. The metrics, known as normalized strand coefficient (NSC) and relative strand correlation (RSC), were used to evaluate the enrichment of ChIP-seq samples analyzed as part of the Encyclopedia of DNA Elements (ENCODE) project. NSC is used to evaluate the ratio between fragment size signal and minimum cross-correlation (signal-to-background ratio) and RSC evaluates the ratio between fragment size signal and cross-correlation at read length, i.e. where the 'phantom' peak is expected to appear (signal-to-noise ratio)[15]. Those metrics are calculated as follows:

- $NSC = \frac{cc(fragment\_size)}{min(cc)}$

- $RSC = \frac{cc(fragment\_size) - min(cc)}{cc(read\_length) - min(cc)}$

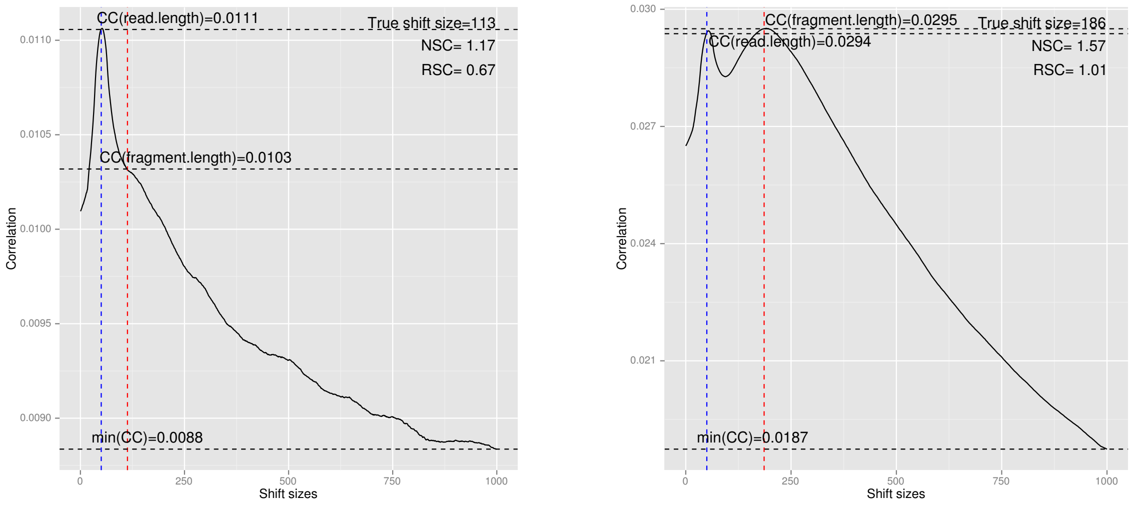
where cc is the cross-correlation value for a particular shift (fragment size or read length) and $min(cc)$ is the minimum cross-correlation value. As the ENCODE project contains various ChIP-seq datasets from different organisms and cell types, the NSC and RSC were used on all of them, and the resulting values were used to derive thresholds for both metrics to define sample qualities. When those were first published, the thresholds used were $NSC < 1.05$ and $RSC < 0.8$ as indicators of low enrichment[15]. Figure 2.5 illustrates how cross-correlation plots with respect to different shift sizes look like for ChIP-seq samples with varying enrichments (Figures 2.5a, 2.5b and 2.5c show samples with low, marginal and high enrichment, respectively). The plots also show the minimal cross-correlation values and those at fragment and read lengths used to calculate the corresponding NSC and RSC values.

2.2 *chequeR*: state-of-the-art package for fragment size estimation and enrichment testing

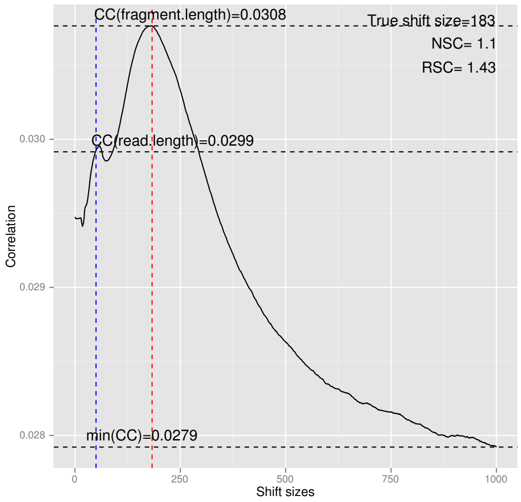
knowledge of mappability allows to remove the 'phantom' peak, RSC and NSC provide enrichment metrics of ChIP-seq samples, and the *R* package *chequeR* was developed to execute FSE while utilizing those features[11, 12]. Overall, this package contains the functions `featureRegions`, `estimateShift`, `smoothCorrs`, `plot` and `chequeRQuality`, addressing different aspects of FSE. The order of utilization for those functions, in addition to the input used in each function, is displayed in Figure 2.6, and elaborated upon in the following paragraphs.

Specific proteins are generally known to bind at particular genomic regions (for example, chromatin modification H3K27ac is known to bind transcription start site (TSS) regions of genes), thus executing FSE on such specific, informative regions could reduce the analysis runtime while still returning the correct result (this feature is elaborated upon in Section 2.2.4). To generate such regions, the function `featureRegions` is used to query the online database *biomaRt*[70] to retrieve such regions given an organism identifier, chromosome name(s), database version, region size and feature of interest (TSS regions, exons, 5' untranslated regions (UTRs), 3' UTRs or transcripts) as input. The resulting regions can then be input into the function `estimateShift`, which executes FSE.

The main function of *chequeR* is `estimateShift`, which is used to execute FSE. Its main parameter is `file`, a binary alignment/map (BAM) file containing alignments of genomic sequences to a target genome. By default, the function generates two binary vectors corresponding to alignment start positions on positive and negative strands



(A) Failed (B) Marginal



(C) Successful

Figure 2.5: Plots of cross-correlations between positive- and negative-strand reads in terms of shift sizes, with vertical lines at the read length and 'true' fragment size and horizontal lines at their equivalent cross-correlations and one horizontal line at the minimum cross-correlation value. The NSC and RSC values were calculated as mentioned before. Plots were generated by own designed *R* package *chequeR*[11, 12]

in each chromosome separately, and calculates the cross-correlation between alignment positions on both strands at different shifts (as mentioned in Section 2.1.1). The cross-correlations at different shift sizes are averaged over all chromosomes, and the shift size with maximum cross-correlation is returned as the 'true' fragment size. While the default function uses all reads for the analysis, five parameters allow to select reads from particular regions: `features`, `map_file`, `nregions` with `size` and `chroms`. Parameter `features` can be set as an object containing genomic regions generated by the function `featureRegions` to execute FSE with reads in those regions. Parameter `map_file` takes in the path to a mappability file in order to estimate the fragment size using reads in

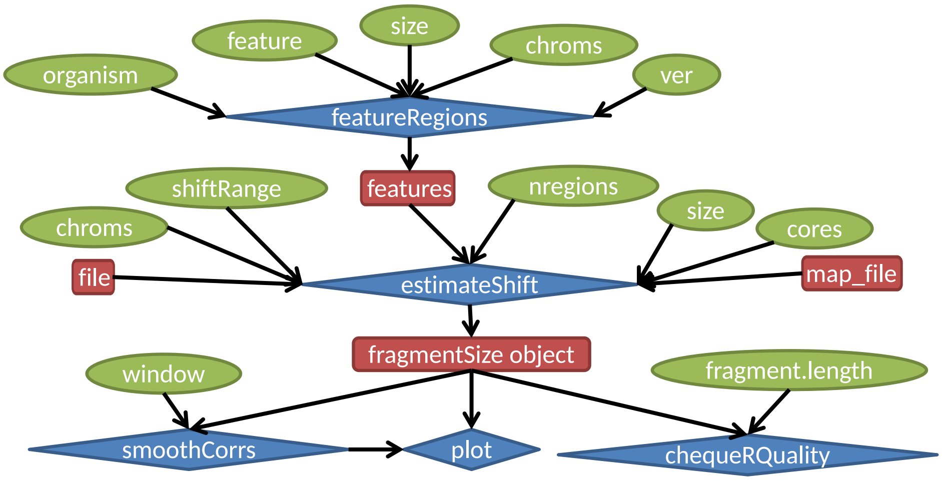
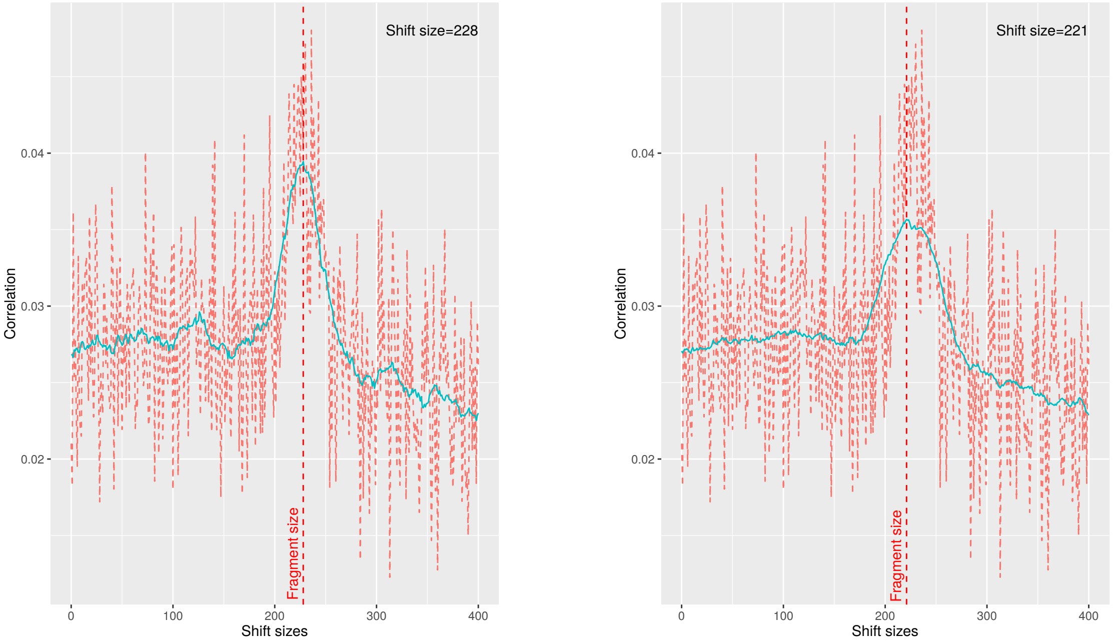


Figure 2.6: Diagram of functions available in *chequeR* package according to the order they should be used. Green ellipses refer to simple numeric or string input, blue diamonds refer to the different functions available in the *chequeR* package and red rectangles indicate complex objects such as files and regions.

uniquely-mappable regions. Parameters `nregions` and `size` can be set together to randomly select a certain proportion of regions (by default, `nregions` = 0.5) of a particular size (by default, `size` = 20000) and use those regions to construct the binary vectors for the analysis (this aspect is elaborated upon in Section 2.2.4). Finally, parameter `chroms` allows to select one or more specific chromosomes to use in the analysis. Other parameters of `estimateShift` include `shiftRange` to determine the range of shifts the estimation is executed for, and `cores` to execute the analysis using multiple cores for a faster runtime.

Once the `estimateShift` function is executed, the resulting `fragmentSize` object can be used to visualize the cross-correlations with respect to the different shift sizes using the function `plot`. Alternatively, function `smoothCorrs` can be executed to smoothen the cross-correlations over a particular window size selected by the user and plot the resulting cross-correlations. By default, `estimateShift` smooths the cross-correlations with a window size 30 in order to obtain a clear maximum cross-correlation point, and the `smoothCorrs` function allows the user to adjust it (Figure 2.7).

chequeR provides the function `chequeRQuality` to calculate the NSC and RSC metrics to evaluate the signal-to-background and signal-to-noise ratios, respectively (as discussed in Section 2.1.3). It is important to mention that this function takes in the `fragmentSize` object generated by `estimateShift` using all reads in the sample, not just the mappable or region-specific ones, as this will allow to evaluate the overall IP enrichment. Another input is the 'true' fragment size found by executing `estimateShift` with mappability and region-specific reads, as the cross-correlation at 'true' fragment size is used to calculate RSC and NSC. With this input, `chequeRQuality` generates a



(A) Smooth window 30

(B) Smooth window 60

FIGURE 2.7: Plots of cross-correlations between reads in a particular sample as a function of different shift sizes. Both show the regular and smooth cross-correlation values, with 2.7a showing the default 30-base window smoothing and 2.7b shows the cross-correlations smoothed with a 60-base window using function `smoothCorrs`.

plot of the cross-correlations as a function of shift sizes, alongside the RSC and NSC values, whether they pass their corresponding thresholds ($NSC \geq 1.05$ and $RSC \geq 0.8$) and the cross-correlation values at the read length, 'true' fragment size and minimum cross-correlation (Figure 2.5).

In addition to the various functions implemented in the *chequeR* package, several particular features within these functions allow a fast, memory-efficient FSE. These include the usage of indexed, random-access binary input and mappability files, bit-vectors used to calculate cross-correlation between strands, use of non-parametric reflective Pearson correlation to provide a statistically valid calculation, selection of informative regions where reads are expected to appear for particular protein targets and the use of multiple computational cores to reduce the analysis runtime.

2.2.1 Indexed, random-access input files

Files containing genomic data are commonly big in size, involving information across multiple chromosomes. While such data can be represented as text files, this would require an algorithm utilizing serial access, which is time-consuming as it has to move through all ordered records preceding a target record. For that purpose, it is necessary to have an approach which allows extracting information from genomic data in a non-serial manner to provide direct access to specific genomic regions in a fast, memory-efficient manner. The solution comes in the form of binary, indexed files with the ability for random access. First of all, the binary form of the data provides a compressed format compared with text-based files, thus requiring less memory to handle the file. Secondly, the indexing of binary files indicates exactly where each data point within the file is located, thus allowing algorithms accessing the file to directly access a specific piece of data within it, rather than going through multiple records in order to reach the target. Having such random access can be particularly useful for FSE, as it can be used to retrieve chromosome-specific reads to allow a per-chromosome FSE, rather than calculating cross-correlation over data from the entire genome.

In the context of FSE with *chequeR*, two parameters used by the `estimateShift` function take in random access indexed binary files: the input BAM file with read alignments and the bigWig file containing uniquely mappable regions. First of all, input BAM files are compressed with blocked GNU zip format (BGZF), which is a block compression that allows random access to BAM files by utilizing indexed queries. While the regular GNU zip (gzip) format condenses the data into a single block, BGZF compresses it into blocks of 64Kb (2^{16} bytes) with each block requiring its own header, thus allowing indexing of blocks with virtual file offsets, where each byte offset from a particular compressed block (position of a byte from the beginning of a block) is mapped to its offset in the uncompressed data. Therefore, this binary file has indexing of blocks which allows random-access to data. Alternatively, the mappability file is based on a wiggle (WIG) text file containing 4 columns indicating coordinates of regions (chromosome, start and end positions) and mappability score for a particular read length in a specific organism. This file is converted into a binary, indexed format, known as bigWig, which the `estimateShift` function uses by means of random access to retrieve uniquely mappable regions in every specific chromosome.

2.2.2 Bit vectors, Boolean operations and C++ integration

Operating systems nowadays utilize particular architectures whereby a certain amount of bits are used to represent a single byte, with the most common ones being 32- and 64-bit systems. Under such operating system architectures, each 0 and 1 in FSE array would require a multitude of bits to represent it, and with millions of bases contained within chromosomes of common research organisms such as human, mouse and drosophila, the strain on memory usage is quite severe. This brought about the idea of utilizing bit vectors within *R* to represent the arrays as single bits, thus reducing the memory footprint by 32 or 64 fold (depending on the architecture used). For example, in a 64-bit system, an array of 0's and 1's representing chromosome 1 from human results in an object of almost 2 terabytes, whereas representing it as a bit vector results in an array of about 30 gigabyte (Gb), which is a significant reduction in size.

Representing arrays in *R* as bit vectors was done using data structure `bit` within a package of the same name, where this package supports both bit vector data structures and allows the logical bit operation AND to establish whether two positions in the bit vector are TRUE. Nevertheless, shifting across bits within this data structure is not straightforward, and requires converting it from one class type to another (bit to numeric). Since this conversion can increase the object size and therefore be a strain on the memory, an alternative approach was necessary. Following some research, we have come across the `dynamic_bitset` class within the *C++* library *boost*, which can store bit vectors and be compared entirely with the AND bit operation (i.e. no need for an explicit `for` loop across all elements in the arrays). The main advantage of this library, however, is the simple operation to shift bits in an array, which can be used within a `for` loop to determine number of read starts appearing on both strands and then shift only those on the negative strand. As such, two *C++* scripts were written in order to count the number of times read starts would appear at positions b and $b + R - 1 - d$ on the positive and negative strands, respectively, with one script written for the analysis of mappable regions and the other for analyzing any region.

Finally, in order to integrate between the *R* code calling reads from the target BAM file in all or particular regions and *C++* script calculating the number of positions having reads on both strands for each shift, the *R* package *Rcpp* was used. It supports mapping *R* objects into *C++* ones and vice-versa, which allowed to convert the `bit` vectors in *R* into `IntegerVector` objects in *C++* to be later converted into `DynBit` arrays. In terms of practical *C++* script integration into *chequeR*, building this package automatically compiled the *C++* scripts and exported them using a `.Call` function in *R*, which allowed the *R* scripts in the package to use them as standard functions.

2.2.3 Statistical validity

In terms of statistical validity, most contemporary fragment size estimators use a regular Pearson cross-correlation as shown in Equation 2.1, which assumes a parametric read distribution. However, sequenced ChIP-seq reads often display a non-parametric distribution [19], which comes across in the models used by various tools to analyze them, including the Poisson distribution used by the peak caller model-based analysis of ChIP-seq data version 2 (MACS2)[17] (further discussed in Section 3.4) and negative binomial distribution that differential expression calculator DESeq2 utilizes[20] (further discussed in Section 3.5). As such, *chequeR* uses a reflective Pearson correlation, which calculates a non-parametric cross-correlation:

$$r_d(f,g) = \frac{\sum_{b \in M_R^d}(f(b) * g(b+d))}{\sqrt{\left(\sum_{b \in M_R^d} f(b)\right) * \left(\sum_{b \in M_R^d} g(b+d)\right)}} = \frac{\sum_{b \in M_R^d}(f(b) \wedge g(b+d))}{\sqrt{\left(\sum_{b \in M_R^d} f(b)\right) * \left(\sum_{b \in M_R^d} g(b+d)\right)}} \tag{2.3}$$

In this equation, the left hand numerator indicates a product between values in doubly uniquely-mappable positions b and $b+d$ on the positive and negative strands, respectively, which is equivalent to an AND bit operation in the right hand numerator, since they are represented by 0 or 1 and can be treated as FALSE and TRUE, respectively. The denominator is a product of read sums on the positive and negative strands, used to normalize the number of cases reads appear on both strands with the total number of reads on both strands. While Equation 2.3 is similar to Equations 2.1 and 2.2 in terms of the relationship between $f(b)$ and $g(b+d)$, it does not depend on the means of those functions at all, thus making it non-parametric and therefore statistically sound in terms of read distribution.

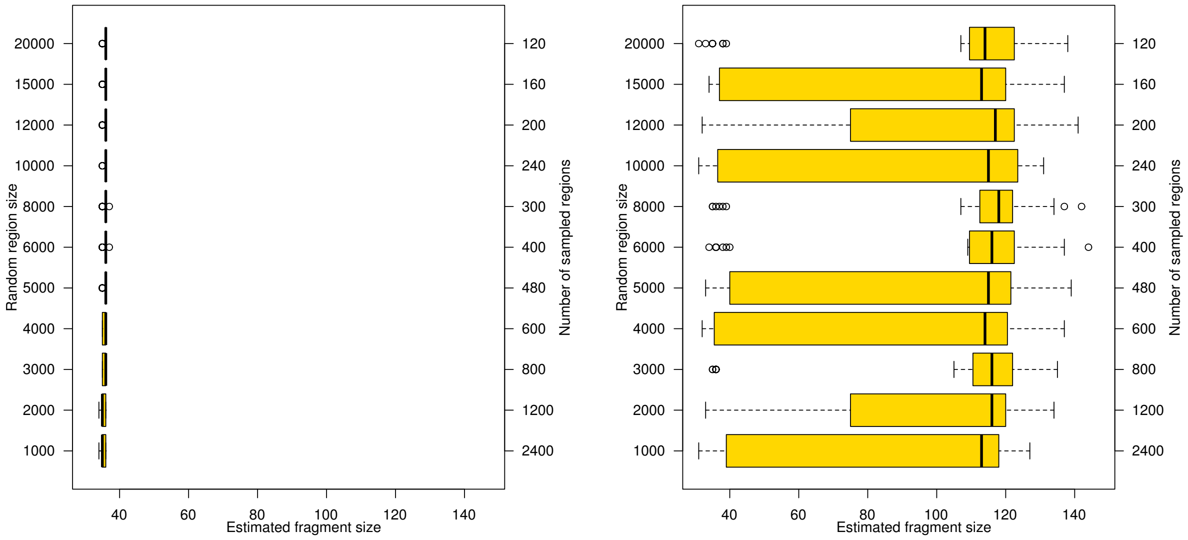
2.2.4 Informative region (sub)selection

Fragment size estimators nowadays are commonly using all reads in the genome or those found in particular chromosomes. However, utilizing reads from the entire genome will result in calculating cross-correlation between two long arrays, resulting in extensive runtime, while the second case may result in an insufficient number of reads to find an incorrect fragment size (especially if the selected chromosome has scarcely any reads). Fortunately, proteins binding to DNA generally bind in very specific genomic regions, thus they are expected to have the majority of reads appearing in them. For example, chromatin markers H3K4me3 and H3K9ac are predominantly localized around TSS regions, while marker H3K79me3 is mostly associated with gene-body binding[21]. Equivalently, TFs are commonly found in TSS regions, as they predominantly regulate gene expressions by binding in the promoter area. As such, focusing on reads in those regions would be advantageous to estimate fragment sizes, as it reduces the size of cross-correlated arrays, thus reducing the overall runtime, while maintaining an informative set of reads to correctly predict the fragment size. To this end, the `estimateShift` function in *chequeR* was implemented with an option for the user to provide particular regions which would be used in FSE.

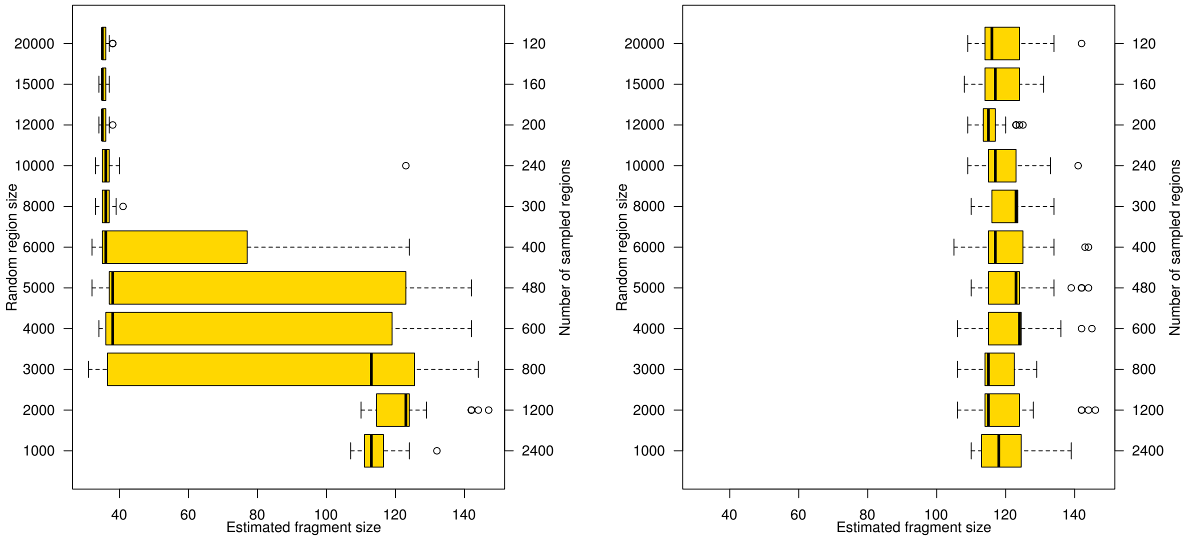
In an attempt to further reduce runtime, the developed FSE was implemented with the option of executing it on a subset of regions. The rationale behind this was an extension of the informative region principle, whereby they are sufficiently informative that not all reads would be required for a correct FSE. This was specifically demonstrated for a set of human, drosophila and roundworm ChIP-seq samples, whereby FSE was executed 35 times on reads from varying random regions of different sizes. Figure 2.8 shows human ChIP-seq sample targeted by H3K9me3, where reads extracted from 120-2400 random regions of sizes 1000-20000 bps, combined such that the resulting regions are of 2.4 giga bps (Gbps), were used to estimate the fragment size. Figure 2.8a using reads from all regions resulted in consistent failure to retrieve the correct fragment size, while Figures 2.8b and 2.8c, using mappable or TSS regions, respectively, show a wide variation in estimated fragment sizes. Nevertheless, Figure 2.8d illustrates that taking TSS regions which are also mappable results in very little fluctuations in the estimated fragment size. Therefore, it indicates that utilizing those randomly selected regions for the analysis, FSE is fairly successful in predicting the correct fragment size.

2.3 *chequeR* memory and runtime benchmark

Utilizing features such as random-access binary files and bit vectors with boolean operations should in principle reduce memory footprint of FSE, while including the option of randomly-sampled informative regions should reduce its runtime. To test this, a human ChIP-seq sample targeting H3K9ac histone modification was used as a benchmark[22]. The comparison was done between *chequeR* using different parameters (whole genome vs. random sampling, with or without mappability and with or without informative regions) and mappability-sensitive cross-correlation (MaSC)[13]. While *chequeR* utilizes binary input and mappability files, MaSC takes as input a text-based BED file with read information for the samples and directory path to text-based per-chromosome map files containing coordinates of uniquely-mappable regions. As such, the memory footprint of MaSC was expected to be higher than for *chequeR*, and as shown by Figure 2.9, this was found to be true. The results showed a maximum resident set size (non-swapped physical memory that a task uses) of 1.352 Gb for MaSC compared with a minimum of 535 megabyte (Mb) for TSS random region selection with *chequeR* (Figure 2.9a). The results also displayed a maximum virtual memory (random access memory (RAM)) of 1.228 Gb for MaSC compared with a minimum of 292 Mb for TSS random region selection with *chequeR* (Figure 2.9b). Interestingly, the memory footprint for *chequeR* when using all reads appears lower than when using mappability and informative TSS regions for the analysis. Nevertheless, this can be explained by the fact that uniquely-mappable and informative regions need to be stored in *R* variables, so despite reducing the overall size of FSE arrays, more variables require storing thus consuming more memory.



