EPFL

Thèse n° 7795

Computational study of transcription factor binding sites

Présentée le 7 juillet 2020

à la Faculté des sciences de la vie
Unité du Prof. Deplancke
Programme doctoral en biologie computationnelle et quantitative

pour l'obtention du grade de Docteur ès Sciences

par

Romain Fernand Pietro GROUX

Acceptée sur proposition du jury

Prof. A. Radenovic, présidente du jury
Prof. B. Deplancke, Dr Ph. Bucher, directeurs de thèse
Dr N. Guex, rapporteur
Dr I. Kulakovskiy, rapporteur
Prof. D. Suter, rapporteur

École
polytechnique
fédérale
de Lausanne

2020

Le fait de pouvoir élire librement des maîtres
ne supprime ni les maîtres ni les esclaves.
— Herbert Marcuse

Acknowledgements

To my family, my friends, my colleagues and the countless people who made this work possible, in one way or another.

Lausanne, March 1, 2020 R. G.

Abstract

Any living organism contains a whole set of instructions encoded as genes on the DNA. This set of instructions contains all the necessary information that the organism will ever need, from its development to a mature individual, to environment specific responses. Since all these instructions are not needed at the same time, the gene expression needs to be regulated. Eukaryotic genomes are stored inside nuclei as chromatin. The chromatin is the association of DNA with dedicated storage proteins - the histones - and the necessary machinery to regulate and express genes (RNA polymerases or RNAPs).

In the nuclei, histones are assembled into octamers around which are wrapped 148bp of DNA. This structure is known as the nucleosome. The repetition of nucleosomes along the genome allows to drastically compact the genome, eventually allowing to fit it inside the nucleus. However, this comes at the cost of rendering the DNA sequence inaccessible to DNA readers, such as the RNAPs and transcription factors (TFs).

TFs are a class of proteins that have the remarkable property of recognizing and binding specific DNA sequences. More striking, each TF can recognize a multitude of different - but similar - DNA sequences providing TFs with a wide sequence specificity range. Eventually, this allows the cell to recruit TFs at dedicated locations in the genome called regulatory elements (REs).

The action of TFs at REs is crucial to gene expression. Indeed, TFs are involved in many processes such as the opening of the chromatin structure or the recruitment of RNAPs. However if TFs can influence the chromatin structure, the opposite is also true as histones impede TF binding on DNA. Thus the regulation of genes relies on a subtle and complex crosstalk between the chromatin and TFs.

To better understand how TFs and chromatin interact together to regulate gene expression, I conducted several projects prospecting TF binding specificity and the chromatin structure at REs in human.

First, I used ENCODE next generation sequencing (NGS) data to explore how TF binding influences the nearby nucleosome organization and the propensity of some TFs to bind together. The results suggest that regular nucleosome arrays are found near all TFs. They also point out two special cases. When CTCF binds with the cohesin complex, it seems to drive the nucleosome organization, which is a unique feature among all TFs investigated. Additionally I present evidence supporting that EBF1 is a pioneer factor - a special class of TFs able to bind nucleosome.

Second, I developed several clustering algorithms and software to partition genomic regions

according to NGS data and/or on their DNA sequences. These methods allow to discover important trends, for instance different nucleosome architectures. I illustrated the usefulness of these methods for the study of chromatin accessibility data and the identification of REs.
Third, I participated to the assessment of SMiLE-seq, a new microfluidic device that generates TF specificity data. The creation of TF specificity models and their comparison with other publicly available models demonstrated the value of SMiLE-seq to study TF specificity.
Finally, I participated in the development of a software that predicts TF binding sites. A careful benchmarking suggested that this software is - at the time of writing - the best available software in terms of speed while showing other performances similar to its competitors.

Résumé

Tout organisme vivant contient un jeu d'instructions encodé sous la forme de gènes dans son ADN. Ces instructions contiennent toutes les informations nécessaires à la vie de l'organisme en question, de son développement à l'adaptation à des conditions environnementales spécifiques. Étant donné que ces instructions ne sont pas toutes nécessaires en même temps, l'expression des gènes doit être régulée. Chez les eucaryotes, le génome est stocké dans dans le noyau sous forme de chromatine. La chromatine est l'association de l'ADN, de protéines dédiées au stockage de celui-ci et de toute la machinerie nécessaire à la régulation et à l'expression des gènes. Dans le noyau, les histones sont assemblés en octamères autour desquels s'enroulent 148pb d'ADN et forment le nucleosome. La répétition de nucleosomes le long du génome permet de le compacter fortement et d'être contenu dans le noyau. Cependant, cela se fait au prix de rendre certaines séquences d'ADN inaccessibles aux facteurs le lisant, tels que la machinerie de transcription (les ARNs polymerases ou ARNPs) ou les facteurs de transcription (FTs).

Les FTs forment une classe de protéines qui possède la remarquable capacité de pouvoir reconnaître et lier spécifiquement certaines séquences d'ADN. Plus encore, chaque FT est capable de reconnaître une multitude de séquences différentes – mais similaires – étendant ainsi sa spécificité de séquence. Cela permet à la cellule de recruter certains FTs à des endroits précis du génome appelés éléments régulateurs (ERs).

Le rôle des FTs au niveau des ERs est crucial pour l'expression des gènes. En effet, les FTs sont nécessaires pour plusieurs processus tels que la décompaction locale de la chromatine ou le recrutement de la machinerie transcriptionnelle. Cependant, si les FTs sont capables d'influer sur la structure de la chromatine, l'inverse est aussi vrai. Les histones sont capables d'empêcher la liaison des FTs à l'ADN. La régulation de l'expression des gènes s'appuie donc sur un phénomène subtil et complexe d'interactions entre la chromatine et les FTs.

Afin de mieux comprendre ces interactions et comment elles participent à la régulation de l'expression des gènes, j'ai conduit plusieurs projets ayant pour sujet la spécificité des FTs et la structure de la chromatine dans les ERs.

Premièrement, j'ai utilisé les données de séquençage à haut débit (SAD) générées par ENCODE afin d'explorer comment la liaison des FTs à l'ADN influence l'organisation des nucleosomes proches ainsi que la tendance de certains FTs à s'associer. Les résultats suggèrent que des agencements de nucléosomes réguliers se trouvent autour des sites de liaison de tous les FTs. Cependant, seule l'association entre CTCF et la cohésine semble capable d'influencer cette organisation. De plus, d'autres observations suggèrent fortement que le FT EBF1 est un facteur

pionnier – une classe de FTs spéciaux capables de lier les nucléosomes.
Deuxièmement, j'ai développé plusieurs algorithmes et sofwares de clustering permettant de grouper les régions du génome en fonction de données SAD et/ou de leur séquence ADN. Ces méthodes permettent d'identifier des tendances, par exemple différentes organisation des nucléosomes. J'ai illustré l'utilité de ces méthodes au travers de l'étude de l'accessibilité de la chromatine et de l'identification d'ERs.
Troisièmement, j'ai participé à l'évaluation du SMiLE-seq, une nouvelle plateforme microfluidique permettant de générer des données sur la spécificité des FTs. La mise au point de modèles de spécificité et leur comparaison avec d'autres modèles disponibles a permis de démontrer la valeur du SMiLE-seq pour les études de spécificité des FTs.
Finalement, j'ai participé au développement et à l'évaluation d'un software prédisant les régions liées par un FT, le long d'un génome. Le processus d'évaluation suggère que ce software est actuellement – au moment de la rédaction – le meilleur en terme de rapidité tout en présentant d'autres performances similaires à ses compétiteurs.

Contents

Acknowledgements i

Abstract (English/Français/Deutsch) iii

1 Introduction 1

Introduction 1

1.1 About chromatin . . . 1
1.1.1 The chromatin structure . . . 2
1.1.2 The chromatin is dynamic . . . 2
1.1.3 About nucleosome positioning . . . 4
1.2 About transcription factors . . . 7
1.2.1 TF co-binding . . . 7
1.3 Gene regulation in a nutshell . . . 9
1.3.1 The chromatin barrier . . . 9
1.3.2 TFs cooperative binding . . . 9
1.3.3 Pioneer TFs . . . 10
1.3.4 Regulatory elements . . . 10
1.3.5 The genome goes 3D . . . 11
1.4 Measuring chromatin features . . . 12
1.4.1 Measuring TF binding in vivo . . . 12
1.4.2 Measuring TF binding in vitro . . . 13
1.4.3 Measuring nucleosome occupancy . . . 14
1.4.4 Digital footprinting . . . 15
1.5 Modeling sequence specificity . . . 17
1.5.1 Aligning binding sites . . . 19
1.5.2 Platitudes . . . 20
1.5.3 Predicting binding sites . . . 20
1.6 Over-represented patterns discovery . . . 21
1.7 General aims of this work . . . 23

2 Laboratory resources 25

Laboratory resources 25

2.1 Mass Genome Annotation repository . . . 25
2.1.1 MGA content and organization . . . 26
2.1.2 Conclusions . . . 27
2.2 Eukaryotic Promoter Database . . . 28
2.2.1 EPDnew now annotates (some of) your mushrooms and vegetables . . . 29
2.2.2 Increased mapping precision in human . . . 30
2.2.3 Integration of EPDnew with other resources . . . 31
2.2.4 Conclusions . . . 31
2.2.5 Methods . . . 31

3 ENCODE peaks analysis **33**

ENCODE peaks analysis **33**
3.1 Data . . . 34
3.2 ChIPPartitioning : an algorithm to identify chromatin architectures . . . 35
3.2.1 Data realignment . . . 37
3.3 Nucleosome organization around transcription factor binding sites . . . 37
3.4 The case of CTCF, RAD21, SMC3, YY1 and ZNF143 . . . 40
3.5 CTCF and JunD interactomes . . . 46
3.6 EBF1 binds nucleosomes . . . 48
3.7 Discussion . . . 50
3.8 Methods . . . 51
3.8.1 Data and data processing . . . 51
3.8.2 Classification of MNase patterns . . . 52
3.8.3 Quantifying nucleosome array intensity from classification results . . . 53
3.8.4 Peak colocalization . . . 54
3.8.5 NDR detection . . . 54
3.8.6 CTCF and JunD interactors . . . 57
3.8.7 EBF1 and nucleosome . . . 58

4 SPar-K **61**
4.1 Algorithm . . . 62
4.2 Implementation . . . 63
4.3 Benchmarking . . . 63
4.3.1 K-means . . . 66
4.3.2 ChIPPartitioning . . . 66
4.3.3 Data . . . 66
4.3.4 Performances . . . 67
4.4 Partition of DNase and MNase data . . . 67
4.5 Conclusions . . . 70

5 SMiLE-seq data analysis **71**

SMiLE-seq data analysis **71**

5.1 Introduction . . . 71
5.2 Hidden Markov Model Motif discovery . . . 73
5.3 Binding motif evaluation . . . 74
5.4 Results . . . 75
5.5 Conclusions . . . 77

6 PWMScan **79**
6.1 Algorithms . . . 79
6.1.1 Scanner algorithm . . . 80
6.1.2 Matches enumeration and mapping . . . 80
6.2 PMWScan architecture . . . 81
6.3 Benchmark . . . 83
6.4 Conclusions . . . 85