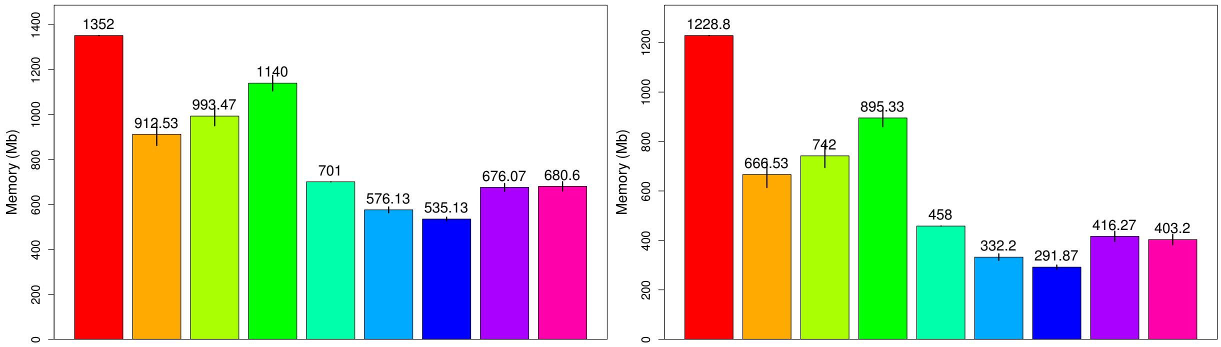
(A) All regions (B) Mappable regions



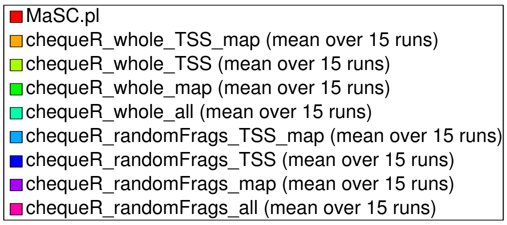
(C) TSS regions (D) Mappable TSS regions

FIGURE 2.8: Box-and-whisker plots for fragment sizes found for human ChIP-seq H3K9me3 sample. FSE is executed for reads in different regions (all, mappable only, TSS only or mappable TSS) where each estimation selects random regions of a particular size, such that the total number of bases covered is 2.4 Gbp (for example, 120 sampled regions of size 20000 bases, 2400 sampled regions of size 1000 bases, and so on). The estimation was executed 35 times for every combination.

The same sample used for memory benchmarking was used to compare the runtime between *chequeR* and MaSC. When comparing both algorithms with a single core, *chequeR* shows a runtime that is mostly lower than MaSC, with a maximum runtime for MaSC at about 20 minutes, and the minimum runtime for TSS random region selection in *chequeR* at about 3.5 minutes (Figure 2.10a). While MaSC can only be executed using a single computational core, *chequeR* can utilize multiple cores in order to calculate the fragment size for several chromosomes at the same time. Therefore, when utilizing 6 cores to execute *chequeR*, its runtime is consistently smaller than for MaSC, whereas



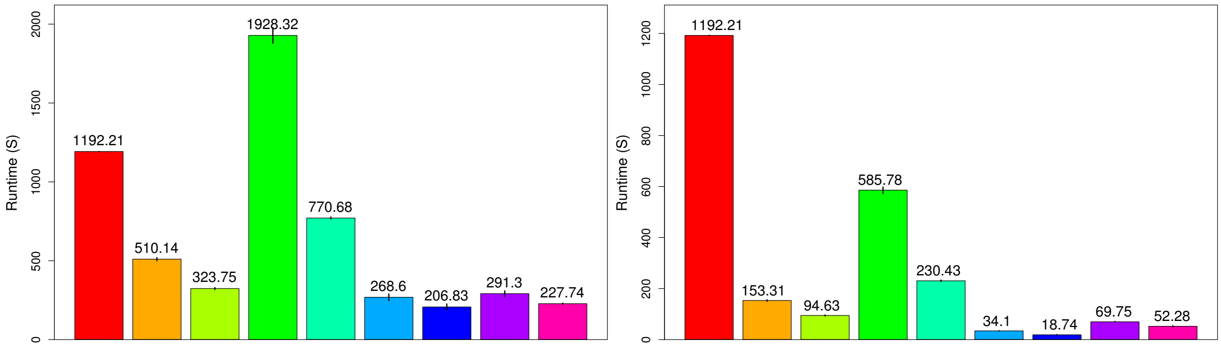
(A) Virtual memory (B) Resident size



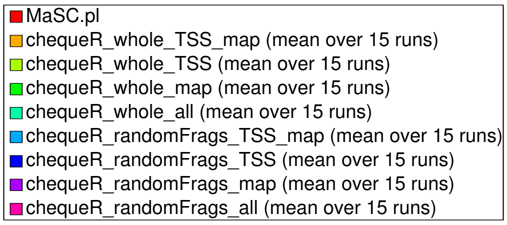
(C) Legend

Figure 2.9: Plots for memory benchmarking in terms of virtual memory and resident size in megabytes (mb) for MaSC and *chequeR* on a human ChIP-seq targeting H3K9ac. Figure 2.9c shows the legend for the different parameters used, specifically for *chequeR*, with “whole” indicating all regions of a particular type, “TSS” to TSS regions, “map” to uniquely-mappable regions and “randomFrags” to 20 randomly selected regions of sizes 10000 bp.

the maximum runtime for MaSC is still about 20 minutes and the minimum runtime for *chequeR* is around 20 seconds using TSS random region selection (Figure 2.10b).



(A) Runtime 1 core (B) Runtime 6 cores



(C) Legend

Figure 2.10: Plots for runtime benchmarking in seconds (S) for MaSC and *chequeR* (using 1 or 6 cores) on a human ChIP-seq targeting H3K9ac. Figure 2.10c shows the same legend shown in Figure 2.9c.

2.4 Overview

The development of *chequeR* was implemented with several features not present in contemporary fragment size estimators before, and managed to maintain good results in terms of both predicting the correct fragment size and having low runtime and memory footprint. The package has been initially published as it was used in a study by Halder *et. al.* on chromatin modifications during learning[21] and was later used in a study involving p73 regulation of multiciliogenesis[23]. In both cases, *chequeR* calculated NSC/RSC metrics and ran FSE to obtain fragments later used for peak calling. The results obtained using from those studies are further discussed in Chapters 5 and 6, respectively. As mentioned before, *chequeR* has also been published as part of a paper by Pena *et. al.*[11] and is currently available for download from the Figshare repository[12].

In terms of further developments to the package, one prominent idea that came up involved using it for the saturation correlation instead of the *R MEDIPS* package (see Section 3.3.4), which is further discussed in Section 7.1.

Chapter 3

Development of ChIP-seq analysis workflow

ChIP-seq is used to study binding of proteins to DNA by generating tens of millions short sequenced reads corresponding to areas nearby protein-binding loci. By determining the genomic locations of those reads, it is possible to find the protein binding locations and therefore ascertain which genes are targeted by it. In order to establish those connections, multiple tools have been developed in recent years to analyze large-scale ChIP-seq data in various stages, including tools for alignment of sequenced reads to the genome, checking data quality, finding enriched regions where proteins are expected to bind and identifying the binding motifs from those regions. With many studies in recent years utilizing ChIP-seq came the necessity for a robust workflow to analyse this data, resulting in multiple instances of such workflows. One example is SeqMonk[24], which is a Java-built program meant to enable visualization of mapped data by allowing importation of alignment files to visualize, quantify and perform statistical analyses to compare regions between them. Another workflow is HiChIP[25], which includes genome alignment, quality control of sequences, identification of both broad and narrow enriched regions (further discussed in Section 3.4.1) and downstream motif finding and functional enrichment analysis. Finally, the workflow framework Galaxy[26] provides several workflows dedicated to ChIP-seq, where its main steps include genome alignment, enriched region detection and visualization of the data. Although there are multiple workflows for ChIP-seq, they do not provide sufficient flexibility for the analysis and generally are fairly slow in their execution. One example is that contemporary ChIP-seq workflows are incapable to switch between samples from single or multiple conditions.

ChIP-seq data can be analysed for a single condition or multiple conditions, with different aims. When a single condition is considered, the purpose is to find where the target protein is binding in order to better understand what genes, and therefore biological mechanisms, it associates with. For example, our study on TAp73-targeted ChIP-seq human samples was executed in order to determine the association of p73-bound genes with those deregulated by p73 knock-out (KO)[23]. Alternatively, when two or more conditions are considered, a differential occupancy (DO) analysis might be implemented to determine differences between binding of proteins in different conditions. For example, our study on cell-type specific epigenetic changes in learning mice used ChIP-seq data to determine which chromatin marks are enriched between naive and learning mice[21]. Therefore, the analysis of ChIP-seq data requires a standardized workflow that would be flexible in its ability to handle both single- and multi-condition analyses, while being robust and fast to execute the data analysis easily, quickly and efficiently.

This chapter focuses on the development of a ChIP-seq analysis workflow, providing both DO (Figure 3.1a) and binding analysis (Figure 3.1b) variants. Both variants start with FastQ files corresponding to different samples and containing sequenced reads, which are aligned to the target genome with bowtie2[27] and Samtools[28], and finally filtered to generate high-quality BAM files. To check data quality, FastQC[29] and *R* packages *chequeR*[11, 12] and *MEDIPS*[30] are used on the high-quality BAM files. In case several replicates are involved, the BAM files are merged, which is used to generate a bigWig file using *MEDIPS* and the script `wigToBigWig` to visualize the data in a genome browser. The merged BAM file is then input into *chequeR* to estimate the fragment size (as described in Section 2.2) and peaks are called using MACS2[17] with the BAM file and fragment size as input. In case a single condition is present for all samples, the peaks can be used for motif discovery with hypergeometric optimization of motif enrichment (HOMER)[31], and/or for functional enrichment analysis by annotating them with nearest genes using an in-house Perl script `regionAnnotations` and submitting the resulting genes to web-based gene set analysis toolkit (WebGestalt)[32]. Alternatively, if multiple conditions are present, the samples can be used to execute DO analysis with *DESeq2*[20] and the DO peaks can be used for motif discovery and/or functional enrichment analysis.

3.1 Genome alignment

The first step in the analysis of ChIP-seq sequences involves finding their genomic locations, which is a critical step influencing all downstream analyses. In brief, digital reads corresponding to sequenced tags are aligned against the target genome in order to determine their most likely location. While various tools have been developed for genome alignment utilizing alternative approaches, the tool selected for the ChIP-seq workflow is bowtie2[27], which aligns shorter seeds of reads to the genome using end-to-end alignment (complete seed is mapped to the genome), extends the aligned seeds and assigns a probability for each alignment being correct.

3.1.1 FastQ format

The input for an alignment algorithm is a FastQ file, which is the result of sequencing tags within a sample to generate a file with millions of reads corresponding to those tags. The FastQ format consists of 4 lines per read, corresponding to the read name, its sequence represented as strings of A, C, G and T, an optional line with the read name, and the sequencing quality of each base in the read. Base qualities are represented by the Phred quality score, which corresponds to the probability of an incorrect base (nucleotide) being called during sequencing, and those probabilities are used to calculate overall sequencing quality (discussed further in Section 3.3.1). Furthermore, the sequencing quality may be utilized in the alignment step to filter out reads with overall low sequencing quality, thus preventing false-positive alignments.

3.1.2 Bowtie2 usage and parameters

Bowtie2 was found to be the best option for the workflow due to its runtime and memory performance compared with other alignment tools such as Burrows-Wheeler Aligner (BWA)[33], BWA's Smith-Waterman alignment (BWA-SW)[34], short oligonucleotide alignment program 2 (SOAP2)[35] and Bowtie[36]. In terms of runtime and memory footprint, bowtie2 showed good performance by finding high alignment rates at comparably shorter time and around the same, if not lower, memory footprint[27]. One of the features bowtie2 possesses is usage of full-text minute (FM)-index, which compresses the target reference genome while still allowing quick substring searches, thus keeping the memory footprint small. In addition, bowtie2 is able to handle gapped and local alignments to account for possible insertions/deletions in the sequenced material. Finally, it can utilize multiple processors to reduce the analysis runtime.

Once bowtie2 was selected as the tool used to execute read alignment, it was necessary to determine which parameters would optimize the alignment results. Given that bowtie2 showed best performance in terms of runtime, memory footprint and alignment percentage when the default parameters were used in the paper by Langmead *et. al.*[27], the same set of parameters were used in the ChIP-seq workflow. Implementing bowtie2 into the pipeline involved default parameters, which means the complete seeds (end-to-end) built from reads are aligned to the genome without any mismatches (-N 0). In order to confirm the choice in parameters used by bowtie2, two particular metrics reported by the alignment were considered: its score AS and uniqueness reflected by MAPQ. The score, calculated based on gap and mismatch penalties, reflects how well reads align to the genome, whereas MAPQ is derived from the probability an alignment is wrong based on how many times a read aligns to the genome.

3.1.2.1 Reporting alignments bowtie2 vs. bowtie1

In terms of reporting the highest scoring alignment, its score AS is important. In the default end-to-end mode, AS starts from 0 and gets penalized for gaps (-5 for gap opening, -3 for its extension) and mismatches (-6), thus `AS:i:0` indicates an exact alignment of the read to a particular genomic location. By default, multi-mapped reads would have a flag `XS:i` showing the second best alignment score, where it could be equal to or smaller than the best one. In situations where two alignments have the same best alignment scores (i.e. `AS`=`XS`), it was necessary to understand which alignment is chosen by bowtie2, and this was studied by benchmarking it under different parameters and against bowtie1.

As mentioned before, reads can align to multiple locations in the genome, where each locus has its own alignment score. When an aligner is set to report a single position a read aligns to, the results usually show the single best alignment, even if the read aligns to multiple positions. When the first and second best alignments have different scores, the best alignment should clearly be reported, but when they share the same score, the correct alignment position becomes ambiguous. In order to illustrate the usefulness of bowtie2 as an aligner in reporting the best alignment, it was benchmarked using various sets of parameters and compared with bowtie1 under different parameters. For the various parameter combinations, an example FastQ file with 3 reads associated with the mouse *mm10* genome were to be aligned: (1) multi-mapped read with one best alignment, (2) multi-mapped read with multiple best alignments and (3) uniquely mapped read. Given those reads, bowtie2 and bowtie1 were executed with the different parameters in order to determine which aligner reports the most accurate results, and under which combination of parameters.

To study the reads being reported for bowtie1 and bowtie2 for different parameters, several combinations were selected to execute the example FastQ file (Table 3.1).

With those parameters, bowtie1 and bowtie2 were executed for the example FastQ file to generate small sequence alignment/map (SAM) files showing how the different reads align. As can be seen from Table 3.2, bowtie2 generally has a better capability of showing the MAPQ compared with bowtie1, thus giving the user a better judgment of the alignment quality for various reads. Furthermore, bowtie2 returns the MAPQ

Bowtie version	Alignment parameters	Description
bowtie2	`-t -N 1 -k 1`	1 mismatch, reporting 1 alignment without considering MAPQ, with `-t` records runtime for different steps
	`-t`	Default (0 mismatches, report 1 best alignment with MAPQ)
	`-t -a`	0 mismatches, report all alignments
bowtie1	`-l 25 -n 1 -m 1 --best`	1 mismatch, seed length 25, report uniquely best aligned read (m=1, –best)
	`-v 2 -m 1 --best`	report end-to-end hits with ≤ 2 mismatches (v=2), report uniquely best aligned read
	`-a -m 1 --best`	report all alignments per read (-a), uniquely best aligned read
	`-a --best`	report all alignments per read
	`-n 2 -e 70 -l 28 --maxbts 800 -y -m 1 --best --strata --phred64-quals`	Default except m=1 and phred64-quals (Input qualities are ASCII chars equal to the Phred quality plus 64)

Table 3.1: Parameters tested for bowtie1 and 2 benchmarking

scores when it is not forced to report 1 alignment exactly (`-k 1`) and is easier to handle when it does not report all secondary alignments (`-a`). As such, bowtie2 shows a better capability to return the alignment quality and a minimal set of best alignments when executed with default parameters.

3.1.2.2 Read filtering with mapping quality

Another aspect important to reporting alignments is mappability quality MAPQ, which is defined as $MAPQ = -10 * log_{10}Pr$(mapping position is wrong)[28]. Given this calculation, as the probability of a particular mapping position to be wrong increases, MAPQ decreases, where a certainty that the mapping position is wrong (i.e. 1) results in $MAPQ = 0$. In order to better understand the impact of MAPQ on results obtained from ChIP-seq data, a comparison was done between uniquely-mapped and high-quality reads in the same data. Alignments can be distinguished based on their quality (high or low) and their mapping (uniquely- or multi-mapped) using MAPQ, as shown in Table 3.3.

Since multi-mapped reads are ambiguous in their genomic alignment, they were often removed in various NGS studies by using uniquely-mapped reads only. However, in the context of ChIP-seq data, various HPTMs and TFs can be found in intergenic regions, which are repetitive, thus removing multi-mapped reads will ignore such regions and prevent from finding additional targets of protein binding. Therefore, in order to

Read Name	Bowtie version	Parameters	NumAligns	AS	XS	MAPQ (for best position)
read1	bowtie2	`-t -N 1 -k 1`	1	0	*	255
read2			1	0	*	255
read3			1	0	*	255
read1		`-t`	1	0	-13	34
read2			1	0	*	42
read3			≥ 1	0	0	1
read1		`-t -a`	6425	0	*	32
read2			1	0	*	255
read3			5326 (40 best)	0	0	1
read1	bowtie1	`-l 25 -n 1 -m 1 --best`	1	*	*	255
read2			1	*	*	255
read3			0	*	*	*
read1		`-v 2 -m 1 --best`	0	*	*	*
read2			1	*	*	255
read3			0	*	*	*
read1		`-a -m 1 --best`	0	*	*	*
read2			1	*	*	255
read3			0	*	*	*
read1		`-a --best`	3	*	*	0
read2			1	*	*	255
read3			1384 (56 best)	*	*	255
read1		`-n 2 -e 70 -l 28 --maxbts 800 -y -m 1 --best --strata --phred64-quals`	1	*	*	255
read2			1	*	*	255
read3			0	*	*	*

TABLE 3.2: Testing parameters in bowtie1 and 2

determine the effect of multi-mapped reads on downstream results, it was necessary to compare those generated for uniquely-mapped and high-quality alignments. We executed this comparison in our lab with data we used in our Halder *et. al.* publication[21]. The read sets were created such that set (1) had alignments with MAPQ 0, 2, 3 or 4 filtered out (high-quality read set), and set (2) had alignments with MAPQ 0 or 1 filtered out (uniquely-mapped read set). The resulting BAM files for high-quality and uniquely-mapped reads were analyzed by calling peaks for them (Section 3.4) and finding differential histone post-translational modifications (DHPTMs) for different conditions (Section 3.5). When comparing the results, both peak calling and DHPTMs were fairly similar, indicating that using high-quality reads, including multi-mapped ones, does not generate worse results than uniquely-mapped reads, thus justifying its use. As such, the ChIP-seq workflow was implemented with filtering of low-quality alignments by removing those with MAPQ values of 0, 2, 3 and 4 from the SAM file, followed by creating and indexing the BAM file as one with high-quality reads.

MAPQ	Pr(mapping position is wrong)	Classification
0	1	Multi-mapped Low-quality
1	0.7943282	Multi-mapped High-quality
2	0.6309573	Uniquely-mapped Low-quality
3	0.5011872	Uniquely-mapped Low-quality
4	0.3981072	Uniquely-mapped Low-quality
5	0.3162278	Uniquely-mapped High-quality
6	0.2511886	Uniquely-mapped High-quality

Table 3.3: Table with MAPQ values, their corresponding probability of the mapping position being wrong and the classification as uniquely- or multi-mapped and low or high quality (derived from the forum discussing MAPQ scores calculated from bowtie2 alignments).

3.2 Data visualization

One of the most important results in ChIP-seq data is finding binding regions of proteins to DNA. While there are various tools used to detect those regions automatically (see Section 3.4), it can be useful for the user to have a visual confirmation that those regions are indeed enriched. For that reason, the ChIP-seq workflow has been implemented with a technique to convert large, alignment BAM into random-access bigWig files which can be imported into various genome browsers, such as integrated genome viewer (IGV)[37], integrated genome browser (IGB)[38], University of California Santa Cruz (UCSC)[39] or a custom genome browser developed by our lab[21]. The first step involves importing BAM files into *R* using the *MEDIPS* package[30], binning the reads over windows of 50 bps and normalizing them over the library size to obtain reads per million (RPM) counts in each bin. These are then exported as WIG files and converted into binary bigWig files using wigToBigWig script from UCSC. An example of the custom genome browser developed by our lab can be found in Figure 3.2, where clear peaks can be seen for ChIP-seq samples targeted by HPTMs H3K27ac and H3K4me3.

3.2.1 Replicate BAM merging

In order to obtain results that can be considered as statistically sound, it is a common practice to sequence replicate ChIP-seq samples. However, when dealing with a multitude of replicates, visualizing and comparing them on many different tracks can be difficult, and even more so when comparing replicates from multiple conditions. For that reason, merging replicates would result in a single track, making it easier to visualize samples from one or more conditions. Practically, replicates are merged at BAM file level, using the command ‘samtools merge’ to combine multiple replicate BAM files into a single file. While this approach is sensible for technical replicates, as it reduces their distinct technical errors, it is somewhat questionable when dealing with biological replicates, since it tends to reduce the biological variability between them (this is further discussed in Section 7.2.2).

3.3 The importance of quality control

When preparing samples for ChIP-seq processing, various technical and biological errors may occur, including improperly kept samples, incorrectly followed protocols, sample contaminations, high biological variance and sequencing errors. In addition, the IP strength (affinity between antibody and target protein) can deviate from one protein to another and between samples, which is why IP strength should be evaluated. In addition, within the specific context of ChIP-seq data, quality of the samples can be reflected by the IP enrichment, indicating whether the signal coming from the data is a reflection of the protein-DNA binding or simply noise due to weak IP. While the results generated for particular analyses are important, their validity depends on their quality. As such, various tools have been developed to assess quality of ChIP-seq samples, and this section deals with particular quality control (QC) tools used specifically to evaluate the quality of ChIP-seq samples.

3.3.1 Determining sequencing quality: FastQC

One aspect of sample quality that can be looked at is the sequencing quality. As mentioned in Section 3.1.1, FastQ files contain Phred quality scores representing the probability of incorrectly calling bases, with $Q = -10log_{10}P$ calculates the Phred score Q in terms of the probability P. While the calculation displays a straightforward relationship between the score and probability, different sequencing companies have their own formats and ASCII encodings of the FastQ scores. Therefore, the tool used to assess sequencing quality ought to recognize the quality encoding and be able to handle it accordingly. Fortunately, FastQC[29] is able to recognize the encoding automatically and produce a set of results pertaining to the sequence quality. This includes an overall file summary, per-base, per-tile and per-sequence qualities, per-base sequence, GC and N contents, sequence lengths, duplication levels and overrepresented sequences (any, adapter and k-mers). When provided with a FastQ file of a particular sample, FastQC returns a text file containing the numbers pertaining to a specific criterion, an HTML file with the plots generated from those numbers and a text file indicating for each criterion whether the sample gets a PASS, WARN or FAIL mark. While FastQC cannot denote a sample as overall good or bad, the abundance of criteria with WARN and FAIL should be a warning sign for bad quality. For example, low quality data tends to have reads with low per-base quality overall (Figure 3.3a), whereas high quality data will tend to have high quality bases (Figure 3.3b).

On a sequence-based level, when considering the average quality of all reads, FastQC returns plots with the number of reads with a particular average quality, where good data would show many reads with high quality scores (Figure 3.4b), whereas bad data might show several reads with lower quality on average (Figure 3.4a).

3.3.2 Alignment rate and genomic coverage

When aligning millions of reads to the reference genome, some restrictions are often required in order to prevent the consideration of low quality reads in downstream analyses. As such, not all reads are aligned to the genome and 100% alignment rate is practically non-existent. At the same time, having a low percentage of reads aligning to the genome indicates many erroneous reads (due to sample contamination, errors during sequencing, or other such issues). Therefore, the minimum alignment rate considered to indicate a good quality sample is 70%. Furthermore, in the context of using high-quality alignments (Section 3.1.2.2), the workflow returns proportions of high-quality alignments and uniquely-mapped reads to get an idea of alignment qualities within all mapped reads (how many alignments are high-quality compared with uniquely-mapped).

In addition to the alignment rate, the average base coverage is considered to determine how much of the genome is covered by reads, calculated as $C = N * L/G$, where C is the coverage, L is the read length, N is the number of used reads in the data (i.e. high-quality reads) and G is the genome size[40]. It should be acknowledged, though, that ChIP-seq is not sequenced over the entire genome but over specific protein-binding regions, so its coverage is expected to be less than 1.

3.3.3 ChIP-seq enrichment

The IP strength is important to determine how well the protein binds to DNA (As discussed in Chapter 2). By treating reads originating in the area of protein-binding as signal and all other sequenced reads as noise/background, it is possible to refer to this enrichment test as a signal-to-noise or signal-to-background ratio. As mentioned in Section 2.1.3, those ratios have already been devised in the form of NSC and RSC[15], and their calculation was implemented into the *chequeR* *R* package[11] (Section 2.2).

As calculating NSC and RSC statistics for a particular sample requires the correct fragment size, the enrichment test implementation into the ChIP-seq workflow involved both FSE and the NSC and RSC calculations. The FSE was implemented by using all reads from doubly-uniquely mappable regions within 2500 bps around the TSS, and executing the estimation on shifts 0-1000 using a single core. Once the fragment size was obtained, the code checks whether the true fragment is smaller than double the read length, as this would reflect a 'phantom' peak event (see Section 2.1.2). In such a case, the script would search for a shift with maximum cross-correlation and size bigger than twice the read length, and determine it to be the 'true' fragment. This would then be used to generate the NSC and RSC statistics.