7 Chromatin accessibility of monocytes **87**
7.1 Monitoring TF binding . . . 87
7.2 The advent of single cell DGF . . . 88
7.3 Open issues . . . 88
7.4 Data . . . 88
7.5 Identifying over-represented signals . . . 89
7.5.1 ChIPPartitioning algorithm . . . 89
7.5.2 EMSequence algorithm . . . 89
7.5.3 EMJoint algorithm . . . 93
7.5.4 Data realignment . . . 94
7.5.5 Soft aggregation plots . . . 94
7.6 Data processing . . . 95
7.7 Results . . . 95
7.7.1 Aligning the binding sites . . . 95
7.7.2 Exploring individual TF classes . . . 97
7.8 Discussions . . . 99
7.9 Methods . . . 100
7.9.1 Code availability . . . 100
7.9.2 Data sources . . . 100
7.9.3 Data post-processing . . . 101
7.9.4 Model extension . . . 101
7.9.5 Extracting data assigned to a class . . . 102
7.9.6 Programs . . . 104
7.9.7 Fragment classes . . . 105
7.9.8 Simulated sequences . . . 106
7.9.9 Binding site prediction . . . 106
7.9.10 Realignment using JASPAR motifs . . . 107
7.9.11 Per TF sub-classes . . . 109

8 Discussion **111**

Discussion **111**

9 Published articles **115**

A Supplementary material **117**
A.1 ENCODE peaks analysis supplementary material 118
A.2 SPar-K supplementary material 128
A.3 SMiLE-seq supplementary material 141
A.4 Chromatin accessibility of monocytes supplementary material 141
A.4.1 Fragment size analysis 141
A.4.2 Measuring open chromatin and nucleosome occupancy 142
A.4.3 Evaluation of EMSequence and ChIPPartitioning 145
A.4.4 Other supplementary figures 151

Bibliography **155**

Bibliography **168**

Curriculum Vitae **169**

1 Introduction

Each living organism contains DNA which is the molecular support on which genes are encoded. Genes are the hereditary unit of life and code for a set of instructions involved in all the aspects of life, from an organism development to the functions of a specific cell type. However, since all these instructions are not needed at the same time, gene expression needs to be regulated.

Transcription factors (TFs) are a class of nuclear proteins that can bind to specific DNA sequences and drive target gene expression. Thus TFs are a major regulator of gene expression.

The results reported in this work can be sub-divided in two sub-topics. The first topic focuses on data mining projects and reports the results of different computational genomic research projects that were focused on the characterization of the chromatin structure in the vicinity of TF binding sites as well as TF-TF interaction identification and on the modeling of TF DNA sequence specificity. The second topic focuses on the development of algorithms and computational methods to solve bioinformatic problems that are met in genomics. Several algorithms to identify important chromatin signatures or over-represented DNA sequences in the genome using a data partitioning approach are presented as well as a software that predicts TF binding sites in a genome given a specificity model. The two topics are not entirely separated and can be presented jointly in some chapters.

1.1 About chromatin

In eukaryotes, the DNA is stored in the nucleus. Each human cell contains about two meters of DNA. In order to fit the DNA inside the nucleus, the cells have to organize and compact the genome while maintaining it readable. Unbeatable, evolution came out with an elegant solution : the chromatin. The chromatin is the association of the DNA with specialized proteins - the histones - around which it wraps. Other families of proteins are also found in the chromatin, such as RNA polymerases, histone chaperones, helicases or TFs.

1.1.1 The chromatin structure

In human, there are four major (canonical) histones : H2A, H2B, H3 and H4. These four histones are found assembled together into an octamer, composed of two H2A/H2B and two H3/H4 hetero-dimers, around which 146-148bp of DNA wraps, forming the nucleosome core particule (that I will later simply refer to as "nucleosome", Figure 1.1A). The DNA is kept wrapped around the histone octamer because of strong electrostatic interactions. Indeed, the DNA backbone, which is negatively charged in physiological conditions, shows a high affinity for the positively charged histones. As a consequence, the nucleosome is a quite stable structure.

The histone proteins are highly conserved among eukaryotes at both the sequence and the structure level. All the histones share the overall same design. They are composed of a N-terminal tail, a central histone-fold domain and of a C-terminal tail. Histones associate with each other through their histone-fold domains which compose the center of the nucleosome. In contrast, the histone N-terminal tails are protruding out of the nucleosome and are hotspots for post-translational modifications (PTMs) (Kouzarides, 2007).

For completeness, it should be mentioned that "variant histones" - also called "replacement histones", by opposition to the "canonical replicative histones" - exist and can replace canonical histones in nucleosomes, at specific genome locations, to fulfill dedicated functions (Henikoff and Smith, 2015). However, this topic is outside the scope of this work.

The genome is organized into a repetition of nucleosomes, each separated by a linker DNA, forming the 11-nm chromatin fiber. This chromatin conformation is quite relaxed and the DNA accessible. The H1 linker histone can be recruited in the chromatin, in which case it binds the linker DNA and makes it inaccessible. The 11-nm fiber is itself stored into a denser and less accessible structure called the 30-nm fiber (Figure 1.1B). Eventually, higher order structure are achieved, further increasing the genome compaction level (McGinty Robert K. and Tan Song, 2014).

It is now commonly accepted that the compaction of the genome comes with a trade-off. The DNA sequences found in nucleosomes are thought to be unaccessible for DNA reading processes such as TF binding whereas the linker DNA remains accessible (Weirauch and Hughes, 2011). Thus storing the genome impedes its readability. Because transcribing genes is all about reading the DNA template, the state of the chromatin eventually impacts gene expression. Consequently, the cell faces a situation where it needs to keep only the immediately useful genomic regions readable while keeping the ability to open/close other regions on demand.

1.1.2 The chromatin is dynamic

Because the required activated genes may vary over time, for instance because of lineage commitment, the chromatin structure needs to be adapted. Some regions need to become

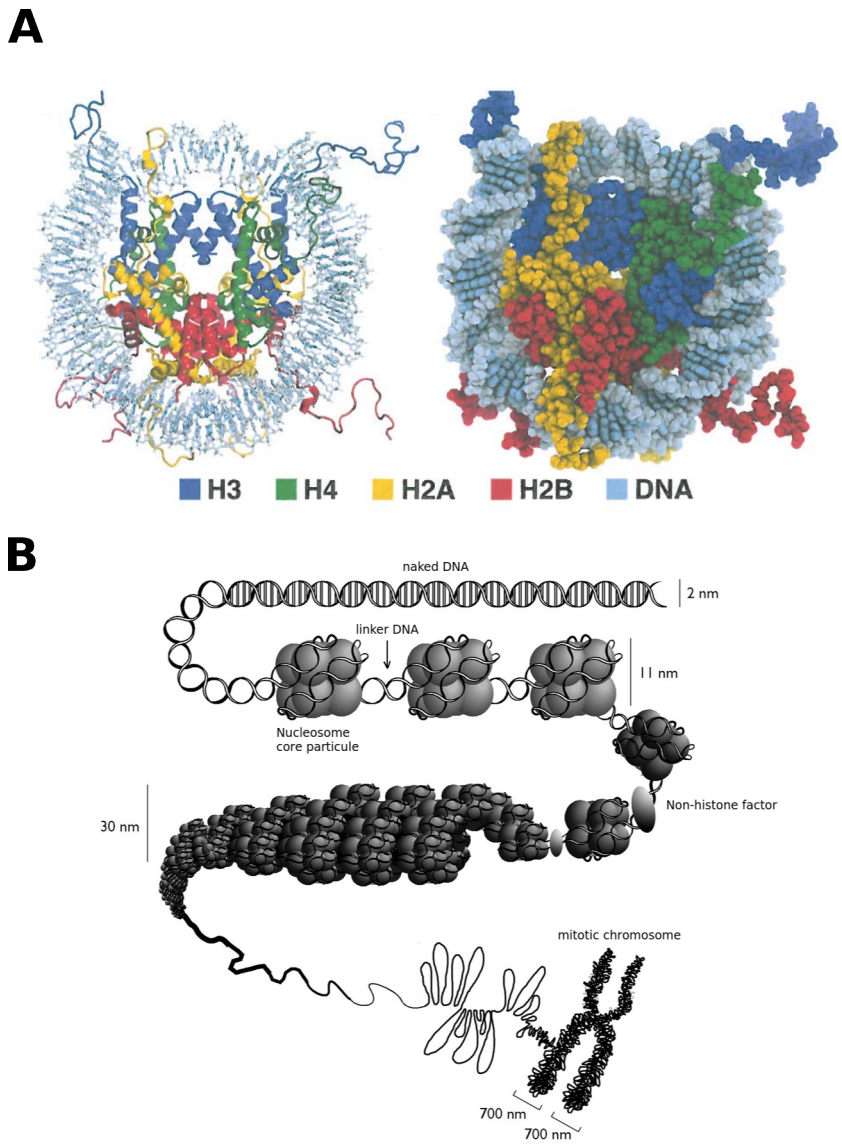


Figure 1.1 – A Top view of a nucleosome core particle (NCP) displayed as a ribbon representation on the left and space filling representation on the right. The NCP is made of a four hetero-dimers histone octamer around which 146-148 DNA bp wraps. The histone tails protrude out of the NCP and are accessible to other factors, unlike the inner part of the histone octamer. **B** The chromatin structure. Inside eukaryotes, DNA is wrapped around histones cores forming nucleosomes. Nucleosomes can then be organized into higher-level helical-like structure, compacting the DNA. The ultimate compaction state is reached at mitosis metaphase, when the mitotic chromosomes are visible. Figure and legend taken and modified from McGinty Robert K. and Tan Song (2014).

accessible in order to be read while others are not needed anymore. Consequently, the chromatin is a highly dynamic structure that undergoes constant modifications. Two broad families of chromatin modifiers exist : ATPase chromatin remodelers and histone modifiers.

ATPase chromatin remodelers are a group of proteins that are able to affect the chromatin packaging by interfering directly with the nucleosomes, at the cost of hydrolyzing ATP molecules. Chromatin remodelers can be subdivided into 4 sub-groups, each fulfilling a different function (Längst and Manelyte, 2015). SWI/SNF members can slide and/or evict nucleosomes from DNA and are linked with chromatin opening. ISWI members tend to recognize unmodified H4 histones and catalyze nucleosome spacing and chromatin compaction. CHD members are less well functionally characterized but bear chromo domains that allow them to recognize histone methylation. Finally, INO80 members seem to be able to slide and evict nucleosomes and seems to be involved in DNA repair and replication.

Histone modifiers are enzymes that can deposite PTMs on the histone tails. Different types of PTMs exist such as acetylation or methylation. Each histone has several residues that can be modified, sometimes together. This leads to an astonishingly high number of combinations. So far more than a hundred histone PTMs have been identified, each linked with different biological functions. If the deposition of PTMs is made by dedicated factors (referred to as writers), their recognition is also performed by dedicated proteins (referred to as readers) (Kouzarides, 2007; Hyun et al., 2017). This allows histone PTMs to be used to recruit specific factors at given genomic location. For instance, H3 lysine 4 di-methylation (H3K4me2) has been shown to be enriched at the promoters of actively transcribed genes and at enhancers (Zhou et al., 2011; Hyun et al., 2017) and to be specifically recognized by CHD1, a member of the CHD chromatin remodelers (Hyun et al., 2017).

1.1.3 About nucleosome positioning

The advent of MNase-seq allowed to draw high resolution maps of nucleosome occupancy in many species, such as in yeast (Kubik et al., 2015), mouse (West et al., 2014) or human (Schones et al., 2008; Gaffney et al., 2012).

The wealth of data collected allowed to determine that nucleosomes do not cover the genome uniformly. Nucleosomes rather seem to show preferred locations where they sit at. Interestingly, single nucleosomes can be visualized from batch sequencing experiments, indicating that an important fraction of the cells bear nucleosomes at the same positions. In these cases, the nucleosomes are said to be "phased" or "strongly positioned" (see Figure 1.2A).

Nucleosome arrays are a striking case of strongly positioned nucleosome. Their most prominent feature is the regularity of the spacing between the pairs of nucleosomes that are part of the array. Arrays can occur throughout the human genome (Gaffney et al., 2012). However, there are regions where they are enriched, for instance at the CCCTC-binding factor (CTCF) binding sites (Fu et al., 2008). It has been proposed that the arrays resulted from

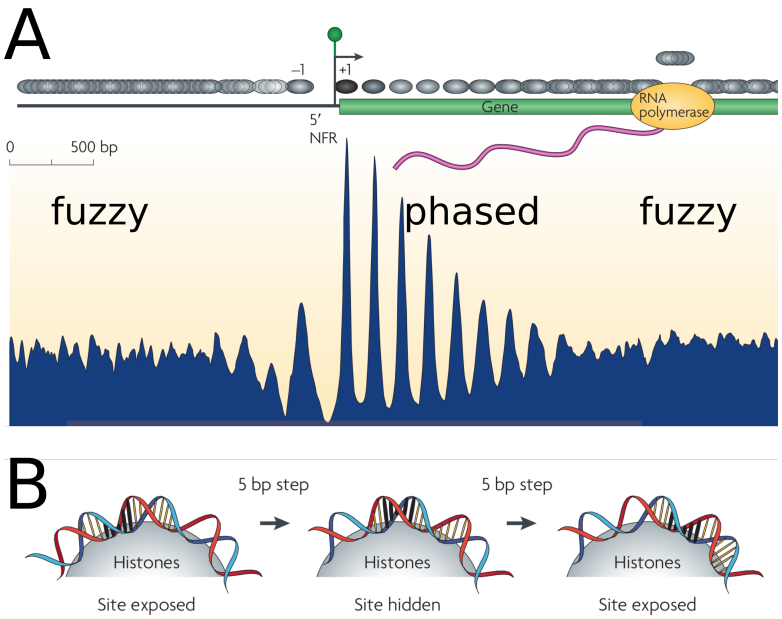


Figure 1.2 – Nucleosome positioning A Activated gene transcription start site (TSS) region. The nucleosomes located immediately downstream of the TSS show a strong positioning. The positioning of the first nucleosome can be influenced by sequence preferences. Eventually the phasing is propagated to neighboring nucleosomes through statistical positioning. The nucleosome array is not anymore visible as the nucleosomes become fuzzily positioned among the cells. **B** Influence of the rotational positioning on the sequence accessibility. Left, a sequence (indicated by the black 'rungs' on the DNA helix) has its major groove facing toward the nucleosome outside and is accessible. Center, a 5bp rotation of the nucleosome hides the sequence as its major groove is now facing the histone octamer. Right, another 5bp rotation makes the sequence accessible again. Both images are taken and adapted from (Jiang and Pugh, 2009).

the nucleosomes organizing with respect to a barrier (or anchor). In this case, the barrier would be CTCF. The regular array organization has been proposed to be propagated far from their anchors because the immediately flanking nucleosome positions are constrained by the barrier. In turn, these nucleosomes become a barrier for the following ones, and so one, eventually forming the array. At some point, because the degree of constraint diminishes at each new nucleosome, the nucleosomes are not sufficiently phased anymore throughout the cell population. They become fuzzy and the signal blurs out at some point. This phenomenon is referred to as "statistical positioning" (Jiang and Pugh, 2009).

Another important driver of nucleosome positioning is the DNA sequence. For instance, strongly positioned nucleosomes are also visible at the transcription start sites (TSSs) of activated genes. However, unlike for CTCF binding sites, the DNA sequence composition seem to be a major factor driving the nucleosome positioning (Dreos et al., 2016). To wrap around the histone octamer the DNA should be curved, which requires some flexibility. WW (W=A/T) and SS (S=C/G) dinucleotides have been shown to curve the DNA by extending the major and the minor groove respectively (Jiang and Pugh, 2009). However, because the major and minor grooves precess around the DNA helix axis, each groove alternatively faces the nucleosome center (the histone octamer) and the nucleosome outside (the opposite direction) every 5bp (thus the DNA helix periodicity is 10.4bp, see Figure 1.2B). Consequently, dinucleotides favoring DNA flexibility are required to occur at different locations around the nucleosome, according to their effect on the DNA helix structure. For instance, stretching the major groove needs to occur when it is facing the nucleosome outside, to force the adjacent DNA segment to be curved toward (around) the nucleosome center. If a nucleosome is bound to a favorable sequence, the next most likely favorable binding sites are located 10bp upstream or downstream. These correspond to the locations at which all the dinucleotides will reacquire the same orientation with respect to the histone octamer. This is referred to as "rotational positioning" (Jiang and Pugh, 2009). In 2011, Trifonov identified the YRRRRRYYYYYR (where R=A/G and Y=C/T) consensus sequence to be a nucleosome positioning sequence matching these criteria (Trifonov, 2011). The first and last positions indicate the cyclic nature of this pattern.

Interestingly, the exact positioning of a nucleosome has a deep impact on the accessibility of the DNA. None 10bp displacements have the potential of changing a sequence orientation with respect to the histone core and thus its accessibility.(Figure 1.2B).

In vivo, both statistical and rotational positioning occur. Additionally, chromatin remodelers are also constantly catalyzing thermodynamically unfavorable nucleosome displacements in exchange of ATP hydrolysis. It is likely that each nucleosome is subjected to all of these phenomenons. However, on a single nucleosome basis, one positioning may predominate over the others.

1.2 About transcription factors

TFs are a special class of proteins that is crucial for gene expression regulation. TFs have the special ability to recognize specific DNA sequences among others. Once recruited on the DNA template, TFs have the ability to regulate transcription by promoting or repressing the activity of the RNA polymerase II complex (RNAPII). In the first case, one speaks of (transcriptional) activators, in the second of (transcriptional) repressors. TFs share a modular architecture. Two types of domains are of particular importance for TF functions: the DNA binding domain (DBD) and the activation domain (AD).

The DBD allows TFs to bind their DNA target. Many different DBDs exist, each one being structurally different than the others. The DBD structure if typically used to classify TFs into families. This is for instance the case in the TFclass database (Wingender et al., 2013). In metazoans, TFs have been grouped into four distinct super-families : i) the basic domain TFs, ii) the zinc coordinated TFs, iii) the helix-turn-helix TFs and iv) the β scaffold TFs. Each type of domain has a different structure and thus can interact with different DNA structures and sequences (Weirauch and Hughes, 2011). Of further importance, a single DBD is able to recognize different yet similar sequences. Because the sequence differences have an impact on the TF-protein interaction interface, each sequence is bound with a different affinity. Biologically, having high and low affinity binding sites may be useful to tune the intensity of one TF action on a given gene.

In addition to their DBD, many TFs also bear an AD that is important for the regulation of transcription. ADs allow TFs to regulate gene expression directly, by interacting with the basal transcriptional machinery, or indirectly by recruiting co-regulators. Coupled with specific DNA recognition, this allows TFs to regulate the transcription of specific regions of the genome. Whether TFs exert an activator or a repressor role, depends on the exact interaction they can exert with the transcriptional machinery and on the co-regulators they can recruit (Latchman, 1997).

Ultimately, the activity of a TF is regulated by controlling its access to the DNA. This can be done by sequestrating it in the cytoplasm (by any mean) or even by occupying its binding sites to impede the TF recruitement on the genome (Latchman, 1997).

1.2.1 TF co-binding

The four above-mentioned TF super-families offer a huge variety of different TFs and thus allows a substantial complexity in terms of transcriptional regulation. Nonetheless, life further expanded the possible complexity of regulatory wiring by evolving different types of combinatorial TF co-binding (Field et al., 2011). By TF co-binding, I mean a functional association of TFs that requires them to bind either as a complex or in close vicinity. Furthermore, from a strictly DNA-centric point of view, the binding of each TF does not need to be synchronous. One TF may bind after the other, even after it left.

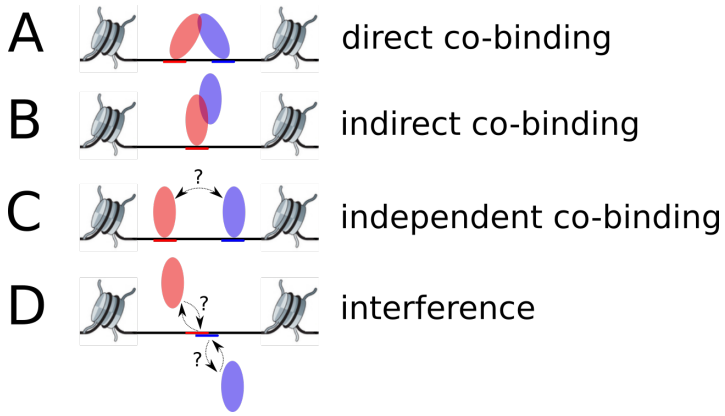


Figure 1.3 – Possible interaction scenarios between TFs A Direct co-binding. The TFs dimerize and bind together on DNA. **B** Indirect co-binding. Both TF dimerize but only one binds the DNA, the other (the blue) is the tethering factor. **C** Independent co-binding. Both TF bind in close vicinity but without forming a complex. Both TFs may not be necessarily bound at the same time. **D** Interference. Both binding sequences partially or totally overlap each other.

First, two TFs can dimerize, forming either homo- or hetero-dimers, and bind to DNA using both DBDs (Figure 1.3A). This is for instance the case of the members of the basic domain super-family, which contains the leucine zipper and helix-loop-helix families, which are obligated dimers in order to bind DNA (Weirauch and Hughes, 2011). This can be referred to as "direct co-binding".

Second, two TFs can dimerize and bind to DNA using only one of the DBDs. This will result in having one of the TF binding to DNA while the other one is tethering DNA through its interaction with the other TF (Figure 1.3B). This can be referred to as "indirect co-binding".

Third, two TFs can both bind DNA using their own DBDs, in close vicinity but without any physical interaction (Figure 1.3C). This is for instance the case at distal REs, where many TFs can be found to be bound at the same time. Synergistic co-binding of several TFs has been proposed as a mechanism by which close chromatin structures could be opened and distal regulatory elements (REs) activated (Field et al., 2011; Heinz et al., 2015). On the other hand, the binding of different TFs to a given region can be asynchronous. This is the case for TFs involved at different time of the activation cascade, such as what is happening during macrophage and B cell progenitors commitment (Heinz et al., 2010). This can be referred to as "independent co-binding".

Finally, two TF binding sequences (or motif occurrences) can overlap (Figure 1.3D). The outcome is not clear but this can result in different plausible scenarios such as both TFs transiently binding the DNA or one TF winning the competition and stably binding, sterically excluding other TFs. This can be referred to as "interference".

1.3 Gene regulation in a nutshell

The regulation of gene expression is a highly complex biological phenomenon which allows a proper allocation of resources to each individual gene such that the overall gene product output fits the cell needs as precisely as possible.

The status of a gene, at a given time, is the results of the actions of activating and repressing mechanisms that, *in fine*, modulate the activity of the transcriptional machinery. This modulation takes place at different steps of the activation cascade of the transcriptional machinery. The really first step that occurs is the binding of general TFs, such as TFIID, that allows to recruit the catalytic subunits of the RNAPII, forming the pre-initiation complex (PIC). The further downstream steps include the recruitment of transcriptional regulator and chromatin remodelers, the proper positioning of the full RNAPII complex at the TSS and the activation of the RNAPII complex. This section will briefly introduce some of these aspects to provide the necessary information for the further understanding of this work by the reader.