3.3.4 Saturation and pairwise correlations: *MEDIPS*

Similarly to the enrichment test checking the signal-to-noise ratio, the saturation test checks the IP signal strength by determining whether the sample coverage in terms of number and spread of reads across the genome is deep enough to call peaks from. The saturation analysis is performed using the *R* package *MEDIPS*[30] to find the saturation correlation for each sample. In brief, the function `MEDIPS.saturation` divides the sample reads into 2 equally-sized sets, and calculates Pearson correlations between subsets of randomly-selected reads from those sets (0% to 100% of the reads with steps of 10%). This establishes the correlation profile of the sample by showing the relationship between increasing read number and the increased corresponding correlation. While the saturation analysis calculates the correlation between all reads n within a sample, at most the correlation is calculated between two read sets of size $n/2$. In order to create a correlation with n reads, they are artificially inflated by duplicating each read once, resulting in $2n$ reads. Using the same Pearson correlation between read subsets, the resulting profile is of an estimated correlation, where this saturation analysis calculates the maximum correlation between two sets of size n.

In the ChIP-seq workflow implementation, the function `MEDIPS.saturation` was used with sub-regions of size 100 bps and unique reads extended by 200 bps. The created object was fed into the function `MEDIPS.plotSaturation`, resulting in plots based on the actual and estimated correlation profiles as functions of the read subsets (Figure 3.5), showing the difference between actual and estimated saturations. Figure 3.5a displays a sample with low saturation, showing a large deviance between the actual and estimated saturations, whereas Figure 3.5b shows a highly saturated sample where the actual and estimated saturations are similar.

While the saturation test compares the coverage of a particular sample with the target genome, the pairwise correlation compares coverages between replicates to determine their similarity. As such, the pairwise correlations between samples can be calculated in order to decide whether replicates group together by condition or not. In order to calculate pairwise correlations between replicates, the Pearson correlation was used as part of the `MEDIPS.correlation` function using the same parameters as used for the saturation analysis. The result was a single table showing the correlations between all samples. Nevertheless, while this technique is useful when few replicates are present, a visual assessment of similarity between them might be easier when dealing with a multitude of samples. For that reason, a principle component analysis (PCA) plot has been implemented for the ChIP-seq workflow, where it reduces the multi-dimensional representation of the coverage profiles of samples into two dimensions in order to represent the similarity between them. The PCA was generated by first generating binned counts for different samples using the function `MEDIPS.createSet` with the same parameters as before and writing them as a matrix into a text file with one column per sample. The file would then be read back into *R* and converted into a `DESeqDataSet` object so that it could be transformed with the regularised log transformation[20]. Finally, package *FactoMineR*[41] would be used to calculated the principle components and plot them. The resulting PCA plot would then allow to find any distinct outliers from the other samples, as well as distinguish between them if they associate with different conditions. Figure 3.6 illustrates a PCA plot of published GEO dataset GSE15780, which contains ChIP-seq samples from human osteosarcoma (bone cancer) Saos2 cells targeting protein isoforms TAp73α and TAp73β[42], showing a clear separation between TAp73α and TAp73β samples[23].

3.3.5 Quality control overview

With all aspects of QC described above, the ChIP-seq workflow was implemented with a python script to generate a single text file with all results put together (Table 3.4). The table contains initial number of reads in each tested sample, numbers and percentages of mapped reads, high-quality alignments and uniquely-mapped reads, average base coverage, NSC and RSC statistics, saturation and replicate correlations.

3.4 Peak calling

Finding genomic regions where proteins bind to DNA is one of the main aims of ChIP-seq analysis, and this is when peak calling is utilized. This involves locating regions where the signal derived from sequenced reads is significantly higher than the background noise. This procedure divides into two main steps, the first being FSE of ChIP-seq tags and the second being peak detection itself. While various peak callers have been developed in recent years, the current ChIP-seq workflow was implemented with MACS2[17], as it has been used in various studies with different protein targets and often resulted in results

Sample Name	Raw reads	Mapped Reads	Percent High-Quality Uniquely and Multimapped Reads	Percent Not Truly Multimapped Reads	Average Base Coverage	NSC	RSC	Saturation Correlation	Replicate Correlation
input	19715835	19258395	98.99	83.14	0.30	1.19	0.48	0.62	NA
p73alpha_01	18437574	17680150	99.32	84.35	0.28	2.7	1.89	0.78	0.73
p73alpha_02	18753658	17874666	98.98	83.95	0.28	1.66	1.08	0.64	0.73
p73beta_01	18189000	17706509	99.34	84.14	0.28	5.31	2.22	0.86	0.89
p73beta_02	17954381	17302936	99.1	84.75	0.27	4.49	1.84	0.84	0.89

Table 3.4: Example reduced table of QC results previously generated for our data in the Nemajerova *et. al.* study[23], where samples were taken from human cancer cells and targeted by TAp73 isoforms (for more information see Chapter 6).

that could be validated biologically as true. Before executing MACS2, *chequeR* is used with the same parameters as in Section 3.3.3 to estimate fragment size d from a particular BAM file, and the calculated fragment size and BAM file are input into MACS2. In brief, MACS2 takes in reads from a particular BAM file, removes PCR duplicates (false signals) when an input ChIP-seq BAM file is provided, shifts positive- and negative-strand reads by $d/2$ downstream, slides windows of size $2d$ across the genome and finds candidates with a significant tag enrichment compared with a background Poisson distribution. Once the p-values are obtained for each peak region, MACS2 adjusts them using false discovery rate (FDR) correction to account for the multitude of peaks obtained. The implementation of MACS2 uses mostly default parameters, with d being previously calculated by *chequeR* and the FDR threshold set to 0.05 (i.e. only peaks with adjusted p-value lower or equal to 0.05 are kept). The peaks can be used for downstream analyses, including finding which TFs bind to them (motif enrichment), identifying differentially occupied peaks under different conditions or determining which biological terms are enriched for genes found nearby the peaks.

3.4.1 Broad vs. narrow peaks

Proteins can interact with DNA as TFs, which can bind to various genomic regions, or they might have DNA wrapped around them as histones, where a particular set of HPTMs can bind to specific locations. For example, HPTMs H3K4me3- and H3K9ac-targeted ChIP-seq data shows narrow peaks around TSS regions, whereas HPTMs H3K79me3 and H3K27me3 span a large proportion of gene bodies, thus generating broad peaks. Alternatively, TFs can bind in various different locations, but they always show narrow peaks (Figure 3.7). As such, peak calling should allow the user to select between narrow and broad peaks to be detected, depending on the IPed protein. Fortunately, MACS2 does allow to switch between the different peak types, thus it was implemented within the current ChIP-seq workflow with the ability to select peak calling for broad or narrow peaks.

3.4.2 Peaks from technical or biological replicates

ChIP-seq experiments often rely on using multiple replicates, which means that each replicate has its own set of peaks. Given the multiple peak sets, it raises a question of how they should be combined. In order to study this, several peak rules were tested on published GEO dataset GSE15780 mentioned in Section 3.3.4, where the rules were evaluated based on (1) resulting motif from the different peak sets, and (2) the extent of peaks overlapping particular differentially expressed genes (DEGs).

First of all, the ChIP-seq samples were targeted by TF p73 isoforms TAp73α and TAp73β, where for each isoform two biological replicates were available[42]. Second, four rules were tested for the samples:

1. Intersection — `multiIntersectBed` function within BEDtools[43] used to return regions intersecting between all samples
2. Majority — `multiIntersectBed` overlaps all samples and returns regions appearing in more than half of the samples (given 4 samples here, at least 3 should have a particular region, i.e. 75%)
   - Reduced form — *R* is used to merge nearby regions together
3. Merged BAM — BAM files merged as mentioned in Section 3.2.1, peaks were called for TAp73α and TAp73β merged BAM files and the peaks were combined
4. Union — `multiIntersectBed` used to return intersecting regions in TAp73α and TAp73β replicates separately, and returns overlapping regions between TAp73α and TAp73β
   - Reduced form — *R* is used to merge nearby regions together

One way to evaluate the peak-merging rules was by finding motifs they contain, and since p73 is part of the p53 TF family[44], it is expected to have a similar motif to p53 and p63. In order to determine the motifs from different peak sets, software suite HOMER[31] was used, where the most reliable rule for peak combination would be that with highest similarity to p53 motif, as well as the one with maximum percentage of peaks containing the p73 motif.

Another way to evaluate the rules was by overlapping them with DEGs found by comparing wild-type (WT) and p73 KO mice samples taken at different time points (days 0, 4, 7 and 14)[23]. Since genes deregulated by knocking out p73 are expected to change expression specifically because p73 is unavailable for binding to them, the overlap between genes bound by p73 and deregulated by its knock-out is expected to be strong. Therefore, the peak sets from the human genome would be annotated with nearby genes, converted into a list of homologous mouse genes and overlapped with the DEGs to find the most enriched overlap based on Fisher's Exact test[45].

3.4.2.1 Peak motif analysis

With the peaks called from ChIP-seq data reflecting locations where proteins bind to the genome, motif analysis takes those peaks and tries to find consensus motifs within them to determine the sequence a target protein binds to (this is elaborated upon in Section 3.7). Therefore, using software suite HOMER, specifically Perl script `findMotifsGenome.pl`, the peaks were searched for motifs which they are enriched for. The peak-merging rules would then be evaluated by their similarity to the known p53 and p63 motifs (Figures 3.8a and 3.8b, respectively), as well as most peaks associated with the motif and strongest enrichment (p-value) overall.

While all rules showed a highly similar motif to each other, with two CATG sequences appearing with a linker sequence of six bases between them, the merged BAM rule shows many more peaks associated with that particular motif (almost 19,000), in addition to a lower p-value compared with the other rules, indicating the strongest enrichment for it (Table 3.5).

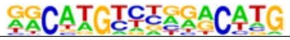
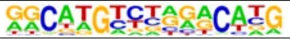
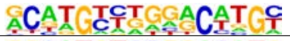
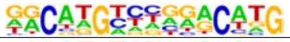
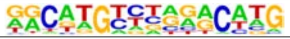
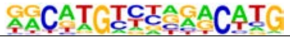
Rule	Motif	Total target sequences	% of targets	Total background sequences	% of background	P-value
Intersection		8722	69.46	40113	3.10	10^{-6847}
Majority		32018	29.49	32158	1.33	10^{-9422}
Majority (reduced)		15854	60.30	33413	3.66	10^{-9213}
BAM merged		36655	52.07	36073	4.38	10^{-15260}
Union		44017	35.62	43947	1.88	10^{-14856}
Union (reduced)		27496	53.95	26985	2.68	10^{-15239}

Table 3.5: Top motif found from the peak sets of each rule, showing the motif itself, number and percentage of target and background sequences, and p-value indicating enrichment of target motif compared with random sequences.

3.4.2.2 Overlapping peaks with DEGs

The first step to overlap the peaks from human samples with DEGs from mouse data was to annotate the peaks with nearby genes. This was done using a Perl script `regionAnnotation.pl` developed by Ramon Vidal and using various functions from BEDtools to associate peaks with nearby genes. It is important to mention theses associations between peak and their nearby genes are not limited to particular thresholds, resulting in some peaks with a nearest gene at hundreds of thousand bases apart. Since p73-bound regions are unlikely to regulate a gene thousands of bases away, it was necessary to set particular thresholds for those distances. Therefore, only peaks at a maximum distance of 5000 bases upstream or 1000 bases downstream were kept, and the number of unique transcripts were returned, to be later converted into mouse genes and associated with mouse DEGs from the p73 KO versus WT test. Due to multiple peaks being far away from their nearest genes, and due to annotations of genes and their homologous between species being somewhat lacking, only 12-30 percent of the original peaks were associated with mouse genes, which still results in a couple of thousand genes (Table 3.6).

Rule	#Peaks	#Filtered peaks	#Human transcripts	#Mouse genes
Intersection	8723	5900	4394	2723
Majority	33143	22469	7261	4170
Majority (reduced)	15861	10823	7202	4158
BAM merge	36651	24250	10239	6899
Union	44841	29703	10471	5564
Union (reduced)	27498	18018	10418	5555

TABLE 3.6: Table with number of peaks found from different rules, peaks filtered for appearing nearby human transcripts (5000 bases upstream to 1000 bases downstream), number of unique human transcripts and number of unique homologous mouse genes.

The mouse genes derived from peaks were then overlapped with DEGs compared between WT and p73 KO at different time-points (details of the differential expression (DE) analysis in Section 4.2.2), and for each overlap a p-value from Fisher's Exact test was used to establish the overlap enrichment. Interestingly, the results showed that peaks called from merged BAM files consistently had the most enriched overlap of all the rules for all different time-points for the DEGs (Table 3.7).

3.4.2.3 Peak calling on replicates — summary

As this small study on the different rules to overlapping ChIP-seq peaks has shown, merging BAM files from replicates and calling peaks for them showed the strongest motif enrichment, as well as most significant overlap with p73 KO DEGs. Therefore,

Rule	Day	#DEGs	#Peaks overlapping DEGs	P-value peaks overlapping DEGs
Intersection	DE_day0	435	40	5.24E-10
Majority			49	7.71E-09
Majority (reduced)			49	7.00E-09
Merged BAM			165	4.58E-32
Union			71	1.30E-14
Union (reduced)			71	1.19E-14
Intersection	DE_day3	439	42	1.26E-06
Majority			63	3.59E-09
Majority (reduced)			63	3.21E-09
Merged BAM			163	1.93E-30
Union			92	8.98E-16
Union (reduced)			92	8.13E-16
Intersection	DE_day7	1065	162	1.17E-33
Majority			218	1.82E-37
Majority (reduced)			218	1.18E-37
Merged BAM			410	1.24E-80
Union			297	4.35E-55
Union (reduced)			297	3.08E-55
Intersection	DE_day14	1240	187	5.71E-34
Majority			259	3.09E-40
Majority (reduced)			258	5.41E-40
Merged BAM			495	2.05E-104
Union			353	2.40E-59
Union (reduced)			353	1.61E-59

TABLE 3.7: Mouse genes based on peaks obtained from different rules overlapped with DEGs from the p73 KO test at different days. P-values were generated for each overlap between peak-related and DEGs using Exact Fisher's test, with the most significant p-values for each day are emphasized with green background.

this rule for peak calling was implemented into the ChIP-seq workflow. It should be noted, however, that the study was done with one specific dataset, and did not consider other approaches dealing with peak calling for replicates. Furthermore, while in this particular situation two biological replicates merged together produced the best peak results, merging such replicates may reduce the overall biological variability. These issues are further discussed in Section 7.2.2.

3.5 Differential occupation analysis

When dealing with ChIP-seq data originating from different conditions such as cell-type, disease and WT, time-points or tissues, a common practice is to execute a DO analysis to determine the difference between protein binding under such different conditions. This analysis involves obtaining read counts for the target samples within previously-called peak regions, and then comparing them between different conditions.

The implementation of this analysis within the ChIP-seq workflow involved an *R* code which counts the number of reads in each sample for the specific peak regions selected, filters out those which are lowly-occupied and executes the DO analysis using *DESeq2* package[20] with default parameters. For the filtration of lowly-occupied regions, the code considers width of peaks, length read and a per-base coverage supplied by the user, such that peaks with an average per-base coverage lower than a particular threshold in both conditions are filtered out. For example, if the user selects a per-base coverage of 2 and the read length is 50, peaks with an average read number lower than $\frac{peakWidth*perBaseCov}{readLength} = peakWidth/25$ in both conditions are filtered out. Read counts for the leftover peaks undergo a differential expression analysis based on the Negative Binomial distribution using the *DESeq2 R* package, which includes estimating size factors and dispersions, followed by negative binomial Gaussian linear model (GLM) fitting and Wald statistics. Finally, the Benjamini-Hochberg multiple testing correction was used to adjust the p-values. Resulting DO regions can later be used for functional enrichment analysis and motif discovery, among other tests. An example of a DO region as found between naive and context-shock samples is shown in Figure 3.9.

3.6 Functional enrichment analysis

Peaks called from ChIP-seq data do not have much biological context in themselves, which is why they are annotated with nearby genes (as discussed in Section 3.4.2.2). In cases where the resulting genes are plentiful, it is possible to give additional biological context to them by finding which functional annotations, including gene ontologies, pathways and protein-protein interactions, they are related to. Various tools have been developed to find biological terms enriched for a particular set of genes, including DAVID[46], g:Profiler[47] and GeneMania[48]. In the case of this ChIP-seq workflow, WebGestalt[32] was chosen, particularly due to its ability for displaying gene ontology (GO) category hierarchy together with highlighting those terms found to be enriched for the target gene set. In brief, WebGestalt takes in a list of ENSEMBL IDs (in the context of ChIP-seq workflow, those associated with nearby peaks), calculates a fold-change between observed and expected number of genes in every GO category, and generates tab-separated files containing GO terms enriched for an adjusted p-value equal to or less than 0.1. The downloaded tab-separated files are then accessed using an *R* script, and the enrichment scores for different terms are plotted in a heatmap with a dendogram clustering terms together based on similar enrichment scores. This allows to associate related biological terms together, as shown in Figure 6.3. It should be mentioned that this tool does not have an application programming interface (API), thus it could not be executed from a script as part of the workflow, and is rather executed by the user externally.

3.7 Motif discovery

As mentioned in Section 3.4.2.1, motif analysis finds consensus motifs within peak regions to determine the sequence a target protein binds to. Therefore, the ChIP-seq workflow has been implemented with software suite HOMER[31], specifically Perl script `findMotifsGenome.pl`, to find enriched motifs within peak regions. This script calculates an enrichment analysis over all peaks to find which motifs of a particular length (depending on the expected motif) are enriched for them compared with a background set of randomly-selected sequences. The result is an HTML file containing *de novo* motifs found for the target peaks, with particular p-values, percentage of peaks associated with that motif and the best matching known motif. Table 3.5 has already illustrated the results HOMER generates for called peaks.

By default, the algorithm searches for motifs of sizes 8, 10 and 12 given fragments of size 200, which in the case of peaks would take regions of size 200 bases around the peak center (i.e. 100 bases in each direction). Nevertheless, since peak sizes can vary greatly, parameter “size” is set to “given”, such that the exact peak sizes are used. With regards to the motif size to be returned, there is some trial-and-error involved, since motifs of various TFs vary in sizes. For example, the discovery of p73 motif in Section 6.3.2 involved using motif sizes of 8, 10, 12, 16 and 18, before it was decided that 16 was the most representative one for that protein. Therefore, motif discovery was implemented into the workflow with parameter “-size given” and a user-selected motif size.

3.8 Automation of ChIP-seq workflow

Once the design, tools and parameters were established for the ChIP-seq workflow, it was necessary to integrate and automate all steps into a functional workflow to allow convenient usage in future analyses. As such, scripts used to execute such analyses have been generated in *bash*, containing the following features:

- Argument parsing — Command-line arguments parsed using single-character parameters
- Help functions — Instructions on how to execute the script and what parameters can be used

- Tool check — Validation that tools used by scripts are installed and of the same version as been previously tested for
- Error-handling — Checks for validity of input/output files/directories chosen by the user, as well as tests throughout the scripts to validate the expected output is generated

The scripts have been divided into 3 main modules: (1) pre-processing, where the genome alignment, replicate merging, visualization and quality control take place, (2) calling peaks, and (3) detecting DHPTMs. Other methods, primarily finding motifs and functional enrichment analysis for GO terms, rely on single scripts or external tools (WebGestalt), and were therefore not implemented within those scripts. Figure 3.10 visualizes the scripts used for the ChIP-seq workflow, including all scripts, in addition to their input and output parameters.

The pre-processing scripts were used to generate BAM and bigWig from FastQ files, in addition to generating their qualities:

- *fastq2bam.sh* — Aligns reads from FastQ files (listed in parameter `inFile`) to reference genome (path to suffix of index files included in `refSuffix`) using bowtie2, filters out low-quality reads from the resulting SAM files (as mentioned in Section 3.1.2.2) and converts the filtered SAM files into BAM. In addition, the script returns numbers of mapped, uniquely-mapped reads and high-quality alignments in each sample
- *mergeBam.sh* — Takes in parameter `inDir` containing replicate BAM files to be merged (see Sections 3.2.1 and 3.4.2)
- *bam2wig.sh* — Takes in parameter `inDir` containing replicate and/or merged BAM files, and with parameter `chromSizes` containing a file with chromosome sizes of target organism, converts those files to bigWig for the visualization in a genome browser (see Section 3.2)
- *qualityControl.sh* — Given parameter `inDir` containing replicate BAM files, the script generates QC of various samples in terms of sequencing quality, enrichment, saturation and pairwise (if replicate samples are available) correlation statistics (see Section 3.3). It should be noted that optional parameter `seqType` allows the user to select the data type to determine the extent of QC (this is elaborated upon in Sections 4.1 and 4.2), `mapFile` allows the user to give a path to the mappability file, and `annotation` indicates the organism ID used to create TSS regions for *chequeR* (as mentioned in Section 3.3.3)

Scripts used to call peaks start with the high-quality replicate and merged BAM files generated in the pre-processing step:

- *fragSizes.sh* — Using parameters `inFile` containing a list of all merged BAM files and `mapFile` as a path to the mappability file, this script estimates fragment size of those files using *chequeR* (see Section 3.4) to generate a file with all fragment sizes
- *callPeaks.sh* — Using same parameter `inFile`, the script calls peaks from merged BAM files using MACS2 (see Section 3.4), with the option of using parameters `shiftFile` as the fragment sizes and `qval` as the q-value for MACS2 peak filtering (by default 0.05)
- *dhptms.sh* — Given parameter `inFile` with replicate BAM files, this script runs DO analysis to identify DHPTMs. With parameter `regions`, the user may select to search for DHPTMs within peaks, gene bodies or TSS regions. If peaks are selected, parameter `peakDir` is the directory containing all peak files. Otherwise, parameter `annotation` is used to select the target organism to extract gene information from. If TSS regions are selected, the user can also provide parameters `upstream` and `downstream` to set the ranges of those regions. Finally, parameters `pval` and `logFC` are used to set the thresholds for p-value and log fold change used by the DO analysis (see Section 3.5)

If more than one condition is available for the data, DO analysis can be executed with the script *dhptms.sh*, which searches for DO peaks by comparing reads counts from different samples within called peaks (see Section 3.5). It is also important to mention that some parameters are redundant to most scripts in Figure 3.10 and were therefore not explicitly included there, such as `outDir` to set the output directory of particular results, `cores` to select the number of cores used by the script (if possible) and `scriptDir` to allow

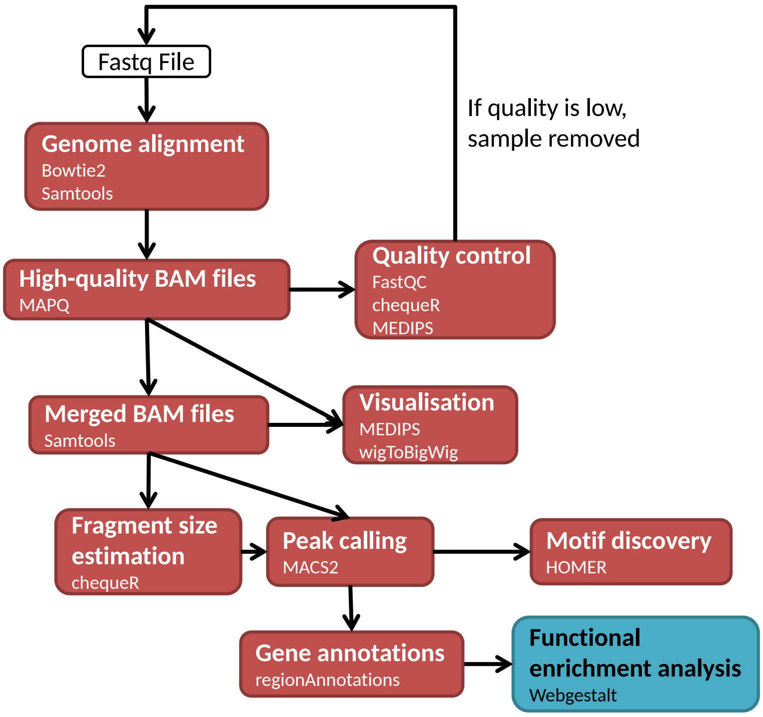
In their current form, the scripts have been tested on data generated by the paper from Halder *et. al.*[21], but in principle, most scripts are applicable to any other instance of ChIP-seq data. This workflow has currently been used in a study of chromatin modifications during learning[21], with the results discussed in Chapter 5. Furthermore, it was used in a study involving p73 regulation of multiciliogenesis[23], with the results discussed in Chapter 6. Finally, the scripts were complemented with a manual explaining how to execute them with example data from Halder *et. al.*[21], were published in a paper by Pena and Shomroni *et. al.*[11] and are available to download from Figshare[12].

3.9 Overview

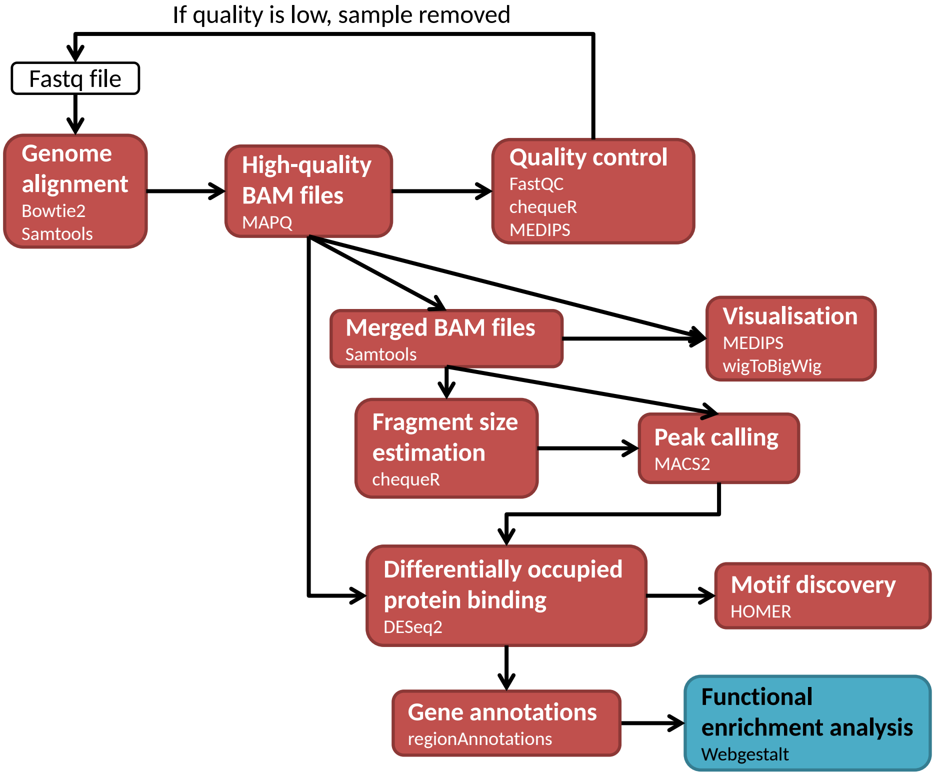
Given the abundance of ChIP-seq data used to study various interactions of proteins with DNA, it brought about the necessity to develop a flexible yet robust workflow with the ability to quickly return results regarding protein binding or DO analysis. For that purpose, a workflow was developed with various bash and *R* scripts providing the different analyses, including genome alignment, data visualization and QC, peak calling and DHPTM identification. With this workflow providing a streamlined analysis of ChIP-seq data, the results can be produced quickly and efficiently in order to integrate it with results from other data (see Chapters 5 and 6).