1.3.1 The chromatin barrier

As discussed above (see section 1.1), the genome is stored as chromatin in the nuclei. Because nucleosome are bound to the DNA, they compete with other factors for binding. As such, the chromatin structure is a barrier to the recruitment of the PIC. On the brigh side, this is though to limit spurious activation of the RNAPII (von Bakel, 2011). On the other hand, this obviously also suppresses any gene expression. In human, the observation that TF binding is hindered by nucleosomes and that REs are nucleosome depleted suggest the existence of a mechanism that opens the chromatin at REs (von Bakel, 2011).

1.3.2 TFs cooperative binding

The cooperative binding of TFs has been demonstrated to be able to open closed chromatin. In essence, this is a step-wise process during which a first TF binds its target on an accessible linker, leading to the destabilization of a neighboring nucleosome. This in turn increases the accessibility of a second TF binding site that can be engaged, further opening the chromatin. Eventually, the nucleosome is displaced or even evicted and the chromatin is locally opened (von Bakel, 2011). ATPase chromatin remodelers and/or of histone modifier can be recruited by TFs to set up a proper chromatin environment (von Bakel, 2011).

The conditions for this phenomenon to ignite are not precisely known however several hypotheses and observations are of interest. First, compacted chromatin has been observed, in vitro, to undergo spontaneous transient local openings at the nucleosome entry sites. This phenomenon has been referred to as "nucleosome breathing" (von Bakel, 2011). This has the potential of creating windows of opportunity for TFs to engage their binding sites, in nucleosome arrays. Second, it has been hypothesized that, in human, the regions that show

a high nuclosome density may facilitate the exclusion of the H1 histone. The rational is that the DNA linkers between any two nucleosomes is too short for H1 to bind. Eventually, H1 exclusion prevents the inclusions of these regions in more condensed chromatin structures while leaving the linkers somewhat accessible (Field et al., 2011). Together with nucleosome breathing, this has the potential of creating engageable - but not open - windows throughout the genome.

1.3.3 Pioneer TFs

Alternatively, a special class of TFs named "pioneer factors" have been shown to be able to bind their target in a closed chromatin environment and to induce chromatin opening after binding (Zaret and Carroll, 2011; Iwafuchi-Doi and Zaret, 2014).

The case of the prototypical pioneer factor FoxA1 (also called HNF3) is enlightening regarding the mechanistics of pioneer TFs. In liver, FoxA1 is able to bind the inactive *albumin* enhancer and prime it for activation (Cirillo et al., 2002). The enhancer activation is possible because of the hybrid nature of FoxA1. It binds DNA through its DBD, which has a similar structure to the H1 linker histone. Strikingly, FoxA1 can bind its motif directly on the nucleosome surface. Furthermore, FoxA1 posses a C-terminal domain that directly binds the histone core, which leads to the chromatin opening (Cirillo et al., 2002). Alternatively, FoxA1 is also able to recruit co-regulators via its N-terminal trans-activation domain (Zaret and Carroll, 2011).

Currently, many other pioneer TFs have been discovered, such as Oct4, Sox2 and Klf4 (Soufi et al., 2015) - also known together with myc as the "Yamanaka factors" - or PU.1 which has been shown to induce nucleosome remodeling at macrophage and B-cell specific enhancers (Heinz et al., 2010). Interesting in this regard, most of the TFs that have been discovered to drive cellular reprogramming, such as the Yamanaka factors which have been shown to be sufficient to reprogram fibroblasts into stem cells (Takahashi and Yamanaka, 2006) are pioneer TFs.

1.3.4 Regulatory elements

Chromatin opening and the recruitment of the transcriptional machinery do not happen at random in the genome but is concentrated at REs. The specific recruitment of the transcriptional machinery regulators at given genomic locations allows to concentrate the regulatory signals on specific target genes. REs can be divided in two broad classes based on their vicinity to the gene(s) they regulate : proximal REs - or promoters - and distal REs. Both classes interact together by the mean of the genome 3D structural organization.

Promoters are located immediately upstream of the target genes they regulate. Promoters functions are to recruit the RNAPII and position it properly for transcription. Interestingly two constrasting promoter groups have been identified with respect to their chromatin architectures (Cairns, 2009). The first group includes house keeping genes. This group chromatin

architecture tends to be constituvely open with a nucleosome depleted region (NDR), promoting gene expression. The second group contains highly regulated genes. Unlike the first group, these promoters tend to be constituvely covered by nucleosomes, hindering TFs and RNAPII recruitment. Their activation requires an active chromatin remodeling that is carried out by SWI/SNF ATPase family members. However, in both cases the chromatin is remodeled and a NDR is formed. The NDR usually contains core regulatory elements (CREs) involved in the recruitment of general TFs leading to the assembly of the RNAPII (Lenhard et al., 2012).

Distal REs are located at distances that vary from kilobases to megabases from their target genes and have the ability to influence gene expression positively, in which case they are referred to as 'enhancers', or negatively, in which case they are referred to as 'silencers'. Distal REs are enriched with closely spaced TF recognition sequences that serve for the recruitement of TFs. In turn, TFs allow to recruit other transcriptional co-regulators such as histone modifiers (Heinz et al., 2015). Through chromatin looping phenomenons, the recruited TFs (and all other factors) are brought in close spatial vicinity with target gene promoters. This increases TF concentrations (as well as other regulatory factors bound) at the promoter level and allows to strengthen regulatory signals directly where the RNAPII is sitting (Heinz et al., 2015).

Distal REs are not always active. Instead they are highly cell line specific and thus are important determinant of the cell identity (Heinz et al., 2015). Distal REs activation requires to open the chromatin in order to be accessible for TFs to bind. Currently, both cooperative TF binding and pioneer TFs are though to be involved in chromatin opening and remodeling (Heinz et al., 2015). Upon chromatin opening, specific histone PTMs are deposited, such as H3 lysine 4 mono-methylation (H3K4me1), H3K4me2 or H3 lysine 27 acetylation (H3K27Ac) (Zhou et al., 2011). For instance, during B-cell and macrophage lineage commitment, PU.1 and EBF1 are essential TFs which action activate cell type specific enhancers, leading to the enforcement of differential genomic programs (Boller et al., 2016; Heinz et al., 2010). Failure to do so leads to lineage commitment defects (Hagman and Lukin, 2005; Kurotaki et al., 2017).

1.3.5 The genome goes 3D

Finally, another layer of complexity involved in the regulation of gene expression can be added : the 3D organization of the genome. Nowadays it is clear that in the nucleus, the genome spatial organization is tightly regulated and that it has a functional meaning (Bonev and Cavalli, 2016).

As described above, enhancers and promoters physically interact together through loops. These looping phenomenons do not happen at random. The genome is organized into compartments, also called topological association domains (TADs). A TAD can be seen as high level chromatin loop in which the physical interactions between loci are favored compared to interactions with loci outside of the TAD. As a matter of fact, enhancers scope of action is limited to the TAD they are located in. Thus TADs can be seen as a functional regulatory genomic domains.

TADs are thought to be established and maintained by a dedicated set of structural proteins and complexes including CTCF and the cohesin complex (Bonev and Cavalli, 2016). CTCF seem to have two major functions. First it seems to facilitate promoter/enhancers interactions, within TADs and to promote gene expression. Second, CTCF has been found to be enriched at TAD borders and seems to be important for their proper delimitation (Ong and Corces, 2014), likely through a loop extrusion mechanism (Ghirlando and Felsenfeld, 2016). This second function is compatible with the insulator function of CTCF. Because it marks the boundary between TADs, enhancer/promoter interactions over this limit cannot happen. Finally, CTCF is often found to interact with the cohesin complex (Stedman et al., 2008). The cohesin complex is composed of four members : SMC1, SMC3, RAD21 and either STAG1 or STAG2 (Losada, 2014). Together they form a ring-like structure in which two DNA molecules are trapped and maintained together. This structure is one of the mechanisms allowing to pinch DNA and to form loops. The cohesin complex is important for both promoter/enhancer interactions and TADs maintenance (Losada, 2014; Bonev and Cavalli, 2016).

1.4 Measuring chromatin features

The occupancy of the different components of the chromatin, the chromatin accessibility or even the sequence preference of TFs can be measured using dedicated assays. This section introduce the necessary information to further understand this work.

1.4.1 Measuring TF binding in vivo

The advent of chromatin immuno-precipitation (ChIP) is central for the study of TFs. In essence, it consists in extracting the chromatin from the cell nuclei, shearing it either mechanically or enzymatically and adding an antibody (Ab) against a DNA binding protein of interest. The IP step allows to pull-down the Ab, its target as well as the DNA fragment it is bound to.

Different methods, with varying throughput, have been used to identify of the purified DNA fragments. First, specific loci of interest were assayed by PCR. Then the growing availability of DNA microarrays allowed to drastically increase the throughput by testing a wide number of pre-selected loci at once (Odom, 2011). Finally, protocols subjecting the purified DNA to high throughput sequencing (ChIP-seq) (Barski et al., 2007; Robertson et al., 2007) allowed to identify the bound loci in an agnostic way, with an unprecedented throughput.

ChIP-seq has truly revolutionized genomics and the study of TFs. In a single assay, it is possible to obtain a digital readout of TF binding sites. Mapping the sequenced reads to the genome of interest allowed to create a per position occupancy score, creating a digital readout of the TF occupancy. However, because the TF binding sites are smaller than the sequenced fragments, the precise location of the TF binding remains unknown.

Interestingly, ChIP-seq allows to list the regions of the genome that are occupied and also

provide an estimate of the binding affinity for the regions. Indeed, the stronger the propensity to bind to a given sequence (the affinity), the higher the probability of binding. This should be proportionally reflected in the density of signal (Jothi et al., 2008). Thus ChIP-seq allows to identify regions +/- 100bp in which a TF binds. Nonetheless, it is possible to identify over-represented DNA sequence motifs from these regions using *de novo* motif discovery methods (see section 1.5.1). Typically, the identified sequence motifs belong to i) the TF of interest and/or ii) co-binders (see section 1.2.1).

1.4.2 Measuring TF binding in vitro

In vivo measurement of TF binding as several drawbacks. ChIP-seq allows to estimate the binding specificity of TF however it has been proposed that *de novo* motif discovery method mostly capture the high affinity features of the TF binding specificity (Stormo and Zhao, 2010). Additionally, in vivo, the chromatin exert an effect on TF binding (see section 1.3). In regard to these limitations, in vitro binding assays offer experimental solutions to investigate i) TF binding over a wider range of affinities and ii) TF intrinsic specificity, without the chromatin influence. In the recent years, many different technologies have been developed to investigate TF binding in vitro.

Microfluidic devices are typically composed of hundreds (if not more) of individual chambers and of the necessary piping to flow all the necessary reagents within each cell to run as many reactions in parallel. The reaction chambers are small and allow to use microliter reaction volumes. Maerkl and colleagues (Maerkl and Quake, 2007; Geertz et al., 2012) have developed the mechanically induced trapping of molecular interactions (MITOMI). This assay is based on a microfluidic device that allows to run hundred of affinity assays with a given TF, in parallel. Each assay is run using a different designed oligo-nucleotide of known sequence.

Originally, systematic evolution of ligands by exponential enrichment (SELEX) has been designed to discover few high affinity binding sequences (Tuerk and Gold, 1990). The SELEX assay was adapted to become high throughput SELEX (HT-SELEX, (Roulet et al., 2002; Zhao et al., 2009; Jolma et al., 2010)). HT-SELEX assays a TF specificity by allowing a binding reaction between the TF and tens of millions of different DNA sequences of typically 20-30bp. The bound DNA molecules are purified by pulling down the TF. The purified DNA can either be sequenced using high throughput sequencing or be subjected to another cycle of selection. Repeated cycles allow to isolate higher affinity binders, eventually only returning a few hundreds. Under a limited number of cycles, this method has a large dynamic scale of binding affinities (Stormo and Zhao, 2010) and allows to obtain a digital readout. However, the repeated cycles can introduce biaises that are hard to model in order to properly estimate the binding affinities.

For completeness, protein binding microarrays (PBMs, (Bulyk et al., 2001; Mukherjee et al., 2004; Berger et al., 2006)) should also be mentioned. Typically, a PBM device is a chip on which tens of thousands of DNA probes are immobilized. The probes are arranged into spots such

that only one probe specie is present per spot. A purified TF is then added on the chip and the TF binding is revealed using a fluorescent-labeled Ab. Because the identity of the probe specie in each spot is known, the affinity to this specie can be directly measured as the intensity of the fluorescent signal. The higher the affinity for a probe specie, the more TF binds to the spot, the stronger the fluorescent signal. The most important limitation of PBM is its limited space on the chip which restrict the number of different spots that can be present. Assaying all possible 4^L sequences of a given length L is not possible passed a given length. To circumvent this, the sampling of the deposited sequences should be performed with caution to maximize the information on a single chip (Berger and Bulyk, 2009). Additionally, it has been suggested that the position of the spot on the chip could influence the TF binding.

1.4.3 Measuring nucleosome occupancy

The micrococcal nuclease (MNase) - an endo-exo nuclease - is a key factor in producing nucleosome occupancy maps. Subjecting a chromatin extract to a MNase treatment, upon proper experimental conditions, releases "a 'ladder' of discrete DNA fragments" (Voong et al., 2017) which sizes correspond to mono-, di-, tri-, and so on nucleosome fragments. The MNase is able to digest accessible linker DNA (endo-nuclease activity) and to trim the nick edges (exo-nuclease activity). The nucleosomal DNA is protected from digestion as the histone octamer sterically hinders the MNase access to its substrate (Voong et al., 2017).

Originally, MNase treated DNA was selectively amplified using PCR to map precise nucleosomes. The advent of microarray technologies allowed to interrogate entire genomes, even though the created map had relatively low resolution (Jiang and Pugh, 2009). Eventually, this limitation was circumvented by subjecting the MNase treated chromatin fragments to next generation sequencing (-seq) (Schones et al., 2008). The advent of MNase-seq lead to the creation of high resolution - down to individual nucleosomes - genome-wide nucleosome maps (Schones et al., 2008; Gaffney et al., 2012; West et al., 2014; Kubik et al., 2015).

Mapping the sequenced fragment of MNase-seq assay against a genome of reference produces a digital readout of the nucleosome density per genomic position. If single-end sequencing is used, the nucleosome center (the dyad) can be inferred by shifting the read position by $\tilde{7}$0bp. If paired-end sequencing is performed, mono-nucleosome fragments can be selected based on their sizes (150bp) and the dyads can be inferred as being their central positions.

If MNase-seq allows to unravel nucleosome occupancy with an unprecedented resolution, it also suffers some limitations.

First, MNase has been demonstrated to exhibit a sequence preference toward A/T rich sequences, which could potentially lead to an overdigestion of nucleosome fragments in A/T rich regions (Voong et al., 2017). Second, some nucleosomes have been demonstrated to be "fragile" to the experimental conditions. In yeast, specific nucleosomes were found to be sensitive to the MNase concentration and could only be detected with reduced MNase

concentrations (Kubik et al., 2015). Here, the MNase sequence preference may be at play. But another case of fragile nucleosomes was found in human, independently of the use of MNase. In this case, the fragile nucleosomes contained replacement histones and were sensitive to regular salt concentrations used during a ChIP-seq experiment (Jin et al., 2009). Thus, it is likely that MNase-seq is not able to map all the nucleosome in a given genome.

1.4.4 Digital footprinting

Digital genomic footprinting (DGF) methods are a powerful mean to reveal the active REs in a genome. DGF gives a measure of the chromatin accessibility genome-wide. The essence of DGF assay relies on reagents - enzymes or chemicals (this work will only cover enzymes) - that are able to generate single- or double-stranded DNA cleavages into a chromatin-stored DNA template (Tsompana and Buck, 2014; Vierstra and Stamatoyannopoulos, 2016)

DGF assays relies on a selective degradation of the loci stored in accessible chromatin followed by high throughput sequencing (-seq). The degradation of the accessible chromatin regions can be performed using either DNaseI (DNase-seq, Neph et al. (2012)) or a modified Tn5 transposon system (assay for transposable accessible chromatin, abbreviated ATAC-seq, Adey et al. (2010); Buenrostro et al. (2013)).

DNaseI is an endonuclease. Under proper ionic conditions, this enzyme introduces double-strand breaks in the genome based on the DNA accessibility, with a minor sequence specificity (Herrera and Chaires, 1994), as shown in Figure 1.4A. On a technical note, DNase-seq assays are quite sensitive assays. Achieving a proper chromatin degradation - that is, avoiding over-digestion - is not an easy task and requires careful enzymatic titrations.

ATAC-seq assays rely on a modified Tn5 transposase enzyme to selectively fragment the accessible regions of the genome (Adey et al. (2010); Buenrostro et al. (2013), Figure 1.4B). The enzyme inserts small double-stranded barcodes inside the DNA wherever it is accessible resulting a the creation of double-strand breaks. This process, known as tagmentation, allows to i) fragment the genome and ii) insert sequencing barcodes at once. It should be noted that the Tn5 acts as an homodimer and thus inserts two copies of the same adaptors separated from each other by 9bp (Adey et al., 2010).

In both cases, the genome is chopped down into fragments starting (and ending) wherever the chromatin is accessible. A sequencing library is then created from the fragments and their ends are sequenced using high throughput sequencing technologies. Finally, the insertion sites are located by mapping the sequenced reads against the reference genome of interest. This eventually leads to the creation of a per position cut (nicks for DNase-seq, insertions for ATAC-seq) density (Figure 1.4C).

Whenever a TF, a nucleosome or any other factor is engaged in a binding interaction with the DNA template, steric hindrance phenomenons protect the DNA from being degraded by the enzyme, leading to the creation of a typical signal diminution called "footprint" (Figure 1.4C).

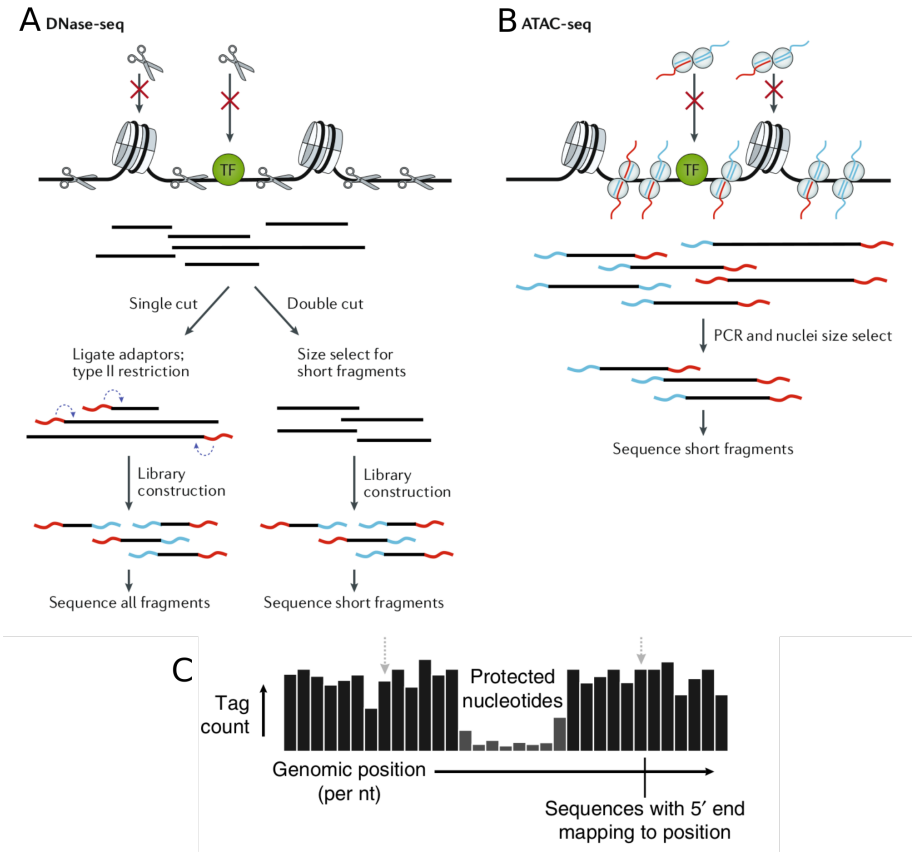


Figure 1.4 – Digital footprinting : A DNase-seq uses the endonuclease DNaseI to cleave DNA within accessible chromatin. Endonuclease cleavage is greatly attenuated at the protein-bound loci (the red crosses denote cleavage blockade). Accessible library fragments are generated by barcoding each cleavage site independently after restriction digestion (single cut) or as proximal cleavage pairs (double cut). **B** Assay for transposase-accessible chromatin using sequencing (ATAC-seq) uses a hyperactive transposase (Tn5) to simultaneously cleave and ligate adaptors to accessible DNA. **C** The purified DNA fragments are then subjected to massively parallel sequencing and mapped to the reference genome to generate a digital readout of per-nucleotide insertion (DNaseI nick or Tn5 transposition event) genome-wide. Figure and legend taken and adapted from (Vierstra and Stamatoyannopoulos, 2016; Klemm et al., 2019).

Formally, a footprint is a degradation signal drop over a DNA sequence that is protected from degradation because of binding event (Vierstra and Stamatoyannopoulos, 2016), but I will later use the term "footprint" the refer to signal drop in a degradation signal for aggregation profiles as well.

DGF assays encounter a yet ever-growing popularity because of the wealth of data produced in a single experiment. Indeed, instead of running thousands - one per transcription factor (TF) - of ChIP-seq assays to know where each TF is binding, it is sufficient to run a single chromatin accessibility assay. Additionally, DGF is totally agnostic in the sense that it does not require a prior knowledge of the factors to look for. However, if DGF reveals the active regulatory regions, it does not provide the information about which factors bound. Methods to circumvent this limitation will be discussed later.

1.5 Modeling sequence specificity

In the nucleus, the number of potential binding sites for a TF is incredibly high. Most of these sites are non-sites and a minority *are bona fide* binding sites. The ability of a TF to distinguish between both is called "specificity". Additionally, all binding site are not equal. Some are bound tighter than others. The forces and the propensity with which a sequence is bound by a given TF is called the affinity.

Modeling TF specificity has been an crucial issue in biology as it allows to predict where a given TF binds in the genome and to infer its regulatory targets. However because TFs recognize degenerated sequence motifs, solving this problem turned out to be complicated.

The physics approach to PWMs

Let us assume a simple binding reaction between a TF TF and a DNA sequence S, at the equilibrium, TF + S $\rightleftharpoons$ TF$\cdot$S. The chemical definition of the affinity is the association constant k_a :

$$k_a = \frac{[TF \cdot S]}{[TF] \times [S]} \tag{1.1}$$

where the square brackets indicate the concentrations. Once at the equilibrium, this reaction releases a standard free energy equals to :

$$\Delta G^0 = -R \times T \times ln(k_a) \tag{1.2}$$

where R is the perfect gaz constant and T the absolute temperature. When $\Delta G^0 < 0$, $K > 1$ indicating that, at the equilibrium, the product TF-S of the reaction is favored. The more

product at equilibrium, the stronger the affinity of TF for S (see equation 1.1). The more energy is released by the reaction, the stronger S is bound by TF.