With the scripts already provided, they can be easily integrated into a framework built for workflow design, such as Galaxy[49] or Knime[50]. With Galaxy, the scripts could be integrated into an online workflow, which already supports various bioinformatics tools such as bowtie2 and macs2, to be used by various users worldwide for ChIP-seq analysis. Unfortunately, this can pose a problem in terms of *R* packages, as perhaps not all of those used by the workflow are supported by Galaxy, in addition to the fact that at least one package (*chequeR*) is in-house. As a workflow framework Knime can be a better option, as it functions locally and should be able to utilise whichever packages are installed in *R*. Furthermore, Knime provides a clear workflow diagram which allows the user to track the various steps of the analysis, as well as visualize the steps and exactly in which step the analysis stopped in case it crashed.

While the workflow is efficient in its analyses and design, it can always have some improvements. For example, while bash is a stable language to program, python allows stricter checks on parameters, error handling and variables set or not, whereas bash, while being capable of error handling and parameter checking, can miss such problems. Furthermore, while bowtie2 is quite beneficial in terms of aligning reads, it can be improved upon by using an expectation maximization (EM) algorithm in order to re-align multi-mapped reads, especially if they are aligned to two locations with large differences in read abundance (the algorithm should derive the likelihood of alignment to more read-abundant region). Finally, as mentioned in Section 3.4.2, merging replicate BAM files was found to be the best approach in terms of merging peak sets from replicate samples. However, the effect of this merging on biological variability has not been fully discussed, in addition to the consideration of other peak-merging rules. These issues are discussed in more detail in Section 7.2.



(A) ChIP-seq binding analysis



(B) ChIP-seq DO analysis

FIGURE 3.1: The main ChIP-seq analysis steps are alignment of FastQ reads to the reference genome, data visualization, quality control, peak calling and downstream analyses for biological context (gene annotation, functional enrichment and motif discovery). The DO analysis uses DO peaks instead of those found directly by peak calling for the biological context analyses. Red squares indicate steps executed in the command-line, while enrichment analysis (blue square) is performed using an online tool.

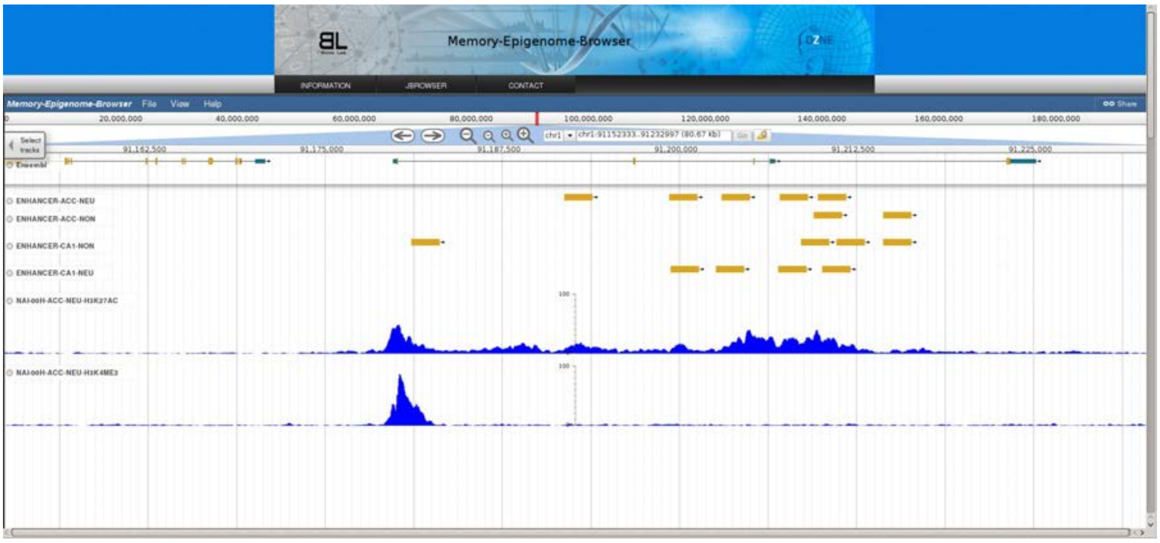
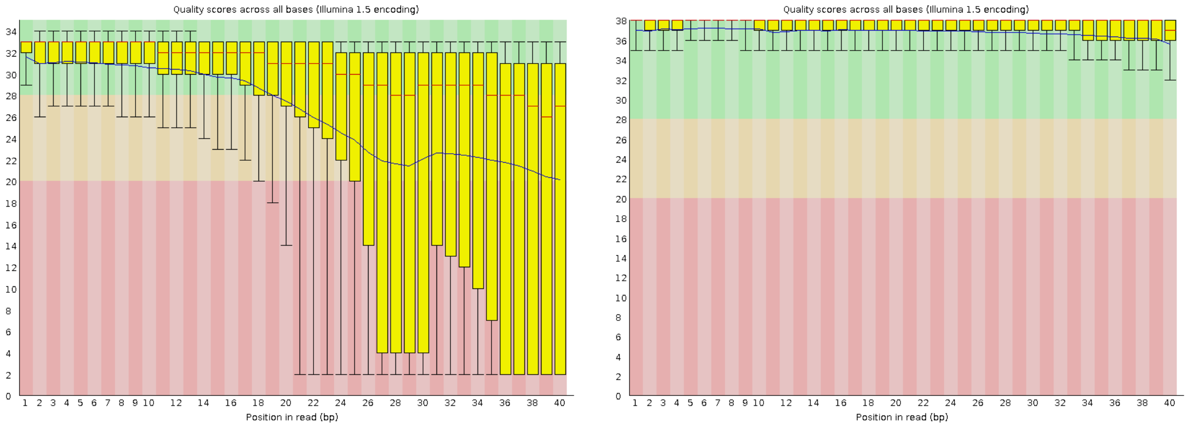


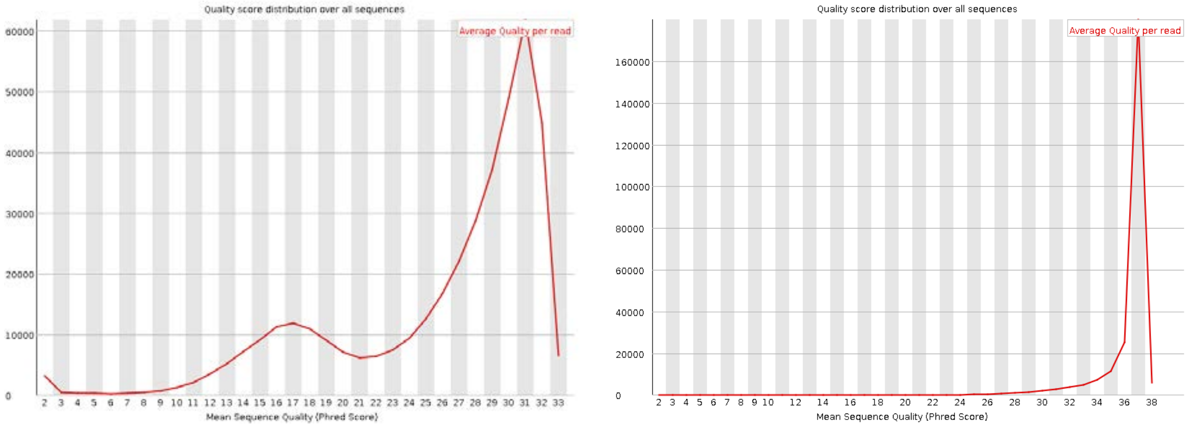
FIGURE 3.2: A custom genome browser developed by the Bonn lab in DZNE Göttingen, using data published by Halder *et. al.*[21]. The browser shows genomic coordinates at the top, a selection menu allowing to zoom in and out of the particular window, 4 tracks showing enhancer regions predicted by Halder *et. al.*[21] and two tracks of samples from the study showing enrichment of particular HPTMs.



(A) Bad per-base quality

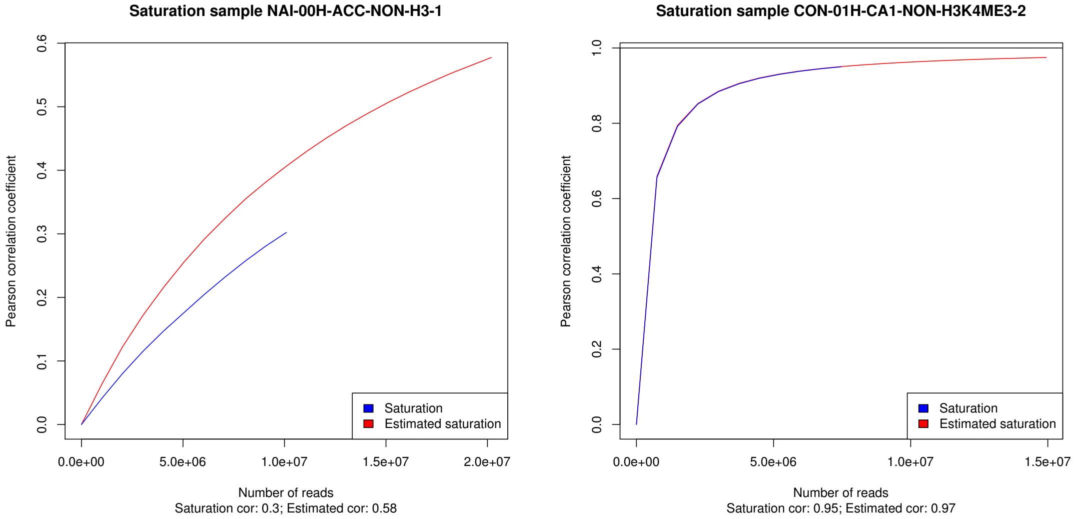
(B) Good per-base quality

FIGURE 3.3: Box-and-whisker plot generated by FastQC for per-base quality of reads length 40 bps, where each base has a particular distribution of Phred scores from very low (red background), marginal (yellow) and high (green).



(A) Bad sequence quality (B) Good sequence quality

Figure 3.4: Number of reads with a particular average sequence quality.



(A) Low saturation ChIP-seq sample (B) High saturation ChIP-seq sample

Figure 3.5: Plot showing Pearson correlation between randomly selected positive- and negative-strand reads as a function of read number randomly selected from.

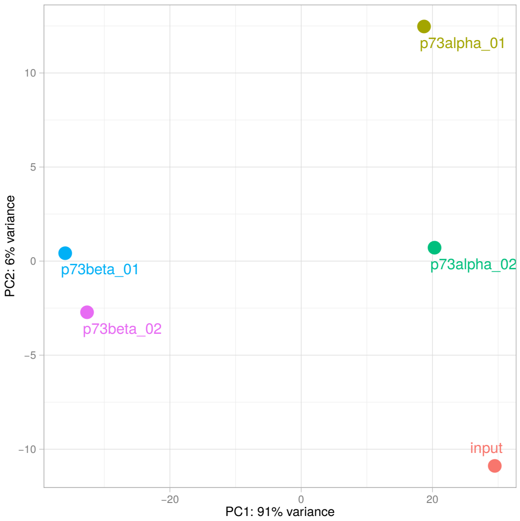


Figure 3.6: Scatter plot of ChIP-seq samples based on their first and second principle components calculated by PCA.

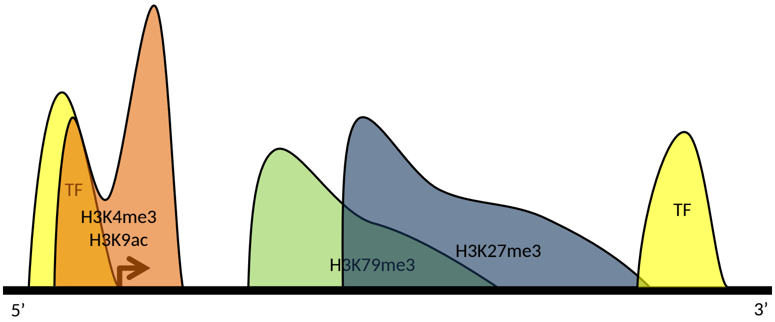
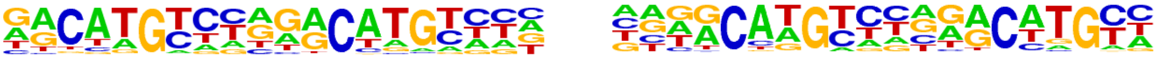


Figure 3.7: Distribution of HPTMs and TF signals appearing in different regions of the genome, where the black arrow indicates the promoter of a particular gene.



(a) p53 binding motif

(b) p63 binding motif

Figure 3.8: Binding motifs of p53 TF family.

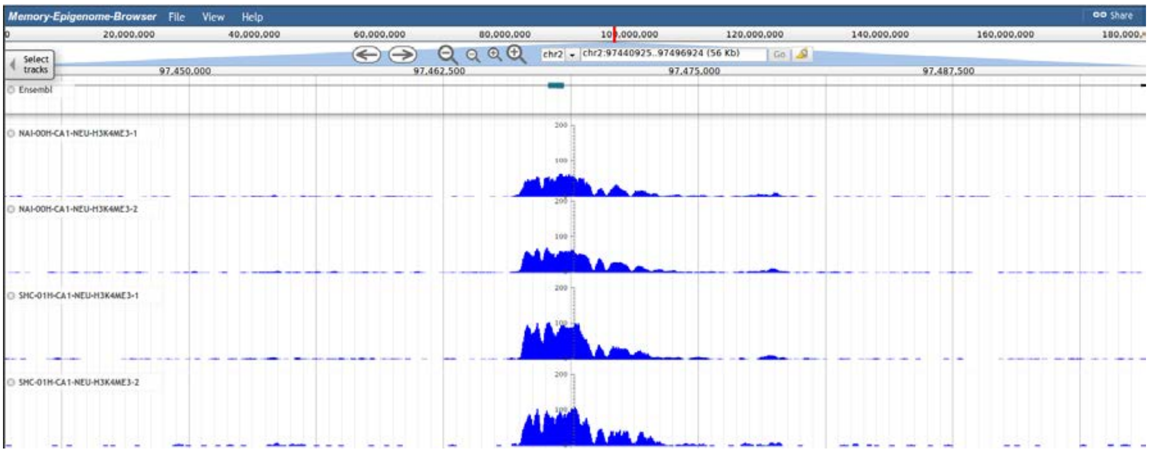


FIGURE 3.9: Genome browser visualization of a differentially occupied region between naive and context-shock samples taken from neuronal CA1 cells and targeted by H3K4me3 (https://memory-epigenome-browser.dzne.de/JBrowse-1.11.4/index.html?data=sample_data/json/mm10).

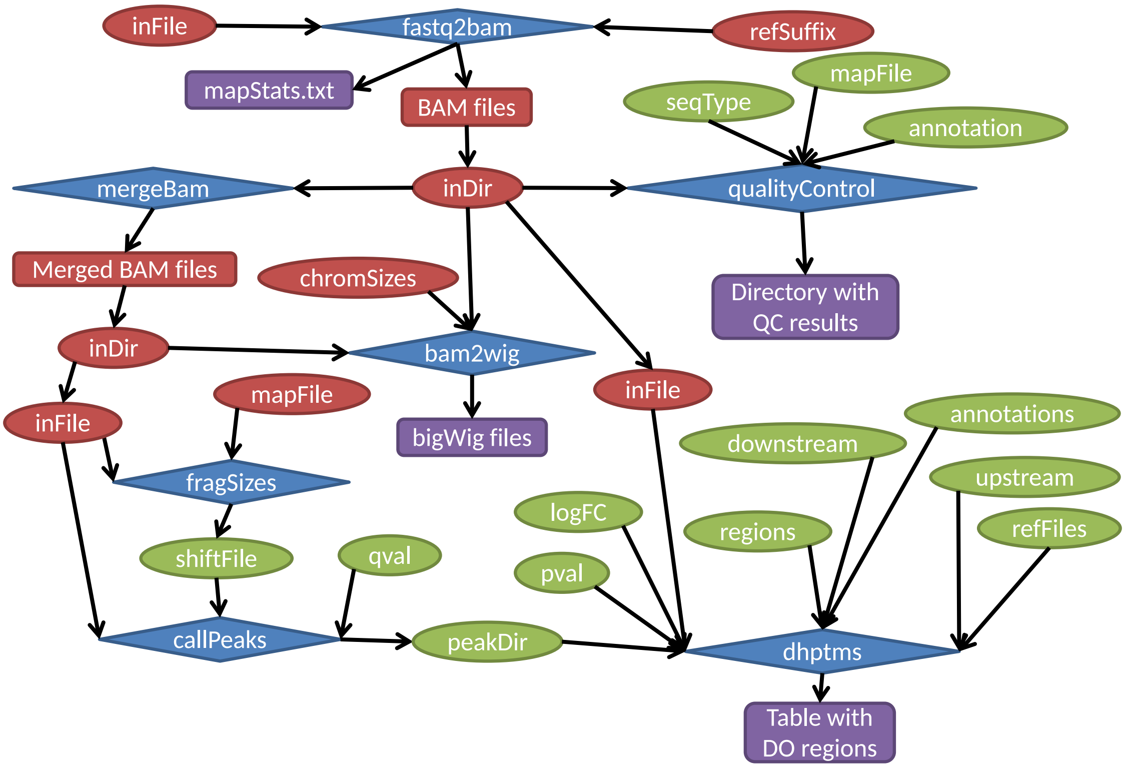


FIGURE 3.10: Overview of scripts used for the ChIP-seq analysis. Blue diamonds represent bash scripts created for main ChIP-seq data analysis, red ellipses represent required input for each script, green ellipses denote optional input and results not used for downstream analyses appear as purple rectangles.

Chapter 4

Development of MeDIP- and RNA-seq analysis workflows

In addition to the ChIP-seq workflow, we developed workflows for MedIP-seq and RNA-seq data as well. MedIP-seq is similar to ChIP-seq in the sense that they both involve IP, although MedIP-seq involves data from methylated DNA rather than proteins. In terms of analysis, both workflows were developed using fairly similar tools, with the main exception being the differential analysis (Figure 4.1).

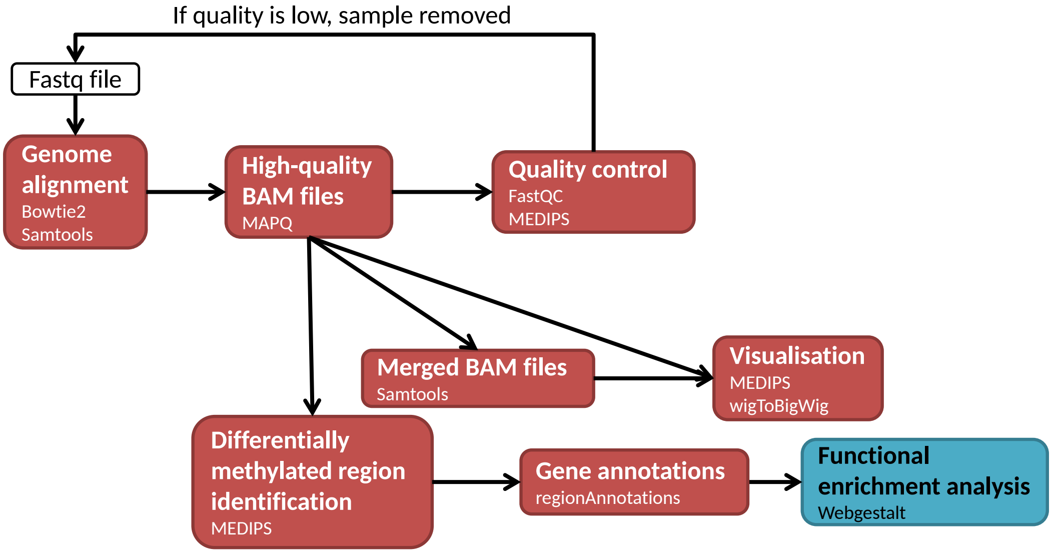


FIGURE 4.1: The main steps of the MedIP-seq analysis workflow involve alignment of the FastQ reads to the reference genome, visualizing the data, establishing its quality, executing differential methylation analysis and downstream analyses for biological context (gene annotation and functional enrichment).

On the other hand, RNA-seq targets are RNA molecules, which means the workflow differs in its approach compared with the ChIP-seq and MedIP-seq workflows. While the steps used for the analysis are similar (alignment to the reference genome, quality control, visualization and differential analysis), the specific tools used and the manner by which they are used differ (Figure 4.2). One crucial example of this is the manner of alignment: since RNA is constructed from spliced exons, aligning RNA-seq reads to the genome would require to account for splice junctions (see Section 1.1.2)

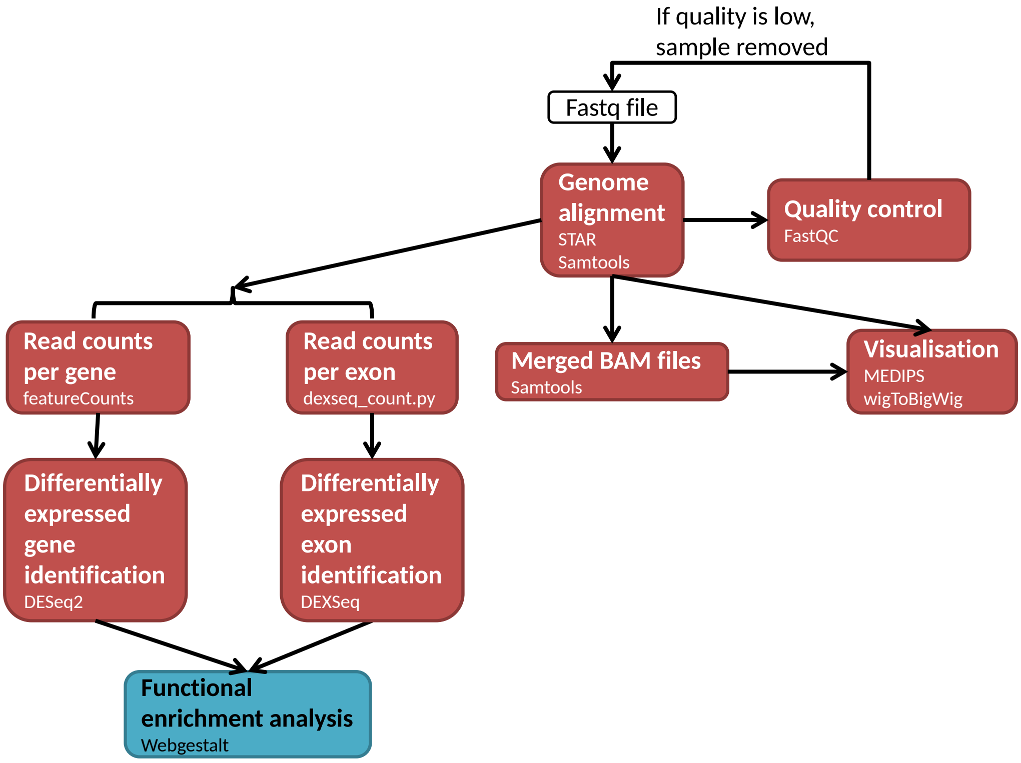


FIGURE 4.2: The main steps of the RNA-seq analysis workflow involve alignment of the FastQ reads to the reference genome, visualizing the data, establishing its quality, executing DE analyses for genes and/or exons and downstream analyses for biological context (functional enrichment).

This chapter discusses the workflows designed for MedIP-seq and RNA-seq, with emphasis on how they are comparable to the ChIP-seq workflow.

4.1 MeDIP-seq workflow

Since MedIP-seq is similar to ChIP-seq in terms of its processing, it is also similar with respect to the tools used for its execution. This means that the genome alignment is done using bowtie2 and samtools as well, with BAM files filtered to keep high-quality alignments using MAPQ scores. In terms of the quality control, FastQC is used in the same way to evaluate sequencing quality of individual reads, and *MEDIPS* package in *R* is used to return saturation correlations for individual samples and pairwise correlations in case replicates are involved. It is worth mentioning that while MedIP-seq does involve IP, it is spread out across the genome such that using a peak caller would return much of the genome as a peak, thus making the enrichment test used for ChIP-seq inapplicable. In case of visualization, the generation of bigWig files from single or merged BAM files would be the same as described for ChIP-seq in Section 3.2. One important difference between the ChIP-seq and MedIP-seq workflows is the differential analysis, since the MedIP-seq workflow is used to identify differentially methylated regions (DMRs) across genomic bins by using *edgeR* package in *R*[51]. Finally, given DMRs identified by the workflow, resulting regions can be annotated and executed through an enrichment analysis using WebGestalt.

4.1.1 Alignment for MeDIP-seq data

With the alignment tool bowtie2 and filtration to keep high-quality alignments implemented into the MedIP-seq workflow, it is important to mention that implementing those features were based on a similar rationale used for ChIP-seq (see Section 3.1.2.2). First of all, while some aligners or analyses of MedIP-seq data involve removing multi-mapped reads due to their ambiguous alignment, previous studies have validated that some DNA methylated regions contain multi-mapped reads, and should thus be included in the analysis. Furthermore, similar to HPTMs and TFs (Section 3.1.2.2), reads from methylated DNA are known to fall within intergenic regions, which are commonly repetitive, thus resulting in many intergenic multi-mapped reads.

4.1.2 Identification of differentially methylated regions

Since methylation can inhibit the expression of nearby genes, identifying DMRs between different conditions can allow to better understand changes in gene expressions. Therefore, the MedIP-seq workflow was implemented with the *MEDIPS R* package to identify DMRs through several steps. First, the genome was binned using a fixed window size and read extension to generate `MEDIPSset` objects. In order to determine the best values for those parameters, they were benchmarked as part of the study by Halder *et. al.*[21], resulting in a window size of 700 bps and read extension of 250 bps. The sequence pattern densities were next calculated for the control objects by considering CpG islands (regions high in CG bases, where methylation are often present) to allow for normalization later. Finally, the DE analysis was performed using 5% of the window size (i.e. 35 bps) as the number of reads a window must contain in order to be included in the statistical test, the *R* package *edgeR*[51] to execute the test, sequence pattern density for read count normalization and Benjamini-Hochberg p-value adjustment. The final reported table contained all windows filtered for high expression and their DE statistics, including the number of reads in each sample (raw and normalized for reads per kilobase of transcript per million (RPKM)), base mean for each condition (raw and RPKM), log fold change, raw and adjusted p-values.

4.1.3 Automation of MeDIP-seq workflow

Most scripts used to automate the MedIP-seq workflow were covered in Section 3.8, while the script *medip.sh* is an additional one used to identify DMRs. The script takes in a text file with two columns referring to (1) BAM files (with paths) used in the test and (2) their condition (control/treatment). The script has been specifically tested on data generated by the paper from Halder *et. al.*[21], and the results generated from the full study using the MedIP-seq workflow are discussed in Chapter 5. The script was also published in a paper by Pena and Shomroni *et. al.*[11] and is available to download from Figshare[12].

4.2 RNA-seq workflow

Since RNA-seq data involves sequencing RNA molecules rather than DNA, the approach to this kind of data is different, primarily in terms of the alignment and differential analysis. First, sequenced RNA is commonly mature and therefore has underwent splicing and contains exonic regions only. For that reason, the aligner utilized should be capable to handle splice junctions (regions where introns are spliced out) or be capable of aligning reads to the transcriptome instead of genome. Similarly, the approach to comparing reads between different samples should be more gene/exon oriented, rather than peak- (as used for ChIP-seq data) or CpG-island-oriented (as used for MedIP-seq data).

4.2.1 Alignment with STAR

In order to align RNA-seq data to a reference genome, STAR was implemented using a maximum of 2 alignment mismatches and a consideration of the splice junction database. In detail, STAR maps seeds to the genome by searching for reads mapping across multiple exons, i.e. with one seed aligning to a donor splice site and another mapping to an acceptor splice site[52]. While bowtie2 was used for the ChIP-seq and MedIP-seq workflows out of historical reasons, STAR was used for RNA-seq analysis as its runtime is much smaller than that of bowtie2. This is due to the fact that bowtie2 uses relatively small index files that are highly compressed and therefore take long to decompress and execute the alignment, while STAR relies on larger index files that decompress quickly and allow a faster alignment. Similarly to the alignment used for ChIP-seq and MedIP-seq data, the resulting SAM file was converted into a sorted, indexed BAM file using the Samtools suite[28].

4.2.2 Differential expression for RNA-seq data

One of the most common tests associated with RNA-seq data is the DE analysis. This test compares samples from different conditions (control and treatment) in terms of their gene expression, and based on a particular model, a set of genes is determined to be differentially expressed. Most commonly, the DE analysis is used to study cases such as WT versus KO genes, different medical treatments or diseases, and so on. In summary, the DE analysis involves testing gene expressions between control and treatment samples, and asserting which genes are DEGs for a particular statistical threshold.

Most often, studying the differential expression of genes is sufficient in order to understand how their expressions differ under different conditions. Nevertheless, since genes can have multiple isoforms with slightly different structures (as mentioned in Section 1.1), studying the genes as a whole might not be enough, and would require finding out which exons specifically determine the differences in expressions between genes under various states. Therefore, each exon in every gene is tested for its expression between control and treatment samples, and the test asserts which exons are differentially expressed exons (DEEs) for a particular statistical threshold.

4.2.2.1 Differentially expressed genes and exons

The first step of the DE analysis for genes involved using `featureCounts`[53] to count the number of reads found in each gene. This involved taking the BAM files for RNA-seq data and generating a count file with gene IDs and number of reads for each gene based on the gene-annotated reference genome. The count files for multiple samples, both control and treatment, were next imported into *R* to be combined into a single matrix. The matrix of read counts for genes in all samples was then analyzed using *DESeq2*[20] with default settings to return a list of genes with baseMean (average expression in all samples), log fold change (logFC) and p-values. Given the large number of genes tested, a multiple testing correction was necessary to adjust the p-values. However, the multiple testing correction is dependent on the number of genes in such a manner that having many insignificant genes would result in several of the truly significant genes to have their p-values adjusted over the threshold. As such, prior to the p-value adjustment, the genes were filtered such that genes with base means under 10 would be filtered out, as low expression genes would not be of interest to the analysis. Once those genes were filtered, the Benjamini-Hochberg[54], or FDR adjustment was executed for all p-values. The final reported table contained all genes filtered for high expression (more than 10 reads on average) and their DE statistics, including the baseMean, logFC, raw and adjusted p-values.

Similarly to the gene DE analysis, number of reads were counted for each exon using the `dexseq_count.py` python script as part of the *DEXSeq R* package[55] and the annotated reference genome with information about the exons in all genes. Each generated count files contained the number of reads in each exon for a particular sample, with each exon pertaining to a specific gene. The count files from multiple control and treatment samples were submitted into *DEXSeq*, and the resulting directory contained an HTML file with the information of all DEEs, as well as plots showing the expression levels of each exon within the genes containing DEEs.

4.2.3 Automation of RNA-seq workflow

Many of the scripts used for the RNA-seq workflow have already been covered in Section 3.8. One script that was particularly adapted from its original use for ChIP-seq data was *fastq2bam.sh*. This was done by setting a data type parameter ("chip", "medip" or "rna"), where selecting either of the first two generates a high-quality BAM file generated by bowtie2, and selecting "rna" return a BAM file with all reads, as generated by STAR. In addition, the detection of DEGs and DEEs from RNA-seq data involved two scripts:

- *countReads.sh* — Counts the number of reads in genes/exons (see Section 4.2.2)
- *deAnalysis.sh* — Runs DE analysis (see Section 4.2.2)

The scripts have been tested specifically on RNA-seq data generated by the paper from Halder *et. al.*[21], with the complete results of the study generated by the RNA-seq workflow are discussed in Chapter 5. This workflow has also been utilized in a study involving p73 regulation of multiciliogenesis[23], with the results discussed in Chapter 6.

4.3 Overview

This chapter has discussed the MedIP-seq and RNA-seq workflows, where some steps were covered in Chapter 3, and others that are unique to them were discussed here, primarily the DMR identification and DE analysis for MedIP-seq and RNA-seq, respectively. In addition, the benefit of having such workflows automated allows to focus more on the integration of data from multiple sources, as discussed in Chapters 5 and 6.

Chapter 5

Case Study 1: chromatin modifications regulating memory acquisition and maintenance

5.1 Study overview

Learning and memory are essential features in many organisms, allowing them adaptation to their environment. The external stimuli from the environment impact a cascading signaling network, resulting in functional and structural changes to cells responsible for establishing and maintaining memories. Memory formation and maintenance is associated with particular learning-related genes such as *Reelin*, *Calcineurin* and *Bdnf*, with those genes being regulated by particular chromatin modifications. The study conducted by Halder *et. al.*[21] was created with the intention of understanding the spatio-temporal impact of chromatin modifications on the expression of learning- and memory-related genes.

The study involved utilization of a learning paradigm known as contextual fear conditioning (CFC), whereby context shock (CS) and context (C) mice are allowed to wander in a training cage for 3 minutes, with a mild 2 second electric shock supplied to the CS mice to induce their fear of the training cage context[56]. Given this learning paradigm, mice were divided into groups and introduced to CS, C or naive (N) conditions, where the naive mice were kept in their cages. Mice under C and CS conditions were sacrificed 1 hour (1h) or 4 weeks (4w) after their conditioning to account for memory formation and maintenance, respectively, and naive mice were sacrificed directly (0 hours, or 0h). From each mice, tissue samples were extracted from the hippocampal CA1 and anterior cingulate cortex (ACC) regions, as the former is known to be involved in memory formation and the latter is known to be involved in memory maintenance[57, 58]. The RNA-seq data was generated directly from the tissue samples, with 5 replicates for various combinations of learning paradigm, time-point and brain region. On the other hand, samples intended for ChIP-seq and MedIP-seq were further processed by sorting cells into neuronal (NEU) and non-neuronal (NON) cells using a batch isolation of tissue-specific chromatin (BiTS) and chromatin immunoprecipitation (ChIP) protocol[59, 60]. The final samples for ChIP-seq and MedIP-seq involved 2 replicates for various combinations of learning paradigm, time-point, brain region and cell type. The complete design of the experiment can be seen in Figure 5.1.

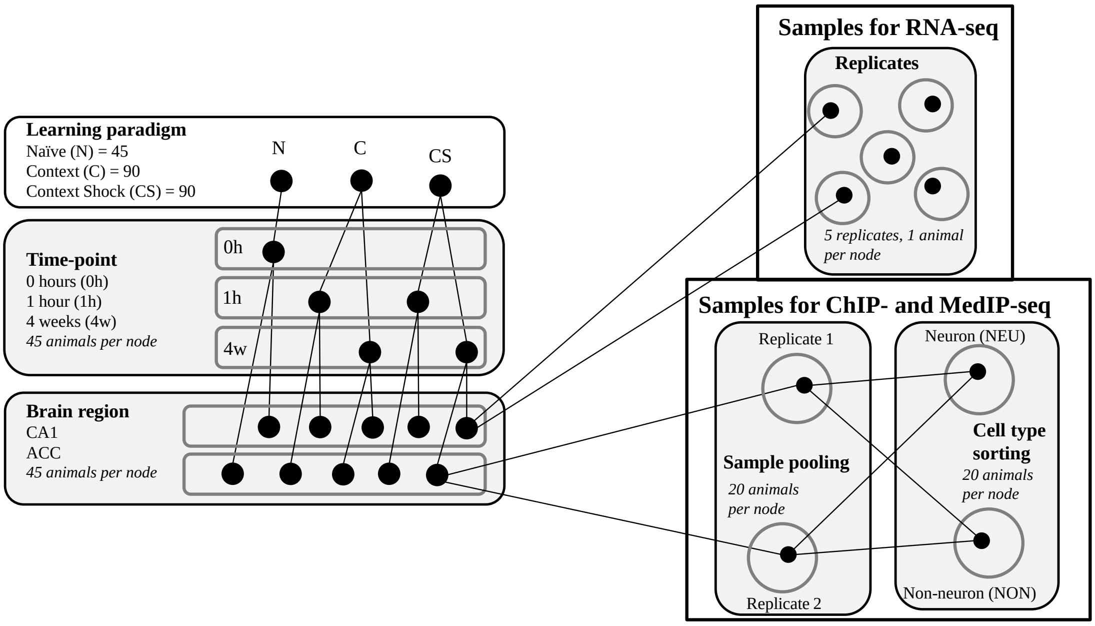


FIGURE 5.1: Design of the experiment used in Halder *et. al.*[21].

In order to confirm the quality of all samples, it was necessary to perform a quality control on each one, which involved checking sample qualities and reporting those that required re-generation. Furthermore, the ChIP-seq data was analyzed for the detection of changes in HPTMs by searching differentially enriched regions between different learning conditions for various histone marks. Both data quality control and analysis of ChIP-seq data for this particular study are discussed in this chapter.

5.2 Histone modification data

The ChIP-seq samples were generated for various combinations of the learning paradigm (N, C or CS), time-point (0h or 1h), tissue (ACC or CA1), cell-type (NEU or NON) and a particular histone modification (H3, H3K27ac, H3K27me3, H3K4me1, H3K4me3 or H3K79me3). This resulted in 108 samples, which were analyzed using the ChIP-seq workflow introduced in Chapter 3, and the results obtained from their quality control and differential enrichment analysis are mentioned here.

5.2.1 Data quality

Section 3.3 discussed the ChIP-seq sample quality can be determined through the enrichment test and saturation and pairwise correlations. Some additional quality measures included in the study were percentages of uniquely mapped and high-quality aligned reads, as well as average base coverage, to assess the quality of alignment.

The first step in analyzing the ChIP-seq samples involved using Bowtie2 to align the reads to mouse NCBI genome. As mentioned in Section 3.1.2.2, the alignment scores of reads can be used to filter out those with low quality, either by removing multi-mapped reads or by keeping both uniquely- and multi-mapped alignments with high-quality. Since there are different ways to filter alignments based on their qualities, both filtering options were attempted to determine which one would results in more reads. The output, as can be seen in Figure 5.2, indicates that filtering for high-quality results in more alignments, thus indicating the samples are of high quality in terms of overall alignment.

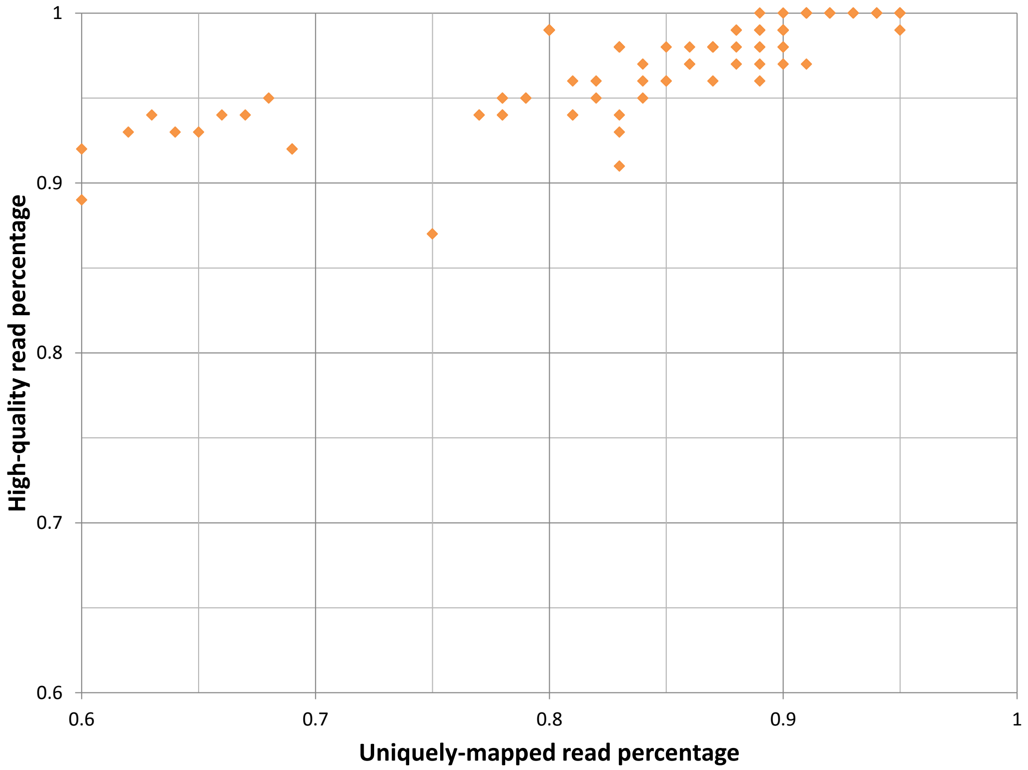


FIGURE 5.2: A scatter plot of percentages from all mapped reads in ChIP-seq samples, with uniquely-mapped reads on the x-axis and high-quality reads on the y-axis.

In addition to the alignment quality, the base coverage over the entire genome is an indicator of reads distribution, and it was calculated for each ChIP-seq sample. In order to determine whether any histone mark shows overall higher genomic coverage than others, the mean and standard deviations for the genomic coverages over the histone marks were calculated (Table 5.1). While it was expected some marks may have larger genomic coverages as they appear over broader genomic regions, the samples for different histone marks showed a fairly consistent average genomic coverage between 0.5 and 0.66, with the lowest coverage appearing for the H3 mark. Although this seems to be a low genomic coverage, it should be mentioned that ChIP-seq extracts reads from specific regions targeted by histone modifications, and as such is not expected to have a genomic coverage around 1 (i.e. spread over the entire genome on average).

Mark	Mean	Standard deviation
H3	0.502500	0.10142321
H3K27AC	0.553750	0.06238322
H3K27ME3	0.668125	0.10502182
H3K4ME1	0.618750	0.13154847
H3K4ME3	0.606875	0.14444001
H3K79ME3	0.571875	0.08084708
H3K9AC	0.597500	0.09583366

Table 5.1: The means and standard deviations of genomic coverage from all ChIP-seq samples in Halder *et. al.*[21], calculated per histone mark.

Another quality measure mentioned in Section 3.3.3 and elaborated upon in Section 2.1.3 is the enrichment of ChIP-seq samples, specifically using the *chequeR* package in order to assess the signal-to-noise ratio by calculating the NSC and RSC (see Section 2.1.3). The paper by Landt *et. al.*[15] suggested that samples with $NSC < 1.05$ and $RSC < 0.8$ be considered as low quality and undergo re-sequencing. In visual terms, Figures 2.5a, 2.5b and 2.5c show samples with low, marginal and high enrichment, respectively.

To look at the overall NSC and RSC statistics for the ChIP-seq samples in the Halder *et. al.* publication, the mean and standard deviations of the samples were taken per histone mark. Table 5.2 shows that histone mark H3 has generally lower NSC and RSC compared with other histone marks, which is known to be a background mark with low enrichment. With the exception of H3K27me3, which has a slightly lower mean RSC, all marks have mean NSC and RSC scores over the thresholds, indicating they are of high quality.

Section 3.3.4 introduced the concept of saturation, which assesses the internal reproducibility of the sample by determining whether it contains enough reads to generate a saturated coverage of the reference genome. Figure 3.5 demonstrates how low-quality samples have a difference between the saturation correlation and the estimated reference

Mark	NSC		RSC	
	Mean	Standard deviation	Mean	Standard deviation
H3	1.06875	0.01821172	0.30375	0.06946222
H3K27AC	1.295	0.05819507	1.175625	0.09172195
H3K27ME3	1.110625	0.03820449	0.75625	0.26580381
H3K4ME1	1.17375	0.03263434	0.920625	0.16299156
H3K4ME3	2.074375	0.37098012	1.1475	0.124713
H3K79ME3	1.111875	0.02833578	1	0.1808867
H3K9AC	1.28	0.06578201	1.159167	0.0727959

Table 5.2: The means and standard deviations of NSC and RSC scores from all ChIP-seq samples in the Halder *et. al.*[21], calculated per histone mark.

genome saturation, whereas the saturation profile of high-quality samples shows they are almost the same as estimated saturation.

Calculating the saturation values of ChIP-seq samples in Halder *et. al.* paper and summarizing them over histone marks shows a low saturation in H3, which is low similarly to the enrichment statistics. The second lowest saturation appears for H3K27me3, which reiterates its slightly lower enrichment compared with the other samples.

Mark	Mean	Standard deviation
H3	0.3756250	0.04912145
H3K27AC	0.8281250	0.02857009
H3K27ME3	0.6400000	0.05138093
H3K4ME1	0.6406250	0.06297817
H3K4ME3	0.9000000	0.04289522
H3K79ME3	0.8518750	0.02663801
H3K9AC	0.6458333	0.06828528

Table 5.3: The means and standard deviations of saturation correlations from all ChIP-seq samples in Halder *et. al.*[21], calculated per histone mark.

Finally, the pairwise correlation values between replicates indicate how reproducible they are, and with an average of 0.802037 ± 0.1336529 for the ChIP-seq samples in Halder *et. al.*[21], their reproducibility seems fairly high.

5.2.2 Marker-specific changes

Previous studies have supported the notion that changes in HPTMs occur shortly after learning, making them strongly associated with memory formation. Given this knowledge, DHPTMs were identified between the conditions (N versus C, N versus CS and C versus CS) using samples from CA1 neuronal and non-neuronal cells within mice sacrificed 1 hour after conditioning. While those samples reflect on the enrichment of histone marks *in vivo*, they are not highly active, thus making the distinction between

C and CS relatively minor. Therefore, published datasets GSE21161 and GSE60192 of *in vitro* samples were analyzed to compare their DHPTMs with those of *in vivo* samples. The *in vitro* samples were taken from cultured neuronal cells before (Untreated, or Un) and after KCl stimulation, and sequenced for histone modifications H3K4me1, H3K4me3, H3K27ac and H3K27me3. It is important to mention that those samples were sequenced using Applied Biosystems' sequencing by oligonucleotide ligation and detection (SOLiD) System 3.0, which uses 2 color encoding. Since Bowtie2 does not support aligning data based on this encoding, Bowtie was used with color space mapping instead. The *in vitro* samples were therefore downloaded from the GEO website, converted into `csfasta` format using `fastq-dump` and aligned to the mouse NCBI genome using Bowtie with color space mapping and allowing for 1 mismatch in end-to-end alignment.

Peak calling has been discussed in Section 3.4 as a common practice to find enriched regions within ChIP-seq data, specifically using MACS2 for the current workflow. As some histone modifications are known to appear in specific regions, such as promoter regions or gene bodies, both *in vivo* and *in vitro* samples were analyzed for DHPTMs based on their histone marks, where H3K4me3 and H3K9ac samples were analyzed for TSS regions of 500 bases upstream and 1000 bases downstream of the TSS, H3K27me3 and H3K79me3 were analyzed for gene body regions, and H3K27ac and H3K4me1 were analyzed for peaks.

Since HPTMs are known to regulate the expression of genes, it is fair to assume that HPTM changes would in turn regulate gene expressions. For this reason, RNA-seq samples, specifically CA1 1h, were analyzed for DEGs between conditions N-C and N-CS, as discussed in Section 4.2.2. The resulting 915 DEGs for N-CS and 1212 DEGs for N-C were overlapped with DHPTMs in the equivalent tests in order to establish the relationship between them.

The differential enrichment analysis for specific regions in different samples was executed by comparing the number of reads in each of the regions and selecting DHPTMs as those with adjusted p-value smaller than 0.1. When comparing Un-KCl and learning induced N-CS analyses, both marks H3K27ac and H3K27me3 showed many more DHPTMs in the Un-KCl analyses, whereas H3K4me1 and H3K4me3 showed a few more DHPTMs in the learning-induced N-CS analyses (see Figure 5.3). Overall, the N-CS analyses showed few DHPTMs, with H3K79me3 being the only exception with 850 DHPTMs, and all other marks showing 86 DHPTMs or less.

Finally, overlap significance was calculated between DHPTMs and DEGs, using those enriched for N-CS comparison in H3K79me3 and Un-KCl in H3K27ac, using Fisher's exact test and a p-value of 0.05 as a threshold fo significant overlap. The result showed that both DHPTMs form N-CS and Un-KCl tests were significantly overlapping with

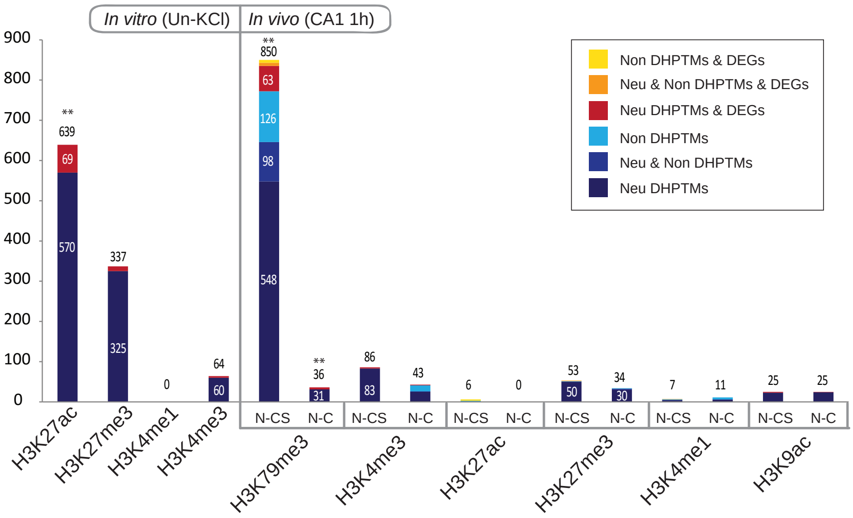


FIGURE 5.3: Bar plots with numbers of DHPTM regions for different HPTMs being tested for untreated vs. KCl treatment (Un-KCl) in *in vitro* samples and naive vs. context (N-C) and naive vs. context-shock (N-CS) in *in vivo* CA1 1h NEU and NON samples. The DHPTMs have also been overlapped with DEGs for N-C and N-CS comparisons using RNA-seq *in vivo* data. Figure corresponds to Supplementary Fig. 22a in Halder *et. al.*[21].

DEGs, thus establishing a non-random associations of those DHPTMs with the DEGs (Figure 5.4).

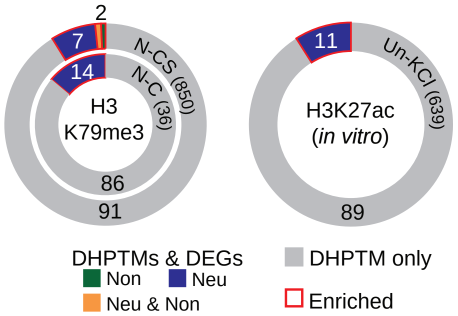


FIGURE 5.4: Pie charts displaying number of DEGs for N-C or N-CS analyses in RNA-seq *in vivo* data overlapping DHPTMs for H3K79me3 *in vivo* data or H3K27ac *in vitro* data. Overlaps enriched for Fisher's exact test ($p - value \leq 0.05$) are marked with a red outline. This figure corresponds to Figure 3c in Halder *et. al.*[21].

5.3 Methylated DNA data

As mentioned before, the MedIP-seq samples were generated similarly to those sequenced with ChIP-seq, except the methylated samples were sequenced for methylated regions rather than histone modifications (MC). The resulting 40 samples were analyzed using the MedIP-seq workflow described in Chapter 4, and their qualities are mentioned here.

5.3.1 Data quality

Similarly to the ChIP-seq samples, reads of MedIP-seq data in the current study were aligned with Bowtie2 to the mouse NCBI genome. The quality control steps introduced in Section 3.3 were similarly used for the MedIP-seq samples, with the exception of RSC and NSC statistics, as those are ChIP-seq specific.

Similarly to the ChIP-seq samples, the percentage of high-quality and uniquely-mapped alignments within all mapped reads were compared for the MedIP-seq samples. Similarly to the ChIP-seq samples, Figure 5.5 shows for all MedIP-seq samples have a higher percentage of high-quality alignments out of all mapped reads than the equivalent uniquely-mapped reads, which again shows an overall good quality of alignment for all samples. The alignment was also evaluated by overall genome coverage, with MedIP-seq samples having an average of 0.538 ± 0.0590424, showing a relatively low coverage. Nevertheless, similar to ChIP-seq, MedIP-seq reads are often found in specific regions, predominantly CpG islands, thus it is unlikely they would have a high genomic coverage.