From this, for a given TF, it is possible to create a ΔG^0 table for all possible sequences which would allow to predict the TF specificity. In competition, the probability of binding to a sequence is directly proportional to the affinity. However, this is at least labor intensive and at most experimentally intractable.

To circumvent this obstacle, the hypothesis of positional independence was formulated. The hypothesis states that the reaction ΔG^0 is an additive function of the individually recognized bases. Thus, each base recognized does not influence the recognition of any base at any other position. From the beginning, this hypothesis was recognized to be an approximation that would apply to most of the cases but no all. Even though, this assumption was a subject of controversy (Man and Stormo, 2001; Bulyk et al., 2002). However, it is nowadays commonly admitted that, even if the assumption is violated in some cases, it does still allow to represent accurately most of the cases (Benos et al., 2002; Zhao and Stormo, 2011).

Originally, the measurement of affinity was a labor intensive - and still is today - and was assayed using gel shift assays. Thus performing one affinity measurement for each of the 4^L sequences of length L did not belong to the realm of the possible. Instead, working under the hypothesis of positional independence allowed to drastically lower the experimental workload and only requires to run an assay for each of the $3 * L + 1$ single position mutants. In turn, this allows to construct a matrix containing the $\Delta\Delta G^0$ induced by each base being recognized at each position of the binding site (Stormo and Fields, 1998). This matrix is called a position weight matrix (PWM) and allows to predict the affinity of TF for any sequence. In such PWMs, the strongest affinity site is usually given a ΔG^0 equals to 0 and all other $\Delta\Delta G^0$ values are expressed with respect to this strongest affinity site.

Finally, it should be noted that inside a PWM, positive (unfavourable) $\Delta\Delta G^0$ are often turned to negative values such that the highest affinity binding site achieves the maximal score (corresponding to the lowest (favourable) ΔG^0).

The statistical mechanic approach to PWMs

The TF sequence specificity has also be tackled from a statistical point of view. Given an alignment A of binding sites of length L, it is possible to construct a letter probability matrix (LPM) or a letter frequency matrix (LFM) that contains either the number of time each base appears at each position of A or the corresponding probabilities, respectively.

In 1986, Schneider and colleagues quantified the sequence conservation at each position of A using the information content (Schneider et al., 1986). In 1987, Berg and von Hippel demonstrated that this statistical approach was mathematically related to the estimation of

binding affinities (Berg and von Hippel, 1988). Given A, a PWM can be build using :

$$\lambda\varepsilon_{jb}^{obs} = ln\frac{f_{j0}^{obs}}{f_{jb}^{obs}} \tag{1.3}$$

where ε_{lb}^{obs} is a dimension less positive number that expresses the decrease of binding energy by the binding of base b at position j, f_{jb}^{obs} is the probability of observing a base b at position j in the binding site, f_{j0}^{obs} is the probability of the base present at position j in the strongest binder present in A and λ is a dimensionless scaling factor.

Stormo and Fields used a slightly different approach (Stormo and Fields, 1998) and proposed to link the information content of A to the estimation of the binding energy using :

$$W_{b,j} = log_2\frac{F_{b,j}}{p(b)} \tag{1.4}$$

where $f_{b,j}$ is the number of occurrences of base b at position j in the alignment A, $p(b)$ the probability of a given base b and W is the PWM. Note that equations 1.3 and 1.4 give inversly proportional results because of the inverted fractions.

1.5.1 Aligning binding sites

The above solution request to have a set of aligned binding sites A to compute the PWM. However, most of the times, we do not have access to this information. Rather, we have a set of longer sequences in which we know that the TF of interest binds, but not exactly where.

Many algorithms - typically called "*de novo* motif discovery" algorithms - have been developed to construct a matrix from a set of unaligned sequences. The algorithm developed by Stormo and Hartzell (Stormo and Hartzell, 1989) finds an optimal alignment by maximizing a modified version of the alignment information content (Schneider et al., 1986). Alternatively, the algorithms developed by Hertz and Stormo (Hertz et al., 1990) and Lawrence and Reilly (Lawrence and Reilly, 1990) build a LPM that maximize the likelihood of observing the data under the hypothesis that they have been generated from the discovered model. Lawrence and Reilly's algorithm solves the alignment problem using an Expectation-Maximization (EM) algorithm, which makes their algorithm a heuristic. This type of framework has also been used to develop MEME, excepted that it explicitly models the data as a mixture of binding and non-binding sites (Bailey and Elkan, 1994).

A

A	-18	95	-426	-607	192	-129	-333	-284	197	-286	-27
C	-38	-36	-857	-387	-328	-10000	-506	193	-571	33	73
G	68	5	-398	192	-563	184	-501	-492	-499	-86	-68
T	-43	-199	196	-278	-326	-10000	194	-567	-488	104	-17

B

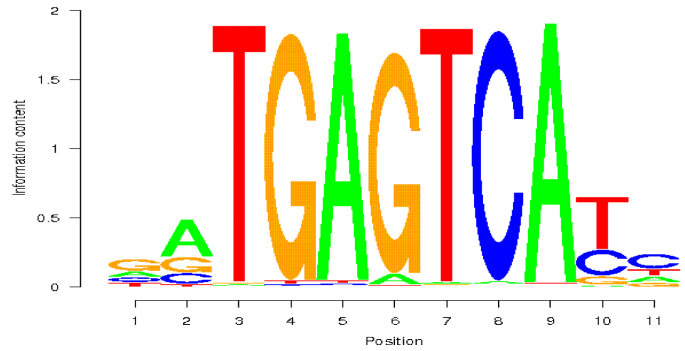


Figure 1.5 – Position weight matrix : **A** Human JunD (JUND_HUMAN.H10MO) PWM from the HOCOMOCO version 10 collection Kulakovskiy et al. (2016). **B** Corresponding PWM logo. The palindromic nature of the recognized motif is explained by the fact that JunD belongs to the basic helix-loop-helix family. As such, it is obligated to hetero- or homo-dimerize with another member of its family to bind DNA. Both **A** and **B** were taken from http://ccg.vital-it.ch/ssa/oprof.php

.

1.5.2 Platitudes

LFMs and LPMs are often use to summarize an alignment A and are often falsely referred to as PWMs. In all cases, LFMs and LPMs can always be converted to a PWM.

Overall, PWMs (and related matrices) are conceptually straightforward to apprehend and easy to work with. For instance, they can be visualized as a sequence logo (Schneider and Stephens (1990), Figure 1.5). Because of this and of the possibility to estimate affinity values from a sequence alignment, PWMs are popular and remain the most widely used type of model to represent TF specificity.

Nowadays, large collections of TF specificity matrices are available from publicly available libraries such as JASPAR (Khan et al., 2018), HOCOMOCO (Kulakovskiy et al., 2018) or CIS-BP (Weirauch et al., 2014) to cite only the most famous.

1.5.3 Predicting binding sites

TF specificity models are typically used for classifications problems. The problem can be stated as follows : given a sequences S of length L and a PWM **W** of dimensions $Lx4$, predict

whether S will be bound. A common way is to define a threshold score t and compute

$$score(S) = \sum_{i=1}^{L} W_{i,b}$$
$$\text{with } b = \begin{cases} 1, & \text{if } s_i = \text{A} \\ 2, & \text{if } s_i = \text{C} \\ 3, & \text{if } s_i = \text{G} \\ 4, & \text{if } s_i = \text{T} \end{cases} \tag{1.5}$$

If $score_S \geq t$ then S is accepted as a binding site. The non-trivial part is to define a meaningful threshold score. A conceptually similar thing can be done with a LPM **M**. It is possible to compute the probability of observing the data given the model, the likelihood $p(S|M)$:

$$p(S|M) = \prod_{i=1}^{L} M_{i,b}$$
$$\text{with } b = \begin{cases} 1, & \text{if } s_i = \text{A} \\ 2, & \text{if } s_i = \text{C} \\ 3, & \text{if } s_i = \text{G} \\ 4, & \text{if } s_i = \text{T} \end{cases} \tag{1.6}$$

In turn, using Bayes'theorem, the posterior probability can be computed :

$$p(M|S) = \frac{P(S|M) \times p(M)}{\sum_i P(S|M_i) \times p(M_i)} \tag{1.7}$$

where $p(M)$ and $p(M_i)$ are model probabilities, which can be interpreted as the prevalence of a given family of sites in the case of a mixtures of binding site families.

1.6 Over-represented patterns discovery

Next generation sequencing (NGS) technologies allowed to easily characterize many of the chromatin features genome wide (see section 1.4). The wealth of such datasets rapidly required automated discovery procedures in order to extract leading trends. For instance, MNase-seq reveals the nucleosomes over a genome. Asking whether the nucleosome architecture is overall the same between two regions or how many different types of nucleosome architectures are present over a set of selected regions are quite reasonable questions. To answer them, automated discovery procedures of over-represented patterns are needed.

The *de novo* discovery of archetypical chromatin architectures, from sequencing read densities,

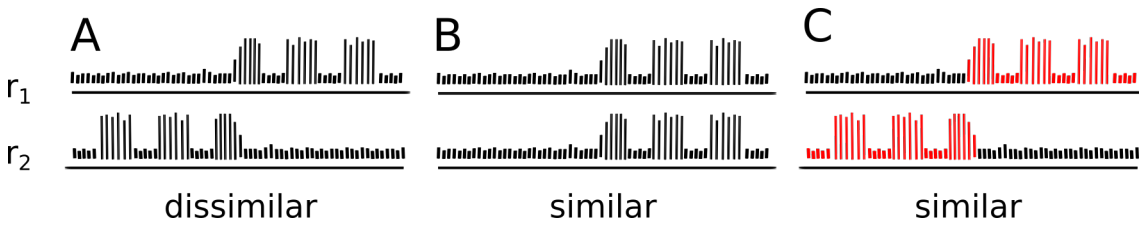


Figure 1.6 – Signal comparison between two regions r_1 and r_2 bearing a nucleosome array, measured by MNase-seq, at their ending and beginning edges. The bars indicate the density of sequencing reads stacked at each position. **A** The regions are compared position-wise, as they are, which results in a strong dissimilarity as the arrays are not aligned. **B** Performing the comparison after flipping r_2 result in both regions being highly similar. **C** A shorter stretch of signal of length $L' < L$ from r_1, highlighted in red and corresponding to the array, is searched in r_2. In this case, because a highly ressembling strech of signal, in red, can be found in r_2, both regions are considered highly similar.

over a set of regions of interest is a long standing problem in bioinformatics and is related to the problem of binding site alignment discussed in section 1.5.1. In summary, the objective is to find common sequencing signals - for instance nucleosome arrays from MNase-seq data - located in different regions, to realign them and aggregate them.

More formerly, given a matrix R of dimensions NxL containing N vectors of read counts $r_1, r_2, ..., r_N$ of length L, each containing the number of reads mapping at a given position in a given region, find $K \leq N$ vectors of length $L' = L$ that contain archetypical signals found in the N regions of R. This can actually be solved using clustering methods which group regions that look alike into K groups. Partitioning methods compare pairs of regions in order to state whether they resemble each other or not and break down R in K groups. The summary of the signal inside each group - for instance the mean signal for the K-means algorithm - can then be interpreted as the archetypical chromatin architectures. Biologically, these different organizations may reflect different functions associated with the regions of interest.

However, this can only be done properly if the positions in two regions are functionally similar. Indeed, a second issue is entangled with this heterogeneity problem : an alignment problem. Two stretches of signal, in two different regions, may be similar but located at different positions in the region. For instance two regions can be covered at 50% by a nucleosome array, in one case over the first half, in the second case, over the second half (Figure 1.6A). In this case, detecting that both regions bear an array can be done in two ways : i) flipping one region such that both arrays are localized at the same edge (Figure 1.6B) or ii) by searching archetypical signals of length $L' \leq L$ by scanning the entire regions (Figure 1.6C).

This alignment problem has several roots. Let us assume that we measured nucleosome occupancy around TF binding sites. The nucleosome occupancy signal may not be aligned from one region to the next for at least three non-exclusive reasons. First, there may be an error in the estimation of the position of the TF binding site. For instance, ChIP-seq estimates

binding +/-100bp. Assuming that a TF binding in the center of this region or where the signal is the highest do not guarantee to properly estimate the binding site. Secondly, the chromatin features can appear at a varying distance from the reference. For instance, two binding sites may be located at different distances from the closest nucleosome. Finally, the regions can show a functional orientation. For instance, TF binding sites have an upstream and a downstream. The same is true for TSSs.

Finally, if both regions are aligned, it is worth mentioning that the signal over one region may be sparser because of a sub-optimal sequencing depth that has nothing to do with biology.

The study of signal distribution over genomic regions has been a quite active field for sequencing experiments during the last decade. Dedicated algorithms and software have been developed to discover chromatin patterns, such as ChromaSig (Hon et al., 2008), ArchAlign (Lai and Buck, 2010), CATCHProfiles (Nielsen et al., 2012), CAGT (Kundaje et al., 2012) and ChIPPartitioning(Nair et al., 2014). However, individual programs do not always handle all aspects of heterogeneity. For instance, ArchaAlign only realign the data while CAGT can only handle orientations issues but does not realign the patterns.

1.7 General aims of this work

This work has been constructed around different projects that all had in common to study the binding sites of TFs. All conducted research followed a general aim that was devised as understanding TF binding beyond their sequence specificity only.

TFs are especially impacted by some properties of the chromatin fibers, such as their accessibility. However, upon binding, TFs become part of the chromatin and modify the chromatin properties. Thus trying to characterize this cross-talk was of interest.

2 Laboratory resources

The Computational Cancer Genomics (CCG) laboratory developed and maintains in-house two important databases.

The Mass Genome Annotation (MGA) repository is the first database. It contains a collection of publicly available NGS datasets. Because publishing nowadays in a peer-reviewed journal requires to release the data, the authors have to deposit them to primary repository, such as GEO (Barrett et al., 2013) or ArrayExpress (Athar et al., 2019). As a matter of fact, tremendous amounts of data are released. As part of an effort to enhance reproducibility and re-usability of these data, the CCG laboratory developed and maintains the MGA repository. The MGA repository has been designed to store publicly available NGS data, in a highly standardized manner with a high quality data annotation (metadata).

The second database is the Eukaryotic Promoter Database (EPD). EPD is an old resource containing a catalog of curated eukaryotic RNAPII TSSs. EPD was initiated by Bucher and Trifonov from the manual curration of experimental data (Bucher and Trifonov, 1986). Since its beginning, EPD was designed as a sequence annotation that indicates, for a given sequence, which positions are used to initiate transcription. With the advent of high throughput transcription profiling assays - such as CAGE and GRO-seq - TSS identification and mapping has drastically changed. In consequence, EPDnew - a new dedicated branch of EPD - was created several years ago (Dreos et al., 2013) to make full use of these new data.

2.1 Mass Genome Annotation repository

This section describes the organization and the content of the MGA repository. The MGA has been described in (Dreos et al., 2018). This work was mostly undertaken by René Dreos, a postdoctoral fellow of the CCG laboratory. My involvement in this project was related to the processing, curration and annotation of some datasets, such as the zinc finger ChIP-seq dataset released by Imbeault and colleagues (Imbeault et al., 2017).

The content of this section has been taken and adapted, with the author permissions, from

(Dreos et al., 2018).

2.1.1 MGA content and organization

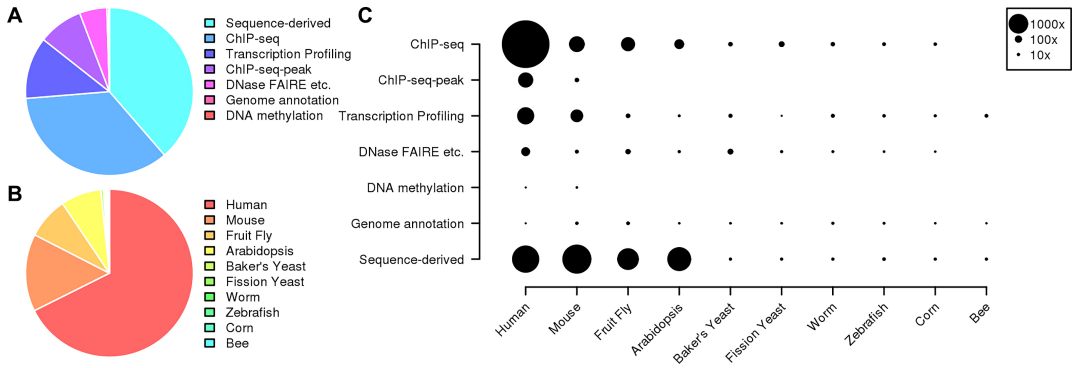


Figure 2.1 – Content of the MGA repository by 2018 A Proportion of samples in the database grouped by type. **B** Proportion of samples grouped by organism. Assemblies belonging to the same organism are merged together. **C** Samples numbers stratified by type and organism. Dot areas are proportional to the total number of samples in that category. The corresponding numbers can be found in a weakly updated table posted on the MGA home page at http://ccg.vital-it.ch/mga. Figure and legend taken and adapted from (Dreos et al., 2018).

Currently, the MGA contains more than 24'000 samples in 15 different species : human, mouse, rat, macaque, dog, chicken, zebrafish, worm, fruit fly, bee, Arabidopsis, corn, Plasmodium, baker's yeast and fission yeast. In all species, except in human, mouse, fruit fly and worm the data are mapped to a single genome assembly, called primary assembly. Among the hosted samples, landmark datasets such as the ENCODE (Consortium, 2012), RoadMap (Roadmap Epigenomics Consortium et al., 2015) or Fantom5 (Lizio et al., 2015) datasets are present. Each sample in the MGA belongs to one of the 13 mandatory data categories :

1. ChIP-seq : raw data (reads mapping coordinates) from classical ChIP-seq experiments targeting transcription factors, protein-DNA intraction, histone variants and modifications, etc.

2. ChIP-seq-invitro: raw data (reads mapping coordinates) from in-vitro ChIP-seq experiments such ad DAP-seq.

3. ChIP-seq-peak: peak regions provided by the authors of the data.

4. Transcript Profiling: raw data from experiments aimed at profiling transcripts initiation such as CAGE, GRO-cap, GRO-seq, PEAT, etc.

5. DNase FAIRE etc.: raw data from chromatin and chromatin accessibility studies such as MNase-seq, DNase-seq, DNase-hypersensitivity, etc.

6. DNA methylation: raw data from methylation studies.

7. Genome Annotation: transcription start sites, transcription end sites, intron-exon boundaries.

8. Sequence derived: PWM matches, Natural Variants, Conservation scores from PhastCons (Siepel et al., 2005) and PhyloP (Pollard et al., 2010), etc.

All the data available on the MGA are stored in Simple Genome Annotation (SGA, Ambrosini et al. (2016a)). SGA is a single coordinate format. In essence, all data are represented as a single coordinate along the genome. ChIP-seq peaks and paired-end sequenced fragments are represented by their middle position, single-end sequenced reads by their 5' end, TSSs are single base coordinates anyway, and so one. Additionally, this minimizes the disk space required to store these data. However, in any case, SGA formatted data can easily be converted to BED or to GFF using dedicated conversion tools (Ambrosini et al., 2016a).

In order to enhance original results reproducibility as much as possible, read alignment files in bed or bam format, if available on the primary repositories, are always preferred. Otherwise, the raw sequencing data are downloaded and processed using a general pipeline comprising i) read mapping using Bowtie (Langmead et al., 2009) or Bowtie 2 (Langmead and Salzberg, 2012), ii) conversion to BED using the SAMTools (Li et al., 2009) and BEDTools (Quinlan and Hall, 2010) suits and a final conversion to SGA using ChIP-seq server conversion tools Ambrosini et al. (2016a). As in GEO, data (called samples) for a given study/article belong to a same serie. Finally, the metadata are created. A full description of the data, their biological significance, their processing is available in HTML format. Two additional machine readable text files are available with i) the sample information ii) the serie information.

Importantly, it should be highlighted that the MGA is fully interconnected with in-house developed analysis tools hosted on the ChIP-seq (Ambrosini et al., 2016a) and Signal Analysis Search (SSA, Ambrosini et al. (2003)) servers. These servers contains tool to perform peak-calling, correlation analyses, sequence analysis, format conversions and much more. All the data hosted can thus be readily analyzed using any of these in-house developed tools.

2.1.2 Conclusions

The MGA repository is an important asset to the scientific community. It allows anybody to undertake quickly a wide range of data analyses, together with the ChIP-seq and SSA servers. Additionally, all these data are readily available and can be downloaded in MGA or BED format. Furthermore, on demand visualization tracks can be created for all the datasets hosted on the MGA. These tracks can then easily be uploaded to UCSC genome browser. Finally the MGA is so convenient that I used MGA hosted data in the projects described in the chapters 3, 5, 6 and 4.

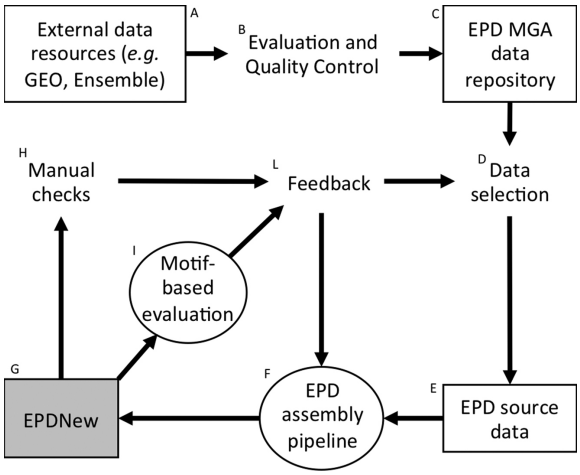


Figure 2.2 – Schematic representation of the EPDnew pipeline A Download of authoritative gene catalogs and primary TSS mapping data from public databases, data repositories and consortium websites. **B** Quality control (QC) of incoming data (e.g. read mapping efficiency, contaminations, etc.). **C** Data passing QC are reformatted and incorporated into the MGA repository. **D** Selection of a subset of TSS mapping experiments for generating a new organism-specific TSS collection. **E** Input data for a new module of EPDnew. **F** Organism-specific automatic database assembly pipeline tailored to the input data, see (Dreos et al., 2013) for a detailed description of the human EPDnew assembly pipeline. **G** Preliminary or final TSS collection **H** Manual sanity checks of individual randomly selected promoter entries using the corresponding entry viewer. **I** Automatic quality evaluation of the TSS collections as a whole by motif enrichment tests, see Figure 2.3 for an example. **L** Feedback is collected from quality evaluation steps H and I. This may lead to the exclusion, replacement or addition of source data sets or modifications (e.g. program parameter fine-tuning) of the computational database generation pipeline. Note that the development of a final, publicly released EPDnew module typically involves several evaluation-modification cycles. Figure and legend taken and adapted from (Dreos et al., 2017).