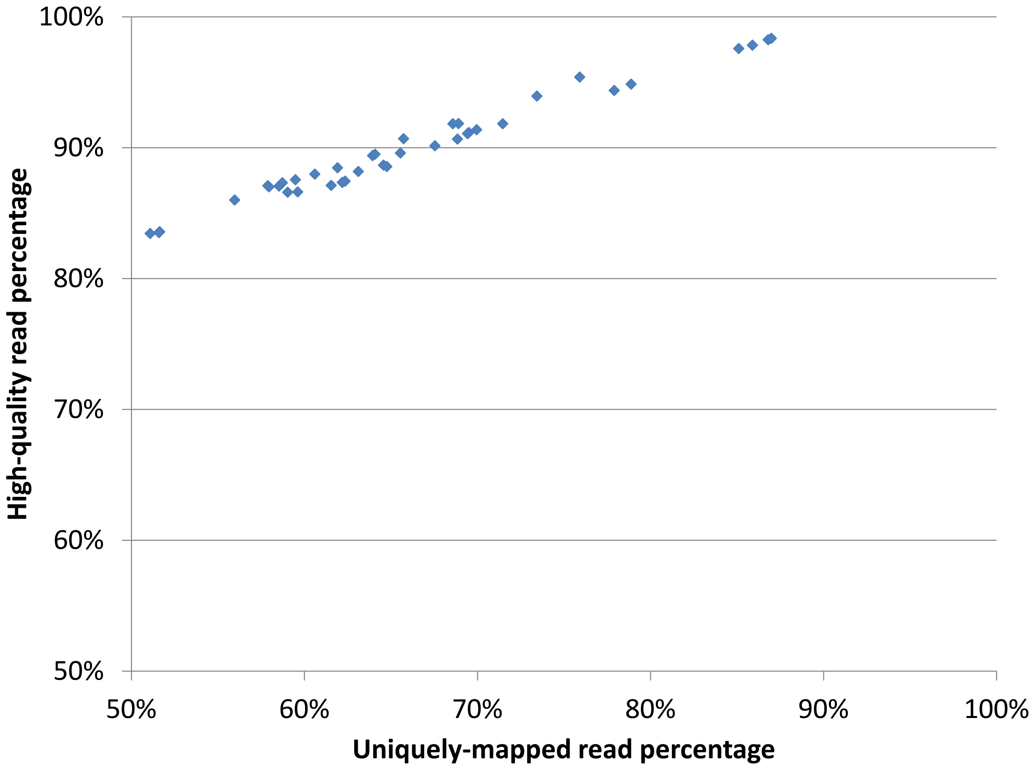


FIGURE 5.5: A scatter plot of percentages from all mapped reads in MedIP-seq samples, with uniquely-mapped reads on the x-axis and high-quality reads on the y-axis.

The MedIP-seq samples were also tested for saturation correlation, showing an average value of 0.7825 ± 0.05028668, which is relatively high. In visual terms, Figure 5.6 shows that even the MedIP-seq sample with lowest saturation has the estimated and sample saturations fairly close, reinforcing the good saturation levels of all MedIP-seq samples used in the study.

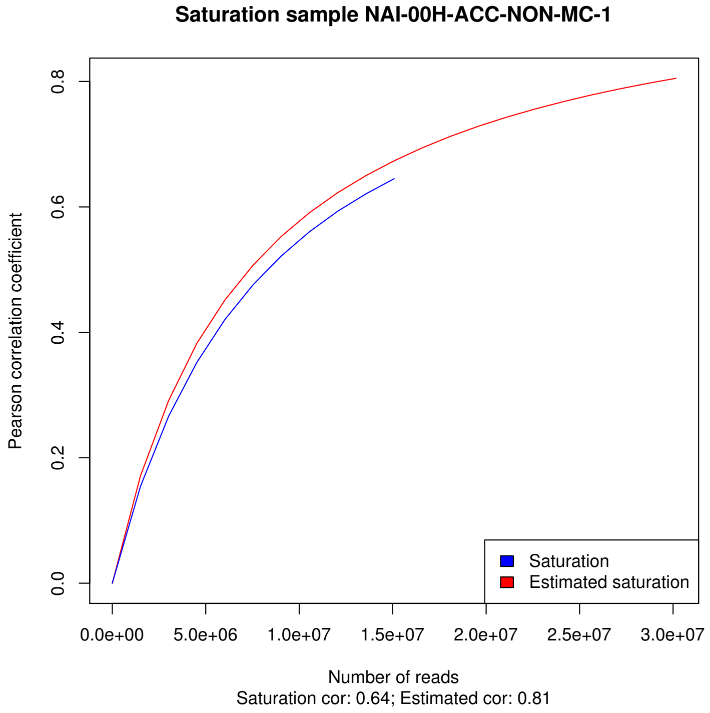


FIGURE 5.6: Saturation correlation for MedIP-seq sample with lowest saturation.

Finally, the reproducibility between replicates for the MedIP-seq samples showed an average pairwise correlation of 0.883±0.036207734, thus suggesting a high reproducibility level.

5.4 Overview

The study of HPTMs deregulated during learning resulted in high-quality ChIP-seq data, but relatively few DHPTMs considering the lenient thresholds (adjusted p-value = 0.1 and no log fold-change threshold). While gene body mark H3K79me3 showed 850 DHPTMs, all other markers had 86 DHPTMs at most, which is relatively small compared with the 915 DEGs found for the RNA-seq data in CA1 1h. An equivalent execution of DO analysis on the published data GSE60192 resulted in low numbers of DHPTMs for marks H3K4me3, H3K27me3 and H3K4me1 (at most 337 for H3K27me3), with the exception of 1106 DHPTMs for H3K27ac. With those low numbers of DHPTMs, little to no overlap was found between them and the DEGs, although DHPTMs from the DO analyses of *in vivo* H3K79me3 samples and *in vitro* H3K27ac ones showed a strong association with the DEGs. In addition, when the strong overlaps were tested with functional enrichment test, many terms were found to be strongly associated with synapses (data not shown).

The final results, as we published in the paper by Halder *et. al.*[21], showed some global HPTMs but mostly few specific ones. An additional result was the difference between DEGs and most specific DHPTMs, which was unexpected due to their common association in terms of cellular differentiation and development[59, 61, 62]. One possible reason for the weak association could be attributed to lack of sensitivity in the analysis. However, since neuronal signaling during learning are rapid and pulse-like, they might regulate another mechanism to cause quick changes in gene expressions, rather than regulating HPTMs which are comparably more permanent regulators of genes in terms of cell differentiation and maintenance.

Chapter 6

Case Study 2: regulation of cilia development by p73

Cilia are microscopic surface organelles growing on various cells and associate with a multitude of functions critical for organisms, including clearing airways from inhaled pollutants and pathogens, helping the flow of cerebral spinal fluid in the brain and providing movement for gamete cells (sperm and eggs). Through multiciliogenesis, cilia develop in large quantities to perform their functions, and when they do not develop properly or at all, a plethora of diseases may occur. For example, the lack of cilia in tracheal epithelial (lung) cells may cause respiratory tract infections[63], brain cells without cilia may lead to hydrocephalus[64], and gamete cells lacking cilia makes them immobile, leading to infertility[63]. Genetically speaking, the development of cilia involves various proteins and TFs, primarily Multicilin and E2F4/5 complex activating specific genes upstream to regulate the function of *Myb*, *FoxJ1* and *Rfx2-4* genes[65] (Figure 6.1).

Another TF called p73, belonging to the p53 family and known to be a tumor suppressor, was studied in p73 KO mice. Interestingly, p73 KO mice exhibited brain defects, infertility and rhinitis (inflammation of the mucus membrane in the nose)[66], which are similar phenotypes exhibited by organisms with defects in multiciliogenesis. Therefore, a hypothesis was established whereby p73 is a regulator of multiciliogenesis. To test this theory, a study was conducted to understand how p73 affects multiciliogenesis, specifically in the context of a regulatory network of genes responsible for it. This chapter focuses on the results generated for the paper by Nemajerova *et. al.*[23] to unravel the function of TP73 within the multicilogenesis regulatory network.

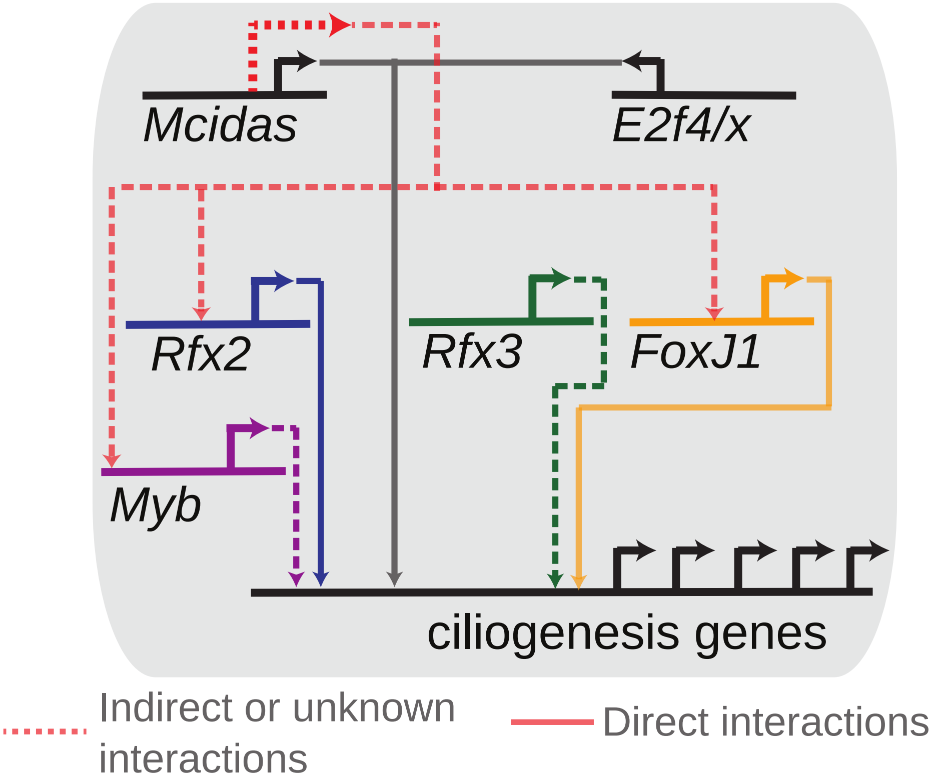


Figure 6.1: gene regulatory network (GRN) of known interactions between TFs contributing to multiciliogenesis. Solid lines show direct binding of TFs to target genes, while dotted edges indicate indirect or unknown interactions.

6.1 Experiment design

In order to confirm the effect of p73 on the development of cilia at different time-points, the study was conducted in murine tracheal epithelial cells (MTEC) (mouse lung cells), with WT and TP73KO samples taken at different time points to reflect on the stages of cilia development (days 0, 4, 7 and 14). For each time-point and condition combination, 3 replicates were taken, and the samples were sequenced using both RNA-seq and small RNA sequencing (sRNA-seq), in order to study both genes themselves and small RNAs (sRNAs) that may regulate their expression.

While the WT and TP73KO samples could be used to find how knocked-out p73 affects multiciliogenesis, it could not be used to determine which multiciliogenesis-related genes are regulated by p73. Therefore, in addition to the data above, the study utilized previously published GEO dataset GSE15780, which contains ChIP-seq samples from human osteosarcoma (bone cancer) Saos2 cells targeting protein isoforms TAp73α and TAp73β[42]. While those cells are not related to ciliogenesis, the assumption was that TAp73 binding is conserved enough that regardless of the organism or cell-type, at least several of the genes bound by TAp73 will be ciliogenesis-related. Additionally, published GEO dataset GSE22142, containing microarray data for human airway epithelium cells at days 0, 7, 14 and 21[67], was used. While the human and mouse samples corresponded to different days, the data correspond to the equivalent stages in development of cilia on mouse and human lung cells, thus making them useful to study comparable WT expression of genes relevant to multiciliogenesis.

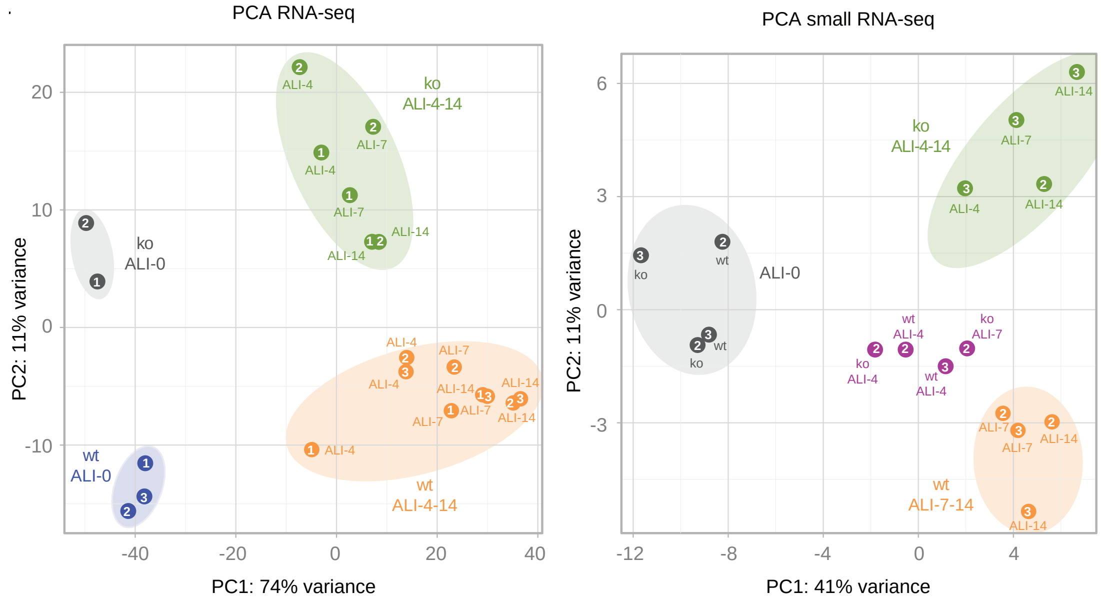
6.2 Differential gene expression during ciliogenesis

To determine the mechanism by which p73 regulates multiciliogenesis, it was necessary to determine which specific genes are affected by knocking out p73, and which of them are also related to multiciliogenesis. The initial analysis, relying on the RNA- and sRNA-seq data alone, focused on the DEGs and differentially expressed sRNAs between WT and p73KO samples across different time-points, in order to determine whether there is a large difference in the DEGs and differentially expressed sRNAs between different multiciliogenesis stages. The RNA-seq was primarily analysed using alignment with STAR (Section 4.2.1), BAM file merging (Section 3.2.1), visualization (Section 3.2), quality control (Section 3.3) and DE analysis (Section 4.2.2). Alternatively, the sRNA-seq data was analysed using Oasis[68], particularly to generate QC for the samples, detect the number of reads associated with each sRNA and identify differentially expressed sRNAs.

6.2.1 Data quality

For the RNA-seq samples, part of the quality control involved alignment rates, particularly the proportion of reads mapped uniquely or unmapped. After alignment of the RNA-seq reads to the mouse NCBI transcriptome using STAR, on average 84.10% of the reads were uniquely aligned, with a standard deviation of 1.72, indicating an overall high rate of unique read alignment. To further reinforce the notion of high-quality alignments, the rate of unmapped reads showed an average of 2.12% with a standard deviation of 1.24, indicating a low rate of unmapped reads and therefore high alignment quality. Alternatively, the sRNA-seq data was aligned to the mouse NCBI transcriptome using STAR with different parameters than for the RNA-seq (for example, STAR accounts for smaller read size when using sRNA-seq). Alignment rates showed to be a bit lower than for RNA-seq samples, with 65.93% average uniquely-mapping rate and standard deviation of 5.39, and 13.72% average unmapped rate and standard deviation of 3.29. While the mapping rate for the sRNA-seq data is lower than for the RNA-seq samples, previous studies have shown that sRNA-seq data often have lower mapping rates compared to other next-generation sequencing techniques, thus showing mapping rates that are considerably high for sRNA-seq data.

PCA plots were used to determine the manner by which samples were grouped according to their condition and time-point. An initial plot generated for all RNA-seq samples using the PCA code from the first module of Oasis[68] showed a separation to WT at day 0, p73KO at day 0, WT at days 4-14 and p73KO at days 4-14. Nevertheless, p73KO replicate 3 samples grouped somewhat closer to WT rather than p73KO, so they were removed to make the groups more distinct, resulting in more DEGs (Figure 6.2a). Alternatively, using the first module of Oasis to execute the sRNA-seq samples resulted in no clear separation into groups, mostly due to replicate 1 samples. Removing those samples resulted in 3 clear groups for WT at day 0, WT at days 7-14 and p73KO at days 4-14 (Figure 6.2b).



(A) RNA-seq samples (B) sRNA-seq samples

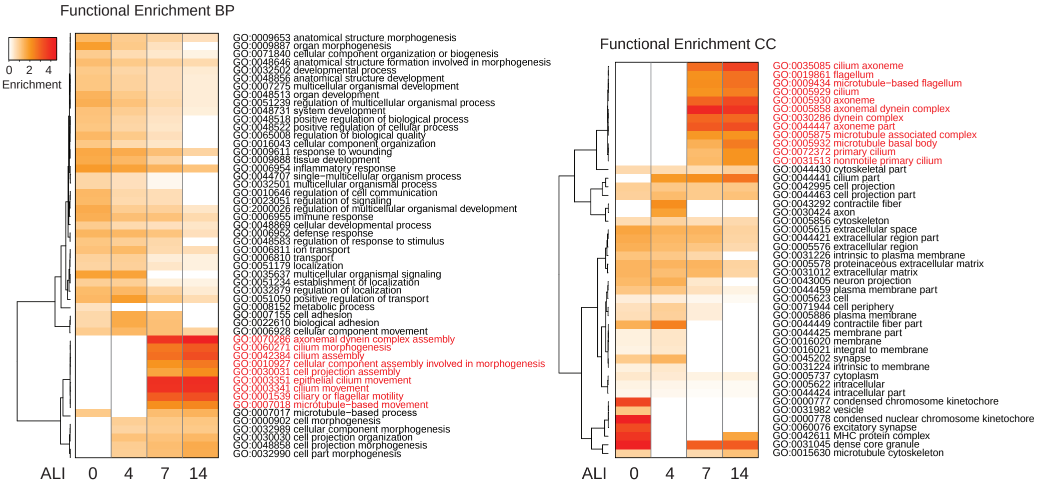
Figure 6.2: PCA plots for WT and p73KO samples, grouped by condition and time-point. For RNA-seq data, replicate 3 p73KO samples were removed to improve on sample grouping. For sRNA-seq data, replicate 1 samples (both WT and p73KO) were removed to improve on sample grouping. Figure corresponds to Supplementary Fig. 4a-b in Nemajerova *et. al.*[23]

6.2.2 Ciliogenesis deregulation

Using the DE analysis as mentioned in Section 4.2.2, the WT and p73KO RNA-seq samples were compared at different time-points, while excluding replicate 3 p73KO samples]. Overall, 1930 DEGs were found at different time-points based on thresholds of adjusted p-value $padj <= 0.1$ and log2 fold-change $log2FC <= 0.6$. Amongst the DEGs at different time-points, genes of TFs known to be associated with ciliogenesis regulation, primarily *Rfx2*, *Rfx3* and *FoxJ1*, were found to be deregulated at later time-points (day 7 for Rfx genes and day 14 for ForJ1), while *Trp73* was shown to be deregulated at day

0. This suggests that *Trp73* is a precursor of *Rfx2*, *Rfx3* and *FoxJ1* deregulation, and may be responsible for it. Alternatively, the second module of Oasis was used to execute DE analysis on the sRNA-seq data for different time-points while excluding replicate 1 samples. This resulted in 45 differentially expressed sRNAs at different time-points, including miR34b/c, which showed deregulation at day 14, and therefore involvement at later ciliogenesis stage.

In order to determine the biological context of said DEGs, WebGestalt[32] was used to execute enrichment analysis by supplying the list of DEGs and returning enriched biological process (BP) and cellular component (CC) GO categories. Specifically, the top 20 most significant terms (lowest p-values adjusted with Benjamini-Hochberg correction) were returned in each GO group for each time-point, and their enrichment scores were plotted in a heatmap (Figure 6.3a for BP categories and Figure 6.3b for CC categories). This particular analysis has shown that several cilia-related terms are exclusively highly-enriched for DEGs in later time-points (days 7 and 14), which corresponds to the critical stages in cilia development.



(A) Heatmap for enriched BP categories (B) Heatmap for enriched CC categories

FIGURE 6.3: Heatmaps with enrichment values of top 20 GO categories according to DEGs at different time-points, with cilia-related terms marked in red. Figure corresponds to Supplementary Fig. 5a-b in Nemajerova *et. al.*[23]

6.2.3 Differentially expressed genes and ciliogenic gene association

With cilia-related terms reflecting the general association of WT-p73KO DEGs to ciliogenesis (Figure 6.3), the next step was to evaluate the specific association between DEGs and cilia-related genes in order to provide additional proof for it. This was done by preparing a list of genes associated with ciliary biogenesis, maintenance and function, relying on 3 publicly available resources:

1. Genes associated with GO terms containing the pattern "cili", thus relating to "cilium", "cilio", "cilia" and so forth. The genes were extracted using the *R* packages *GO.db_3.0.0* and *org.Mm.eg.db_3.0.0* (mouse gene annotations)
2. SysCilia Gold Standard Version 1[69]
3. Manual annotation of particular ciliogenesis-related genes, particularly *Ccdc65*, *Rfx2* and *Spag1*

From the sources, 620 genes were determined to be cilia-related, and those were overlapped with DEGs at different time-points. The resulting percentages of cilia-related genes out of all DEGs was relatively low (Table 6.1). Nevertheless, a Fisher's exact test was executed to determine the significance of overlap between DEGs and cilia-related genes (CRGs). At days 0 and 4, the overlap was found to be non-significant, whereas days 7 and 14 showed an extremely significant overlap (Table 6.1). Similarly to the enrichment analysis executed with WebGestalt, the association of DEGs to CRGs was found to be strong specifically at later stages of cilia development.

Day	Mouse genes	DEGs	# Overlap DEGs CRGs	% Overlap DEGs CRGs	Overlap significance (p-value)
0	38355	435	8	1.84	0.4061
4		437	11	2.52	0.1002
7		1064	82	7.71	$< 2.2 * 10^{-16}$
14		1236	147	11.89	$< 2.2 * 10^{-16}$

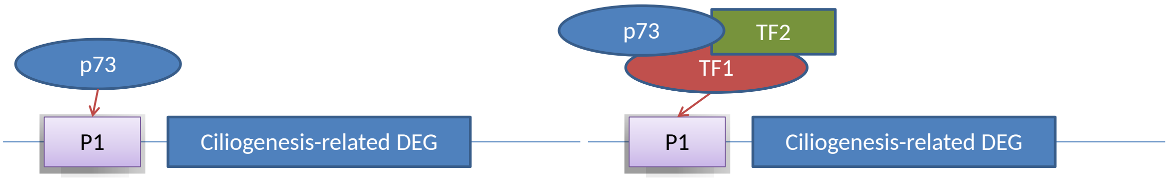
Table 6.1: DEGs for RNA-seq data between WT and p73KO at different time-points, considering how many are up- or downregulated, and how many are cilium-related.

6.3 Genes bound by p73

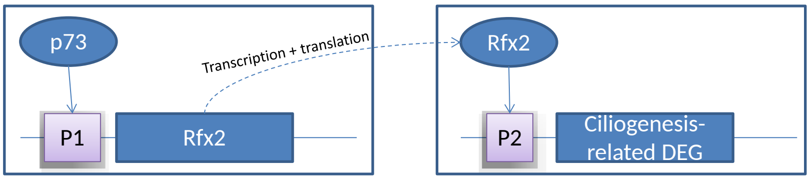
With the knowledge that knocking out p73 deregulates many multiciliogenesis-related genes at different stages of cilia development, it was necessary to determine how many of those DEGs are directly bound by the TF *TAp73* in order to establish its position within the multiciliogenesis regulatory network. To do this, a general set of genes targeted by p73, in addition to its binding site, had to be determined. For this purpose, published human ChIP-seq data was used to find regions enriched for TAp73, determine its binding site from those regions and associate both with specific genes.

Regulation of CRGs by TFs such as p73 can occur in multiple ways: directly binding to an enhancer region to regulate the CRG expression (Figure 6.4a), binding to such a region as part of a protein complex (Figure 6.4b), or by indirect binding, where a TF directly regulating the CRG expression, such as Rfx2, is regulated by p73 (Figure 6.4c).

The ChIP-seq data was analysed using the workflow discussed in Chapter 3, particularly genome alignment with Bowtie2 and high-quality filtering (Section 3.1), BAM



(A) Direct binding (B) Binding as part of protein complex



(C) Indirect binding

Figure 6.4: Models demonstrating how p73 can interact with CRGs.

merging (Sections 3.2.1 and 3.4.2), data visualization (Section 3.2), quality control (Section 3.3) and peak calling (Section 3.4). Since the data consisted of sample replicates corresponding to different p73 isoforms (TAp73 α and β), the BAM files were merged for the isoforms separately, and the peaks called from each merged file were combined by overlapping all regions together.

6.3.1 Data quality

As mentioned in Section 3.3, quality control of ChIP-seq samples involves four main aspects: alignment quality, enrichment test, saturation and pairwise correlations, and all those quality measures were recorded for the ChIP-seq TAp73 samples (Table 6.2). The alignment quality of all samples appeared to be favorable towards high-quality alignments (around 99% of all mapped reads), with uniquely-mapped reads constituting less than 85% of mapped reads, suggesting the presence of many high-quality multimapped alignments. In terms of sequencing depth, the average base coverage is between 0.27 and 0.3, which is to be expected given that TFs bind to specific targets in the genome, thus are not expected to cover a large proportion of the entire genome. In terms of enrichment, both NSC and RSC scores are high enough for the p73 samples to consider them as highly enriched, whereas the input sample has a low RSC, which is expected from an input sample without protein targeting. In terms of saturation, the samples show a score above 0.6, which is considered to be sufficiently high to declare the samples as saturated. Finally, the replicate correlations are 0.73 and 0.89 for the α and β isoform samples, respectively, which while indicating a stronger correlation between β than α samples, still indicates a sufficient association to warrant them as replicates.

Sample Name	% High-Quality Alignments	% Uniquely-Mapped Alignments	Average Base Coverage	NSC	RSC	Saturation Correlation	Replicate Correlation
input	98.99	83.14	0.303836867	1.19	0.48	0.62	NA
p73alpha_01	99.32	84.35	0.279864414	2.7	1.89	0.78	0.73
p73alpha_02	98.98	83.95	0.281967733	1.66	1.08	0.64	0.73
p73beta_01	99.34	84.14	0.280341263	5.31	2.22	0.86	0.89
p73beta_02	99.1	84.75	0.273284246	4.49	1.84	0.84	0.89

Table 6.2: Quality of all TAp73 samples in terms of alignment (percentages of high-quality and uniquely-mapped alignments, and average base coverage), sample enrichment, saturation and pairwise (replicate) correlation.

6.3.2 p73 motif discovery

In order to establish which genes are specifically regulated by the TF TAp73, it was first necessary to determine its binding site in the form of a motif (see Section 3.7). To do this, BAM files for replicate samples of isoforms TAp73α and TAp73β were merged separately, and peak calling was executed on each (Section 3.4). The called peaks were then combined by overlapping all regions together, resulting in a set of peaks corresponding to both TAp73α and TAp73β. The resulting peaks were used for *de novo* discovery of enriched motifs using HOMER[31], specifically using Perl script *findMotifsGenome.pl* with parameters "–len 16" and "–size given" to find motifs of size 16 base pairs (bps) over entire peak regions. Results showed that highest enriched *de novo* motifs for peak sets from TAp73α, TAp73β and the union of both are similar to p53 and p63. Since p73 is known to be of the same family as p53/p63, the top enriched motifs for TAp73 peaks indeed represent p73 binding site. Those motifs showed extremely high-enrichment, with p-values $< 10^{-8832}$ and more than 50% of the peaks in each set associated with top motif, indicating a strong motif presence in the TAp73 ChIP-seq data.

By defining the top enriched motif as that of p73, the motif was used to execute *findMotifsGenome.pl* on the merged peak set, resulting in a list of all peaks with p73 motif, in addition to the motif score and distance for each peak. The motif distances were calculated between each motif and the peak center it was found in, and those were plotted to illustrate the pattern of motif distances from peak centers (Figure 6.5). With the largest frequency of motifs at a distance close to 0, it was concluded that most p73 binding sites were found around the center of TAp73 peaks, which corroborates between the peaks enriched for TAp73 and regions targeted by it.

6.3.3 Annotation of p73 peaks

With general and motif-containing peaks associated with TAp73, all of them had to be annotated for later association with gene expressions of p73KO DEGs. For this, an

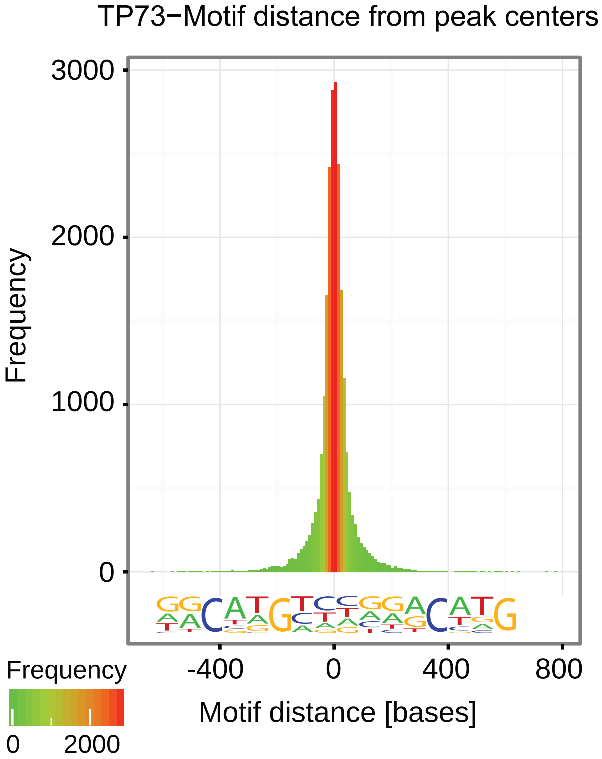


Figure 6.5: Motif of TAp73 binding site, with histogram indicating the frequency of motifs found around peak centers. Figure corresponds to Supplementary Fig. 5c in Nemajerova *et. al.*[23]

in-house Perl script *regionAnnotation.pl*, utilizing the closestBed function from BEDtools[43], was used to retrieve nearest genes to target peaks. As some peaks had nearest genes far away, a distance threshold was set such that only genes with peaks up to 5 kilo base pairs (kbp) upstream or 1kbp downstream were kept. Given this filtering threshold, 66% of all peaks were associated with genes, thus maintaining annotation for more than half of the peaks. Annotating peaks has also provided information about regions they appear in (the majority of peaks, or 52%, appear in intronic regions) and genes types they are associated with (48% appear near protein-coding genes).

6.4 Regulation of differentially expressed genes by p73

In order to determine which DEGs and differentially expressed sRNAs deregulated by knocking out p73 are also directly or indirectly regulated by p73, the results from RNA-seq, sRNA-seq and ChIP-seq datasets were combined. Since the RNA-seq and sRNA-seq datasets were generated from mouse samples, whereas the ChIP-seq dataset was based on human samples, the first step of this combination required converting the resulting human genes from the ChIP-seq analysis into their homologous mouse genes. Once the conversion was done, the DEGs could be overlapped with the p73-peak-related genes.

6.4.1 Homologous genes

The human genes nearby p73 peaks were converted using *R* package *biomaRt*[70]. For the overlap with DEGs, the peak-related genes were converted into mouse ENSEMBL IDs. To overlap the peak-related genes with differentially expressed sRNAs, *piwi*-interacting RNAs (piRNAs) and novel micro RNAs (miRNAs) were excluded, while for the conversion of all other sRNAs with the exception of known miRNAs, mouse ENSEMBL IDs were selected as the target. Alternatively, the known mouse miRNA were converted to their human counterparts by removing the organism prefix ("hsa" or "mmu") and the DNA-end suffix ("3p" or "5p"). The resulting coordinates of differentially expressed miRNAs in the human genome were extended by 10,000 bases in each direction, and those were tested for overlaps with p73 peaks directly. Once all peak-related human genes were converted to their homologous counterparts, the association of DEGs and differentially expressed sRNAs with p73 occupation (peaks) and binding (motifs) could be determined.

6.4.2 Overlaps between p73 peaks and motifs, differentially expressed genes and cilia-related genes

One aspect of the overlap between DEGs and p73 peaks was to determine whether the overlapping genes are still enriched for cilium-related terms. As such, the enrichment tests with WebGestalt were executed again for those overlapping genes at different days (Figure 6.6). While the number of genes submitted to WebGestalt has been cut down (for each day, 37-40% of the original DEGs are kept; see Figure 6.7), the strong association of those genes with ciliogenesis-related terms is maintained, as shown in Figure 6.6.

Another aspect to study the overlap of DEGs with p73 peaks and motifs was by using Fisher's exact test. In each case, the significance of overlap was calculated against the number of DEGs, utilizing a 2x2 contingency table (Table 6.3 shows such a table between DEGs and p73 motif-containing genes which are also cilium-related for day 14).

	DEG	non-DEG
Cilium-related genes with p73 motif	32	1204
Cilium-related genes without p73 motif	79	37104

Table 6.3: Example of contingency table between DEGs and other categories to calculate Fisher's exact test.

Considering each overlap of DEGs with CRGs, p73 peaks and motif-containing regions, a total of 6 combinations were obtained. While the overlaps of DEGs with peak- and motif-containing genes are consistently enriched for all days when considering non-CRGs,

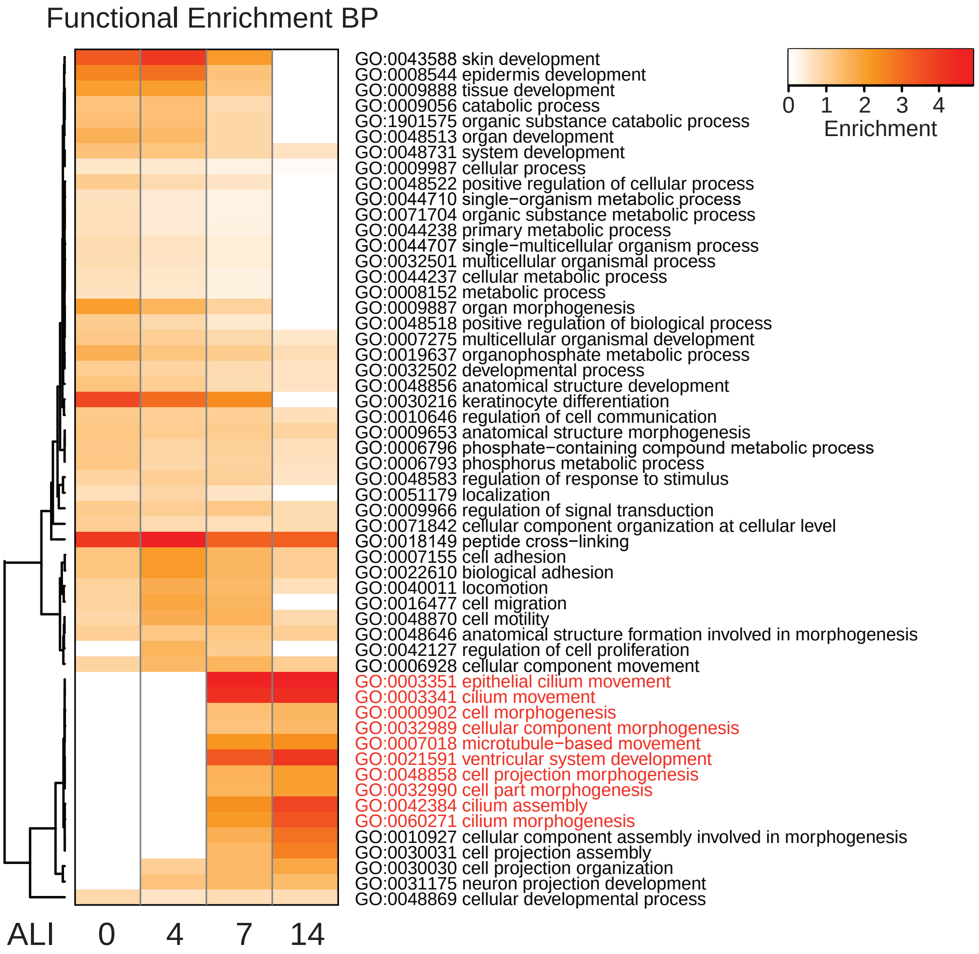


Figure 6.6: Heatmap with enrichment values of top 20 BP categories according to DEGs enriched for p73 at different time-points, with cilia-related terms marked in red. Figure corresponds to Supplementary Fig. 5e in Nemajerova *et. al.*[23]

the same overlaps for CRGs are non-significant for days 0 and 3, yet significant for days 7 and 14 (Figure 6.7). Therefore, it can be concluded that p73 regulation is strongly associated with cilia-related DEGs at later time-points rather than at earlier ones, which also corroborates with Figure 6.6.

6.4.3 Visualization of target genes

Finally, by utilizing the BAM file merging (Section 3.2.1) and generation of visualization files (Section 3.2), the RNA-seq tracks could be visualized for particular genes, specifically *Rfx2* (Figure 6.8a), *Rfx3* (Figure 6.8b) and *FoxJ1* (Figure 6.8c), in order to show that they are indeed differentially expressed between WT and p73KO conditions, as well as show that they contain the *TP73* motif.

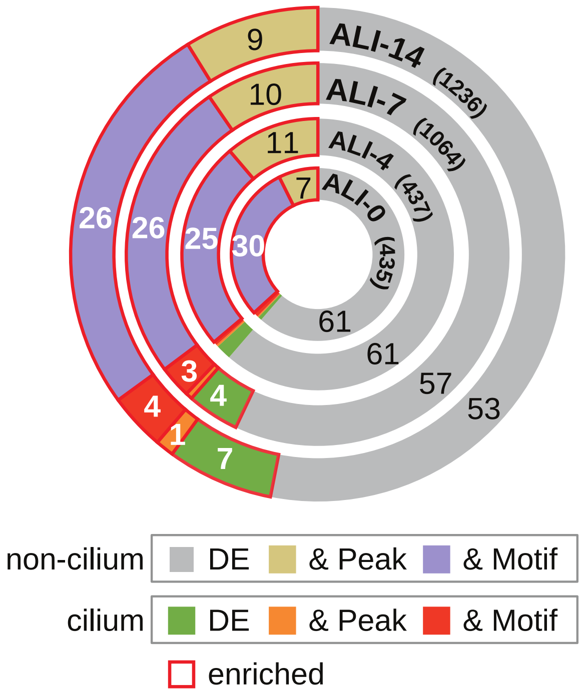
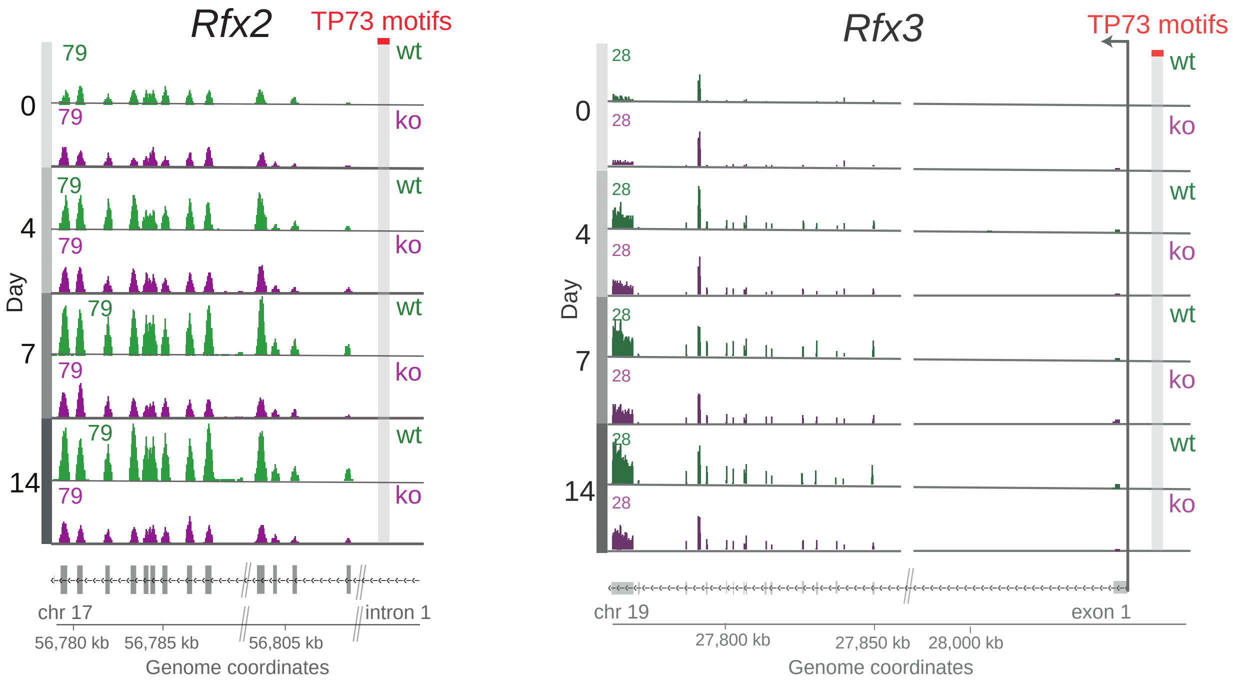


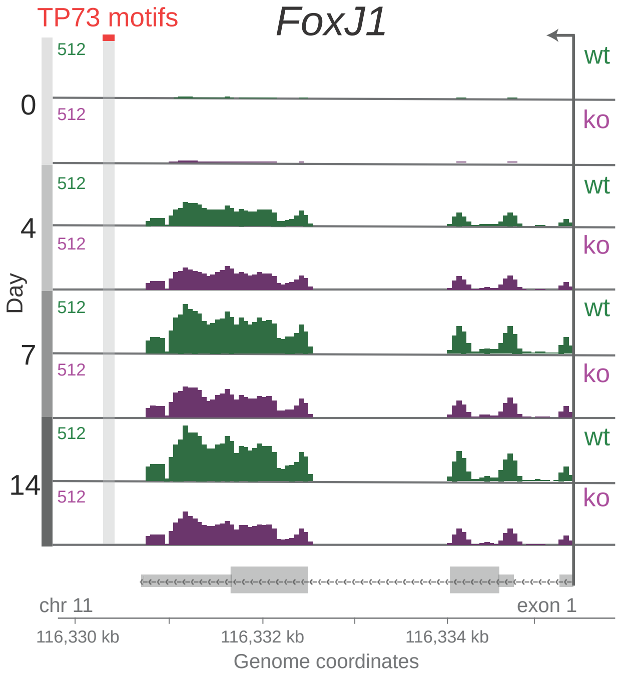
Figure 6.7: Donut plot distinguishing between cilium and non-cilium DEGs, and displaying the percentage of those which are differentially expressed only, differentially expressed and p73 targeted (& Peak) or differentially expressed, p73 targeted and with p73 binding site (& Motif). All regions encompassed by a red square represent overlaps enriched according to Fisher's exact test. Figure corresponds to Figure 4a in Nemajerova *et. al.*[23]

6.5 Overview

This study we conducted, resulting in the publication by Nemajerova *et. al.*[23], has managed to portray the manner by which TAp73 regulates airway multiciliogenesis through deregulation of other cilia-related genes when it is knocked out. TAp73 has also shown its function as a TF, specifically to other ciliogenic TFs such as *FoxJ1*, *Rfx2* and *Rfx3*, as well as the regulatory miRNA *miR34bc*, by having p73 binding motifs in their vicinity. Furthermore, while it was not deregulated by knocking out p73, *Myb* was found to contain TAp73α and TAp73β binding sites as well, thus indicating it to be directly regulated by p73. Finally, incorporating information from previously published data revealed that TAp73 is regulated by MCIDAS/E2F4 complex, which is supposed to drive early cilia development[71]. All this information was used to establish the place of TAp73 in the multiciliogenesis regulatory network, as shown in Figure 6.9.



(A) Rfx2 (B) Rfx3



(C) FoxJ1

FIGURE 6.8: Gene expressions of TFs related to multiciliogenesis in WT and p73 KO conditions, with the genome coordinates appearing at the bottom, and the *TP73* binding motifs appearing as a red box and gray column spanning all tracks. Figures 6.8a, 6.8b and 6.8c correspond to Supplementary Fig. 6a, 6b and Figure 4c, respectively, in Nemajerova *et. al.*[23]

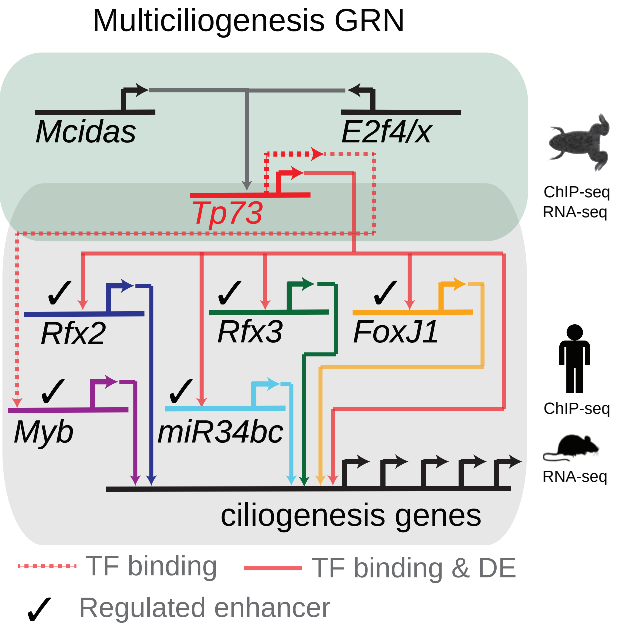


FIGURE 6.9: GRN of multiciliogenesis showing *Tp73* as regulator of ciliogenesis TFs *Rfx2*, *Rfx3*, *FoxJ1*, *Myb* and *miR34bc*. Solid lines show direct binding of TFs to target genes, in addition to being differentially expressed for WT-p73KO testing, while dotted edges indicate TF binding without evidence of differential expression. TFs with checkmarks have been validated for gene expression increase mediated by Tp73 by using a Luciferase enhancer essay. Figure corresponds to Figure 4a in Nemajerova *et. al.*[23]

Chapter 7

Discussion

Particular NGS techniques study specific relationships between the molecules (for example, ChIP-seq focuses on protein-DNA interactions), which is why different studies using the same NGS technique can be analyzed with the same tools and workflows. As such, robust, flexible and fast workflows for NGS techniques, specifically ChIP-seq, MedIP-seq and RNA-seq were developed. In addition, as ChIP-seq was lacking an efficient tool to integrate shift size estimation and enrichment testing, the *R* package *chequeR* was developed as a memory-efficient, fast tool for both analyses. Both NGS analysis workflows and *chequeR* have proven to be efficient in analyzing particular datasets mentioned in Chapters 5 and 6. Given their efficiency, both workflows and *chequeR* can be extended by adding features to them. For example, the workflows can be implemented into a framework to guarantee further stability, in addition to consideration of specific issues within the different analysis steps. Furthermore, the *chequeR* package can be extended by using its cross-correlation calculation for other tests.

7.1 *chequeR*

7.1.1 Comparison of *chequeR* with other shift size estimators

Shift size estimation is usually integrated as part of peak calling tools, such as MACS2[17] and MMDiff[72], and this calculation is used predominantly to obtain the shift size for peak calling. Nevertheless, as previously discussed, the cross-correlation calculated as part of shift size estimation can provide crucial information regarding the enrichment of a ChIP-seq sample in form of NSC and RSC statistics. To date, only *spp*[14], an *R* package developed by ENCODE, calculates these statistics as part of the ChIP-seq data analysis. However, this package does not include a shift size estimation, but rather assumes a constant shift size of 73 hard-coded into the code. In addition, the importance of mappability in calculating shift size estimation has been left out from many shift size estimators, with the exception of MaSC[13], which may result in incorrectly estimated shifts for data with many multi-mapped reads. By integrating those features of mappability and enrichment testing, in addition to focusing on reads in highly-enriched regions, utilizing multiple cores for the analysis, relying on binary input files and using bit-based arrays for cross-correlation calculations, *chequeR* has shown to provide reliable results for shift size estimation with reduced runtime and memory consumption.

7.1.2 Further utilization of read cross-correlations

While the cross-correlation could be used for shift size estimation, it was also used to calculated the NSC and RSC statistics, as mentioned in Sections 2.1.3 and 3.3.3. It could be used, however, to calculate the saturation correlation referred to in Section 3.3.4, as those are generated by the *MEDIPS* package using cross-correlations as well[30]. As mentioned in Section 3.3.4, the saturation test involves taking two subsets of reads within the sample, correlating them against each other and repeating this for different subset sizes to show the increased cross-correlation as the percentage of selected reads increases. To do this, `MEDIPS.saturation` utilizes function `cor` within the *R stats* package, which uses *R* code to correlate arrays together. By simply utilizing a bit array instead of a boolean one used by the `MEDIPS.saturation`, and by calculating the correlations using C++ instead of *R* function `cor`, the memory footprint can be reduced, which is especially beneficial for the saturation tests involving recursive subsets of reads.

Another feature which can be included in *chequeR* is a standard deviation for the fragment. As mentioned in Section 1.2.1, sonication is a stochastic process, resulting in fragments of alternative sizes. While the shift size estimation generates a single value reflecting the fragment length, it is a mean of the fragment length. With the ChIP-seq workflow developed here, only MACS2 utilizes the shift size value, and it only takes a single value (Section 3.4). Nevertheless, some tools which can be used for ChIP-seq analysis as well use both mean and standard deviation of the fragment length, particularly RNA-seq by expectation maximization (RSEM)[73, 74], which attempts to solve the ambiguous alignment of multi-mapped reads (further discussed in Section 7.2.1). By acknowledging that sonication is stochastic, calculating the standard deviation of fragment sizes will allow users to know the sonication efficiency (efficient with small deviations in fragment sizes, or inconsistent with large deviations). As discussed in Section 2.1.1, the cross-correlation denominator contains the square root of multiplying variances from functions f and g, which can be regarded as the standard deviation of the cross-correlation. Applying the same logic to the non-parametric formula used by *chequeR* and defined in Section 2.2.3, the standard deviation of the cross-correlation can be defined as $\sqrt{\left(\sum_{b,b+d \in M_R^d} f(b)\right) * \left(\sum_{b \in M_R^d} g(b+d)\right)}$ and used in tools such as RSEM to improve on the analysis by acknowledging the variation of fragment lengths, and not just their mean.

7.2 NGS analysis workflows

With case studies in Chapters 5 and 6, the NGS workflows have shown to be robust in terms of their analyses and biological validity of their results. As such, it would be possible to incorporate the scripts into frameworks that allow added control of the workflow, error handling and automation. One possibility is Galaxy[49], which is an online tool providing researchers with little to no bioinformatic experience to execute their analyses. In terms of workflow development, Galaxy includes a graphical workflow editor, which can connect tools together by indicating the output from one tool as the input for another, provide control of data formats and tool parameters, and allows a simple automation of the workflow so analyses can be run quickly. Another alternative is Knime[50], which is a Java-developed tool which provides pipeline environment by focuses on workflow modularity. This means that analysis steps are visualized as nodes, where those can be encapsulated into other nodes to provide further modularity. The ability to interact with the process graphically also allows the user to trace the data processing by following which node is currently being executed and in case the process fails, the user knows exactly which node failed. While either Galaxy or Knime provide an efficient framework to combine the scripts together, Knime is more advantageous due to its node visualization that allows better control of data processing, such that if an error occurs, the user knows exactly where the workflow failed.

In terms of specific steps within the workflows that require particular attention, one is the alignment of multi-mapped reads and the other is peak-calling for replicate samples in the ChIP-seq workflow.

7.2.1 Realignment of multi-mapped reads

Reads mapped to the genome are generally short (24-100 bases) and the genome itself is millions of bps long with many repetitive regions, which often results in reads aligning to multiple genomic locations. Since alignment is a core step in all NGS workflows, this ambiguous alignment has been a general issue in NGS analysis. While some tools provide users parameters they can set to return the 'best' alignment according to the mapping quality MAPQ (including BWA[33], SOAP2[35] and bowtie2[27], some studies prefer to remove such multi-mapped reads altogether. Nevertheless, both approaches have flaws in terms of their alignment considerations, since the former approach may select one of the multi-alignments randomly if they have the same MAPQ without knowing which is the 'true' one, and the latter avoids considering them altogether. Multi-mapped reads are important, for example to find protein binding regions as discussed in Section 3.1.2.2, which is why they should not be removed completely, yet at the same time, efforts should be made to remove the ambiguousness from multi-mapped reads and determine more efficiently where they truly align.

One way to solve the issue of ambiguous mapping is through the use of an EM algorithm, which uses expectation and maximization steps to calculate the parameters of a particular model[75]. In the case of multiple aligning positions of a read, by defining the probability of a read to align in a specific location by its features (length, strand it aligns to, read and alignment qualities, and so on), it is possible to define a model that will assess the probability based on those features. Once the model has been defined, the EM algorithm aims to find the model maximum likelihood (ML) by first initializing parameters based on the observed data (E-step), and than maximizing the model likelihood with respect to its parameters (M-step). This is repeated several times until a convergence is obtained, indicating the probability of read alignment in each position.

Two particular tools that use EM on aligned reads are RSEM[73, 74] and eXpress[76]. RSEM was developed for RNA-seq alignment, and originally its model was designed as the probability of a particular read sequence r within R_n random variables (observed data) given a set of parameters: isoform G_n (with n indicating the isoform number), start position S_n (between 1 and the isoform length) and orientation O_n (whether both read and isoform are in the same direction, or one is a reverse complement of another) (Figure 1, [73]). A later extension of this model incorporated global fragment length F, as well as read features length L and quality Q, to consider reads of different lengths and sequencing qualities, respectively. In addition, the extended model included all read features twice to analyse paired-end data (Figure 4, [74]). Alternatively, the eXpress algorithm relies on a model based on fragment length L, target sequence T, position of alignment P and multiple random fragments F generated by pairs of reads from the data. The parameters estimated and optimized are π (sequence biases affecting fragment start and end locations), λ (fragment length distribution), τ (target abundances of fragments) and ϕ (sequencing error probabilities) (Supplementary Figure 11, [76]). Both tools have models consisting of different parameters, thus making it difficult to compare them directly. Nevertheless, this could be done using a R package *rnaseqcomp* to evaluate their specificity and sensitivity using sensitive metrics[77]. The package was specifically used to compare RSEM, eXpress and several other alignment and quantification tools for RNA-seq specifically, with the results showing a slight overperformance by RSEM compared with the other methods. While this shows the usefulness of using RSEM as a tool to correct the ambiguous mapping of reads in RNA-seq data, it was developed specifically for RNA-seq using the transcriptome to correct the multi-mapping issue. As such, it is yet to be tested on DNA-based sequencing (for example ChIP-seq) to determine whether it performs better than bowtie2.

7.2.2 Biological variability between replicates

Currently, the ChIP-seq workflow uses MACS2 in order to call for peaks in different samples, as discussed in Section 3.4. The current workflow takes the replicates in their BAM form and merges them so as to create a single BAM file, and the peaks are called for it. This is useful when the analyzed samples are technical replicates, i.e. the same sample being sequenced multiple times, because the variation between them is a technical one. As such, peaks should be fairly similar between the different replicates, and any differences between them can be attributed to technical errors which can be minimized by replicate merging. However, when dealing with multiple biological replicates, i.e. samples originated from different organisms of the same species with the same properties/conditions/treatments, merging said replicates would result in a reduction of the biological variance between the replicates, which should be accounted for when dealing with biological replicates.

An alternative strategy suggested instead of replicate merging is irreproducibility discovery rate (IDR)[78]. Reproducibility gives a scientific discovery more gravitas by indicating it can be recreated by comparing results obtained from different replicates. The IDR method allows measuring the reproducibility from given biological replicates, and in the context of ChIP-seq it is used to distinguish between peaks of high enrichment (signal) and low enrichment (noise) in multiple replicates. This method takes the sets of ranked peaks from all biological replicates and fits bivariate rank distributions over the replicates in order to generate a p-value for each peak within all sets[78]. This method is often compared to FDR, used to do multiple testing to correct p-values from differential expression analyses, and in the same way, IDR takes the scores obtained from peaks in multiple replicates and compares them such that they are comparable from samples with different background, coverage and quality. Nevertheless, the advantage of comparing the samples in such a way can also be its downfall, since the IDR method would tend to try and rescue the replicate with worse quality, resulting in a bias of the IDR threshold to become more lenient, resulting in many peaks with a large divergence in p-values.

References

[1] Wilhelm J Ansorge. “Next-generation DNA sequencing techniques”. In: *New biotechnology* 25.4 (2009), pp. 195–203.

[2] Andrew Yates et al. “Ensembl 2016”. In: *Nucleic acids research* 44.D1 (2016), pp. D710–D716.

[3] Ron Milo et al. “BioNumbers—the database of key numbers in molecular and cell biology”. In: *Nucleic acids research* 38.suppl 1 (2010). BNID 100434, pp. D750–D753.

[4] A Annunziato. “DNA packaging: nucleosomes and chromatin”. In: *Nature Education* 1.1 (2008), p. 26.

[5] F Sanger et al. “Nucleotide sequence of bacteriophage ϕX174 DNA”. In: *Nature* 265.5596 (1977), pp. 687–95.

[6] Martin Kircher and Janet Kelso. “High-throughput DNA sequencing–concepts and limitations”. In: *Bioessays* 32.6 (2010), pp. 524–536.

[7] Gordon Robertson et al. “Genome-wide profiles of STAT1 DNA association using chromatin immunoprecipitation and massively parallel sequencing”. In: *Nature methods* 4.8 (2007), pp. 651–657.

[8] Laura Arrigoni et al. “Standardizing chromatin research: a simple and universal method for ChIP-seq”. In: *Nucleic acids research* (2015), gkv1495.

[9] Thomas A Down et al. “A Bayesian deconvolution strategy for immunoprecipitation-based DNA methylome analysis”. In: *Nature biotechnology* 26.7 (2008), pp. 779–785.

[10] John C Marioni et al. “RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays”. In: *Genome research* 18.9 (2008), pp. 1509–1517.

[11] Tonatiuh Pena Centeno et al. “Genome-wide chromatin and gene expression profiling during memory formation and maintenance in adult mice”. In: *Scientific Data* 3 (2016), p. 160090.

[12] Orr Shomroni et al. “Genome-wide chromatin and gene expression profiling during memory formation and maintenance in adult mice. [CODE: support files]”. In: (July 2016). DOI: `10.6084/m9.figshare.3487679.v1`. URL: `https://figshare.com/articles/Genome-wide_chromatin_and_gene_expression_profiling_during_memory_formation_and_maintenance_in_adult_mice_CODE_support_files_/3487679`.

[13] Parameswaran Ramachandran et al. “MaSC: mappability-sensitive cross-correlation for estimating mean fragment length of single-end short-read sequencing data”. In: *Bioinformatics* 29.4 (2013), pp. 444–450.

[14] Peter V Kharchenko, Michael Y Tolstorukov, and Peter J Park. “Design and analysis of ChIP-seq experiments for DNA-binding proteins”. In: *Nature biotechnology* 26.12 (2008), pp. 1351–1359.

[15] Stephen G Landt et al. “ChIP-seq guidelines and practices of the ENCODE and modENCODE consortia”. In: *Genome research* 22.9 (2012), pp. 1813–1831.

[16] D Sarkar et al. *chipseq: A package for analyzing chipseq data*. 2013.

[17] Yong Zhang et al. “Model-based analysis of ChIP-Seq (MACS)”. In: *Genome biology* 9.9 (2008), p. 1.

[18] Thomas Derrien et al. “Fast computation and applications of genome mappability”. In: *PloS one* 7.1 (2012), e30377.

[19] Zhong Wang, Mark Gerstein, and Michael Snyder. “RNA-Seq: a revolutionary tool for transcriptomics”. In: *Nature reviews genetics* 10.1 (2009), pp. 57–63.

[20] Michael I Love, Wolfgang Huber, and Simon Anders. “Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2”. In: *Genome biology* 15.12 (2014), p. 1.

[21] Rashi Halder et al. “DNA methylation changes in plasticity genes accompany the formation and maintenance of memory”. In: *Nature neuroscience* 19.1 (2016), pp. 102–110.

[22] ENCODE Project Consortium et al. “An integrated encyclopedia of DNA elements in the human genome”. In: *Nature* 489.7414 (2012), pp. 57–74.

[23] Alice Nemajerova et al. “TAp73 is a central transcriptional regulator of airway multiciliogenesis”. In: *Genes & development* (2016).

[24] S Andrews. *SeqMonk*. 2007.

[25] Huihuang Yan et al. “HiChIP: a high-throughput pipeline for integrative analysis of ChIP-Seq data”. In: *BMC bioinformatics* 15.1 (2014), p. 1.

[26] Björn Grüning et al. “Enhancing pre-defined workflows with ad hoc analytics using Galaxy, Docker and Jupyter”. In: *bioRxiv* (2016), p. 075457.

[27] Ben Langmead and Steven L Salzberg. “Fast gapped-read alignment with Bowtie 2”. In: *Nature methods* 9.4 (2012), pp. 357–359.

[28] Heng Li et al. “The sequence alignment/map format and SAMtools”. In: *Bioinformatics* 25.16 (2009), pp. 2078–2079.

[29] Simon Andrews et al. “FastQC: A quality control tool for high throughput sequence data”. In: *Reference Source* (2010).

[30] Matthias Lienhard et al. “MEDIPS: genome-wide differential coverage analysis of sequencing data derived from DNA enrichment experiments”. In: *Bioinformatics* 30.2 (2014), pp. 284–286.

[31] Sven Heinz et al. “Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and B cell identities”. In: *Molecular cell* 38.4 (2010), pp. 576–589.

[32] Jing Wang et al. “WEB-based gene set analysis toolkit (WebGestalt): update 2013”. In: *Nucleic acids research* 41.W1 (2013), W77–W83.

[33] Heng Li and Richard Durbin. “Fast and accurate short read alignment with Burrows–Wheeler transform”. In: *Bioinformatics* 25.14 (2009), pp. 1754–1760.

[34] Heng Li and Richard Durbin. “Fast and accurate long-read alignment with Burrows–Wheeler transform”. In: *Bioinformatics* 26.5 (2010), pp. 589–595.

[35] Ruiqiang Li et al. “SOAP2: an improved ultrafast tool for short read alignment”. In: *Bioinformatics* 25.15 (2009), pp. 1966–1967.

[36] Ben Langmead et al. “Ultrafast and memory-efficient alignment of short DNA sequences to the human genome”. In: *Genome biology* 10.3 (2009), p. 1.

[37] Helga Thorvaldsdóttir, James T Robinson, and Jill P Mesirov. “Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration”. In: *Briefings in bioinformatics* 14.2 (2013), pp. 178–192.

[38] John W Nicol et al. “The Integrated Genome Browser: free software for distribution and exploration of genome-scale datasets”. In: *Bioinformatics* 25.20 (2009), pp. 2730–2731.

[39] Kate R Rosenbloom et al. “The UCSC genome browser database: 2015 update”. In: *Nucleic acids research* 43.D1 (2015), pp. D670–D681.

[40] Eric S Lander and Michael S Waterman. “Genomic mapping by fingerprinting random clones: a mathematical analysis”. In: *Genomics* 2.3 (1988), pp. 231–239.

[41] Sébastien Lê, Julie Josse, François Husson, et al. “FactoMineR: an R package for multivariate analysis”. In: *Journal of statistical software* 25.1 (2008), pp. 1–18.

[42] Max Koeppel et al. “Crosstalk between c-Jun and TAp73α/β contributes to the apoptosis–survival balance”. In: *Nucleic acids research* (2011), gkr028.

[43] Aaron R Quinlan and Ira M Hall. "BEDTools: a flexible suite of utilities for comparing genomic features". In: *Bioinformatics* 26.6 (2010), pp. 841–842.

[44] Mourad Kaghad et al. "Monoallelically expressed gene related to p53 at 1p36, a region frequently deleted in neuroblastoma and other human cancers". In: *cell* 90.4 (1997), pp. 809–819.

[45] Douglas B Clarkson, Yuan-An Fan, and Harry Joe. "A remark on algorithm 643: FEXACT: An algorithm for performing Fisher's exact test in rxc contingency tables". In: *ACM Transactions on Mathematical Software (TOMS)* 19.4 (1993), pp. 484–488.

[46] Da Wei Huang, Brad T Sherman, and Richard A Lempicki. "Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources". In: *Nature protocols* 4.1 (2009), pp. 44–57.

[47] Jüri Reimand et al. "g: Profiler—a web server for functional interpretation of gene lists (2016 update)". In: *Nucleic acids research* (2016), gkw199.

[48] David Warde-Farley et al. "The GeneMANIA prediction server: biological network integration for gene prioritization and predicting gene function". In: *Nucleic acids research* 38.suppl 2 (2010), W214–W220.

[49] Enis Afgan et al. "The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update". In: *Nucleic acids research* (2016), gkw343.

[50] Michael R Berthold et al. "KNIME-the Konstanz information miner: version 2.0 and beyond". In: *AcM SIGKDD explorations Newsletter* 11.1 (2009), pp. 26–31.

[51] Xiaobei Zhou, Helen Lindsay, and Mark D Robinson. "Robustly detecting differential expression in RNA sequencing data using observation weights". In: *Nucleic acids research* 42.11 (2014), e91–e91.

[52] Alexander Dobin et al. "STAR: ultrafast universal RNA-seq aligner". In: *Bioinformatics* 29.1 (2013), pp. 15–21.

[53] Yang Liao, Gordon K Smyth, and Wei Shi. "featureCounts: an efficient general purpose program for assigning sequence reads to genomic features". In: *Bioinformatics* 30.7 (2014), pp. 923–930.

[54] Yoav Benjamini and Yosef Hochberg. "Controlling the false discovery rate: a practical and powerful approach to multiple testing". In: *Journal of the royal statistical society. Series B (Methodological)* (1995), pp. 289–300.

[55] Alejandro Reyes et al. "Drift and conservation of differential exon usage across tissues in primate species". In: *Proceedings of the National Academy of Sciences* 110.38 (2013), pp. 15377–15382.

[56] Michael S Fanselow. “Factors governing one-trial contextual conditioning”. In: *Animal Learning & Behavior* 18.3 (1990), pp. 264–270.

[57] Jeansok J Kim and Michael S Fanselow. “Modality-specific retrograde amnesia of fear”. In: *Science* 256.5057 (1992), p. 675.

[58] Jason D Runyan, Anthony N Moore, and Pramod K Dash. “A role for prefrontal cortex in memory storage for trace fear conditioning”. In: *The Journal of neuroscience* 24.6 (2004), pp. 1288–1295.

[59] Stefan Bonn et al. “Tissue-specific analysis of chromatin state identifies temporal signatures of enhancer activity during embryonic development”. In: *Nature genetics* 44.2 (2012), pp. 148–156.

[60] Stefan Bonn et al. “Cell type-specific chromatin immunoprecipitation from multicellular complex samples using BiTS-ChIP”. In: *Nature protocols* 7.5 (2012), pp. 978–994.

[61] Artem Barski et al. “High-resolution profiling of histone methylations in the human genome”. In: *Cell* 129.4 (2007), pp. 823–837.

[62] Vicky W Zhou, Alon Goren, and Bradley E Bernstein. “Charting histone modifications and the functional organization of mammalian genomes”. In: *Nature Reviews Genetics* 12.1 (2011), pp. 7–18.

[63] Karin Storm van’s Gravesande and Heymut Omran. “Primary ciliary dyskinesia: clinical presentation, diagnosis and genetics”. In: *Annals of medicine* 37.6 (2005), pp. 439–449.

[64] Dominique Baas et al. “A deficiency in RFX3 causes hydrocephalus associated with abnormal differentiation of ependymal cells”. In: *European Journal of Neuroscience* 24.4 (2006), pp. 1020–1030.

[65] Eric R Brooks and John B Wallingford. “Multiciliated cells”. In: *Current Biology* 24.19 (2014), R973–R982.

[66] Annie Yang et al. “p73-deficient mice have neurological, pheromonal and inflammatory defects but lack spontaneous tumours”. In: *Nature* 404.6773 (2000), pp. 99–103.

[67] Brice Marcet et al. “Control of vertebrate multiciliogenesis by miR-449 through direct repression of the Delta/Notch pathway”. In: *Nature cell biology* 13.6 (2011), pp. 693–699.

[68] Vincenzo Capece et al. “Oasis: online analysis of small RNA deep sequencing data”. In: *Bioinformatics* 31.13 (2015), pp. 2205–2207.

[69] Teunis JP van Dam et al. "The SYSCILIA gold standard (SCGSv1) of known ciliary components and its applications within a systems biology consortium". In: *Cilia* 2.1 (2013), p. 1.

[70] Steffen Durinck et al. "Mapping identifiers for the integration of genomic datasets with the R/Bioconductor package biomaRt". In: *Nature protocols* 4.8 (2009), pp. 1184–1191.

[71] Lina Ma et al. "Multicilin drives centriole biogenesis via E2f proteins". In: *Genes & development* 28.13 (2014), pp. 1461–1471.

[72] Gabriele Schweikert et al. "MMDiff: quantitative testing for shape changes in ChIP-Seq data sets". In: *BMC genomics* 14.1 (2013), p. 1.

[73] Bo Li et al. "RNA-Seq gene expression estimation with read mapping uncertainty". In: *Bioinformatics* 26.4 (2010), pp. 493–500.

[74] Bo Li and Colin N Dewey. "RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome". In: *BMC bioinformatics* 12.1 (2011), p. 1.

[75] Arthur P Dempster, Nan M Laird, and Donald B Rubin. "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the royal statistical society. Series B (methodological)* (1977), pp. 1–38.

[76] Adam Roberts and Lior Pachter. "Streaming fragment assignment for real-time analysis of sequencing experiments". In: *Nature methods* 10.1 (2013), pp. 71–73.

[77] Mingxiang Teng et al. "A benchmark for RNA-seq quantification pipelines". In: *Genome biology* 17.1 (2016), p. 1.

[78] Qunhua Li et al. "Measuring reproducibility of high-throughput experiments". In: *The annals of applied statistics* (2011), pp. 1752–1779.

Acknowledgements

I would like to thank the entire Bonn lab, who were generally fun to work with and did great work even when under heavy pressure. Particularly, I would like to thank the following people:

- Vincenzo Capece, who helped me with some of the analyses and system maintenance, as well as data retrieval
- Ramon Vidal, who gave me advice on how to bridge between the computational and biological aspects of the various studies I was involved in
- Tonatiuh Pena, who helped me organise my projects in terms of logical flow between steps and showed me the proper manner to validate the results at every step

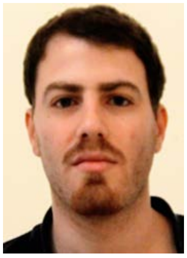
I would also like to express my deepest and most sincere gratitude to my supervisor, Dr. Stefan Bonn, who gave me many varied kinds of projects to expand on my bioinformatic and biological knowledge, and who gave me useful feedback regarding my presentation and organisational skills. Furthermore, I would like to thank him for his patience, for never giving up on me even at extremely stressful moments and for believing in me throughout.

I would also like to thank my Thesis Committee, Prof. Tim Beißbarth and Prof. Stephan Waack, who gave me advice regarding my various projects.

I would like to thank my parents, Ilana and Nir Shomroni, who brought me up to where I am today with hard work, dedication, moral support and much love. Finally, I would like to thank my wife, Ellen Borges, who supported at my most stressful times, and who has so much patience and love for me at my highest and lowest moments.

PERSONAL INFORMATION

Orr Shomroni



Rosmarinweg 1,37081, Göttingen, Germany
+49 (0) 157 3096 8525
orr.shomroni@gmail.com

Sex Male | Date of birth 05/01/1987 | Nationality Israeli, Dutch

APPLICATION AS **Plant bioinformatician for analysis of high-throughput OMICS data**

WORK EXPERIENCE

since 2013

PhD student at DZNE (German Center for Neurodegenerative Disorders)

Von-Siebold-Str 3a, 37075, Göttingen, Germany https://www.dzne.de/

Development of algorithms and workflows for the analysis and visualization of deep sequencing data. Software projects included the co-development of *Oasis*, a web application for the analysis of small RNA-seq data, the development of *chequeR*, an R package for the quality control and shift-size estimation of NGS data, and the *memory-epigenome-browser*, a genome browser for the interactive visualization of NGS data. Data analysis projects included, among others, the identification of epigenetic and gene expression changes during memory formation and maintenance (Nature Neuroscience and Scientific Data 2016) and the elucidation of the gene-regulatory network underlying lung ciliogenesis (Genes & Development 2016).

2011 – 2012

Data analyst at VU Medisch Centrum (VU University Medical Center)

De Boelelaan 1117, 1081 HV Amsterdam, Netherlands https://www.vumc.nl/

Designing NGS pipeline for analysis of whole-genome sequencing data in Python (DNA sequence alignment, variant detection and annotation). Working with Sun Grid Engine, including executing processes in parallel and general maintenance

2010 – 2011

Researcher at LUMC (Leiden University Medical Center)

Albinusdreef 2, 2333 ZA Leiden, Netherlands https://www.lumc.nl/

Analyzing gene expression microarrays using R (chip normalization, differential expression analysis and functional enrichment analysis) to study the relationship between Occulopharyngeal muscular dystrophy and ageing

EDUCATION AND TRAINING

since 2013

PhD, Bioinformatics

Georg-August-Universität Göttingen , Germany

Development of algorithms and workflows for the analysis and visualization of deep sequencing data (see above).
Thesis supervision: Dr. Stefan Bonn from DZNE Göttingen

2009 to 2012

Master’s Degree, Bioinformatics

Leiden University, Netherlands

Mathematical Biology: Metabolic Network Analysis; Molecular Computational Biology; Multi Objective Optimization in Bioinformatics and Chemoinformatics; Evolutionary Algorithms; Natural Computing; Databases and Data Mining; Functional Genomics and Systems Biology; Methodology of Science and Engineering; Pattern Recognition; Advanced Bioinformatics; Basic Java programming; Databases and SQL; Algorithm structures; R scripting; Microarray data analysis; Perl; Matlab.
Thesis supervision: Prof. Dr. Michael Emmerich from Leiden Informatics and Computer Science department

2005 to 2008

BSc, Science

University College Utrecht, Netherlands

Science major: General Biology 1: Molecular Biology and Genetics; Molecular Cell Biology 2; Molecular Genetics of Development and Disease; Advanced Biotechnology; Introduction to Chemistry; Chemistry 2; Biochemistry; Advanced Chemistry; Basic Mathematics: Calculus; Mathematical Methods; Advanced Mathematics; Microbiology, spectroscopy and C and C++ programming labs
Statistics minor: Methods and statistics 1, 2 and 3; Introduction to Mathematical Modeling
Thesis supervision: Frank Witte from University College Utrecht

PERSONAL SKILLS

Mother tongues	Hebrew, **English**
Other languages	Effective operational proficiency in Dutch and **German**

Soft skills

- Excellent oral and written communication skills
- Ample experience in presenting research to large audiences
- Proficiency in teaching bioinformatics to undergraduate and graduate students at the Georg-August University Göttingen (NGS data analysis 2013-2015, Methods in Systems Biology 2015)
- Experienced in teamwork and collaborative projects in an international, multicultural setting
- Goal oriented with ability to handle stress and meet deadlines

Bioinformatic skills

- Expert knowledge of R, shell, python
- Good knowledge of Java Script, HTML5, SQL
- Basic knowledge of Java, C, and C++
- Deep knowledge of bioinformatic algorithms and software for the analysis of NGS data (e.g. samtools, picard, STAR, bowtie, DESeq, MACS) and exposure to workflow management systems (e.g. GALAXY and KNIME)
- Solid grasp of statistical tests and reasoning (especially in the field of NGS)
- Applied knowledge of machine learning algorithms (Random Forests, Bayesian inference)
- Experience with high-performance computational infrastructure and distributed systems
- Good knowledge of LaTex and the Microsoft Office suite

ADDITIONAL INFORMATION

Scientific Publications

- Bettencourt, C., López-Sendón, J. L., García-Caldentey, J., Rizzu, P., Bakker, I. M. C., **Shomroni, O.**, … & de Yébenes, J. G. (2014). Exome sequencing is a useful diagnostic tool for complicated forms of hereditary spastic paraplegia. *Clinical Genetics*, 85(2), 154–8. http://doi.org/10.1111/cge.12133
- Capece, V., Garcia Vizcaino, J. C., Vidal, R., Rahman, R.-U., Pena Centeno, T., **Shomroni, O.**, … & Bonn, S. (2015). Oasis: online analysis of small RNA deep sequencing data. *Bioinformatics* (Oxford, England), 31(13), 2205–7. http://doi.org/10.1093/bioinformatics/btv113
- Bettencourt, C., de Yébenes, J. G., López-Sendón, J. L., **Shomroni, O.**, Zhang, X., Qian, S.-B., … & Rizzu, P. (2015). Clinical and Neuropathological Features of Spastic Ataxia in a Spanish Family with Novel Compound Heterozygous Mutations in STUB1. *Cerebellum* (London, England), 14(3), 378–81. http://doi.org/10.1007/s12311-014-0643-7
- Halder, R., Hennion, M., Vidal, R. O., **Shomroni, O.**, Rahman, R.-U., Rajput, A., … & Bonn, S. (2016). DNA methylation changes in plasticity genes accompany the formation and maintenance of memory. *Nature Neuroscience*, 19(1), 102–10. http://doi.org/10.1038/nn.4194
- Nemajerova, A., Kramer, D., Siller, S. S., Herr, C., **Shomroni, O.**, Pena, T., … & Lizé, M. (2016). TAp73 is a central transcriptional regulator of airway multiciliogenesis. *Genes & Development*, 30(11), 1300–12. http://doi.org/10.1101/gad.279836.116
- Centeno, T. P.*, **Shomroni, O.***, Hennion, M., Halder, R., Vidal, R., Rahman, R.-U., & Bonn, S. (2016). Genome-wide chromatin and gene expression profiling during memory formation and maintenance in adult mice. *Scientific Data*, 3, 160090. http://doi.org/10.1038/sdata.2016.90. *These authors contributed equally to this work

Software

- *Oasis*: Co-development of Oasis (https://oasis.dzne.de), a web application for the analysis of small RNA-seq data
- *chequeR*: Development of chequeR, an R package for the quality control and shift-size estimation of NGS data
- *memory-epigenome-browser*: Co-development of the memory-epigenome-browser (https://memory-epigenome-browser.dzne.de), a genome browser for the interactive visualization of NGS data