2.2 Eukaryotic Promoter Database

This section recapitulates some of the results published in (Dreos et al., 2017) and in (Meylan et al., 2020). Most of the work presented in this section should be credited to René Dreos and Patrick Meylan, two former post-doctoral fellows of the CCG laboratory.

In essence, each EPDnew release is created using a semi-automated computational pipeline that identifies genomic regions showin high mRNA initiations at the beginning of annotated genes. The input data are subjected to a severe quality control checks before entering the EPDnew pipeline. Additionally, the results are manually verified throughout the entire process, ensuring high curration standards. The entire pipeline is depicted in Figure 2.2.

EPDnew database is dedicated to provide an accurate TSS mapping to the research community.

Organism, release	Promoters, genes with TSS, genes	TSS libraries
H. sapiens (6)	29598, 16455, 17056 (96%)	1311
H. sapiens nc (6)	2339, 1894, 3496 (54%)	1311
M. mulata (1)	9575, 9026, 20593 (44%)	15
M. musculus (3)	25111, 20213, 24864 (81%)	965
M. musculus nc (3)	3077, 2938, 10184 (29%)	965
R. norvegicus (1)	12601, 12013, 21919 (55%)	13
G. gallus (1)	6127, 5632, 16837 (33%)	32
C. familiaris (1)	7545, 7321, 19971 (37%)	12
D. melanogaster (5)	16972, 13399, 13660 (98%)	375
A. melifera (1)	6493, 5712, 10727 (53%)	16
D. rerio (1)	10728, 10235, 18606 (55%)	21
C. elegans (1)	7120, 6363, 11786 (54%)	9
A. thaliana (4)	22703, 22701, 27149 (83%)	13
Z. mays (1)	17081, 15828, 26651 (59%)	8
S. cervisiae (2)	5117, 5117, 5819 (88%)	22
S. pombe (2)	4802, 4801, 5128 (94%)	16
P. falciparum (1)	5597, 4028, 4994 (80%)	12

Table 2.1 – Current contents of EPDnew 'Promoters' indicate the number of TSS entries in EPDnew. 'Genes' indicates the number of genes having at least one TSS annotated in EPDnew. 'Genes' indicates the number of protein coding genes contained in the genome annotation (except for nc species). 'nc' stands for non-coding and indicates the long non-coding gene annotations. For 'nc' entries, 'genes' refers to the number of long non-coding genes present in the annotation. In parenthesis are indicated the percentages of genes having a at least one TSS annotated in EPDnew.

EPDnew was firstly focused on the annotation of animal genomes Dreos et al. (2015). However, with the increasing availability and origins of relevant datasets, the database could be extended to least common - often neglected - species. EPDnew currently includes plant and fungi species. Even if not fully in line with the rest of this work, these species are fully part of EPDnew and should be presented as such.

EPDnew contains computational annotations of genome assemblies from publicly available high throughput 5' mRNA sequencing data. The following sections contain a description of the current state of EPD and of the recent novelties introduced.

The content of this section was taken and adapted, with the author permissions, from Dreos et al. (2017) and Meylan et al. (2020).

2.2.1 EPDnew now annotates (some of) your mushrooms and vegetables

With years, EPDnew has substantially grown, from a promoter collection annotating protein coding genes in five animal model organisms (human, mouse, fruit fly, zebrafish and worm,

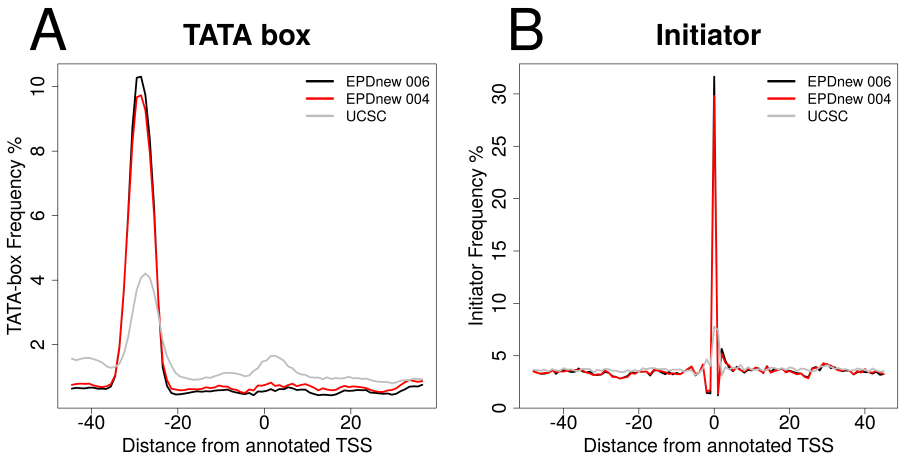


Figure 2.3 – TSS Mapping precision Occurrence of the TATA-box **A** and initiator **B** around *H.sapiens* TSSs from EPDnew releases (004 and 006) and from a list of gene starts from UCSC Gene list, which was used as input for the generation of the EPDnew collection. This figure was created using Oprof from the SSA server (Ambrosini et al., 2003). Detailed instructions to recreate the figure can be found in section 2.2.5.

Dreos et al. (2015)), to 10 (human, mouse, fruit fly, zebrafish, worm, bee, arabidopsis, maize, brewer's yeast and fission yeast, Dreos et al. (2017)) and now 16 organisms annotating protein coding and non coding genes (Table 2.1). The number of genes containing at least one annotated TSS in EPDnew is variable among species. However *H. sapiens*, *D. melanogaster* and *S. pombe* are approaching a complete gene coverage of protein coding genes with 96%, 98% and 94%.

2.2.2 Increased mapping precision in human

The human annotation has been generated with >1300 experiments, containing dozens of billions of reads. It is currently and by far the largest data collection among EPDnew. Importantly, even if the number of TSSs reported increased compared to what has been reported in (Dreos et al., 2017) (25 503 using 1088 datasets vs 29 598 using 1311 datasets) the gene coverage is reaching saturation (95% vs 96%). Thus most of the newly discovered TSSs are alternative TSSs. Nonetheless, the overall TSS mapping precision is still increasing, as shown in Figure 2.3. This is illustrated using the positional distributions of the TATA-box and the initiator motifs which are both core promoter elements that are expected to appear at a fixed distance from the TSS. The increased frequencies seen in EPDnew release 006 compared the other TSS annotations indicate a better alignment of the TSSs.

2.2.3 Integration of EPDnew with other resources

EPDnew is also hosted in the MGA repository (see section 2.1 and (Dreos et al., 2018)). As such, EPDnew TSSs are available together with the ChIP-seq (TF binding, histone marks), the nucleosome occupancy, chromatin accessibility, SNP and sequence conservation data present. As a consequence, EPDnew can easily be integrated into diverse genomic analyses through the tools hosted on the ChIP-seq (Ambrosini et al., 2016a) and SSA (Ambrosini et al., 2003) servers.

Besides, EPD could be explored through a viewer relying on a selection of tracks to be visualized in the UCSC Genome Browser (Dreos et al., 2013). Since then, a major effort to provide a customizable visualization plateform has been undertaken (Dreos et al., 2017). Currently, each specie is provided with a minimal track hub (Raney et al., 2014) containing at least 3 tracks : i) the combined TSS mapping samples used to create the EPDnew annotation for this specie, ii) the EPDnew TSSs on the + strand and iii) the EPDnew TSSs on the - strand. Other tracks are often available depending on the specie such as a gene track, a CpG island track and so one. Finally, a tool to create and upload custom MGA derived tracks (ChIP-Track available at https://ccg.epfl.ch/chipseq/chip_track.php) on the UCSC Genome Browser, in a few mouse clicks, have been developed to fully exploit the possible synergies between the UCSC Genome Browser, EPDnew and the MGA repository.

2.2.4 Conclusions

EPDnew is a valuable resource for the research community. It offers an unprecedented TSS mapping effort in terms of precision and species covered. To enhance and facilitate the integrative analyses, EPD is fully interconnected with the external resources and tools from the ChIP-seq (Ambrosini et al., 2016a) and SSA (Ambrosini et al., 2003) servers maintained by the laboratory. Currently, EPDnew tracks are available on UCSC Genome Browser. Furthermore, EPDnew is totally interconnected with the MGA which allows to easily integrate it in different genomic analyses. Additionally, it is possible to create "on demand" visualization tracks for any dataset hosted on the MGA to complement the tracks already available on UCSC Genome Browser.

Finally, it should be noted that EPDnew genome annotations have been used in the projects described in chapters 3 and 4.

2.2.5 Methods

Motif occurrence profiles

The motif occurrence profiles in Figure 2.3 have been generated using Oprof from the SSA server https://ccg.epfl.ch/ssa/oprof.php. To create the TATA-box profiles, the input data were H. sapiens (Feb 2009 GRCh37/hg19) / Genome Annotation / EPDnew / release 004 or 006 or UCSC, TSS for known Genes / TSS from UCSC known genes. The motifs were choosen from

PWMs from libraries / Promoter Motifs and then TATA-box (length=15) or Initiator (length=8). The borders were set to -50 to +50bp, the window size to 20bp for the TATA-box and 10bp for the Initiator and forward search mode was enabled. All other parameters were left to default.

3 ENCODE peaks analysis

As discussed in Chapter 1, the structure of the chromatin has a deep impact on TF binding. It is now clear that nucleosome occupancy fulfills more than a packaging role. It can also acts as a barrier to impede DNA reading processes and compete with TFs for sequence occupancy. Thus gaining a better understanding of how chromatin is organized around TF binding sites is crucial to understand TF binding beyond their sequence specificity only.

In an effort to better understand how the genome is organized and how its functions are fulfilled, the ENCODE Consortium (Consortium, 2012) released an impressive collection of coherent data representing an unprecedented picture of the chromatin in several human cell lines.

The joint analysis of TF occupancy data -measured by ChIP-seq - with nucleosome occupancy data - generated by MNase-seq - and chromatin accessibility data - generated by DNase-seq - showed that nucleosomes and TFs seem to compete for TFBS as they are nucleosome depleted but are predicted to be occupied, whereas flanking sequences are occupied by 4 positioned nucleosomes, in agreement with predictions (Wang et al., 2012; Kundaje et al., 2012). These observations were also independently reported (Gaffney et al., 2012). Additionally, the analysis of the ENCODE data revealed several interesting things : i) it was confirmed that most TFs bind genome-wide to chromatin accessible regions (Neph et al., 2012), ii) TF occupancy generates characterisitic DNaseI digestion profiles which closely match TF recognition motifs, with a single bp resolution, at single TFBS, probably reflecting the TF-DNA interface (Neph et al., 2012) and iii) TFBS were shown to be flanked by well positioned nucleosomes and to have asymetric nucleosome occupancy on each sides (Kundaje et al., 2012). Together i) and ii) shown that TF are, as expected, occupying open chromatin regions and that DNaseI is a good proxy for TF occupancy. Besides, iii) suggested that TFBS could act as anchor around which nucleosomes can organize.

To beginning with, the main starting questions I addressed were to verify these current views, namely : i) whether TF binding and nucleosome are refractory to one other and ii) to explore how nucleosomes are organized around TF binding sites. The latter question also implied to

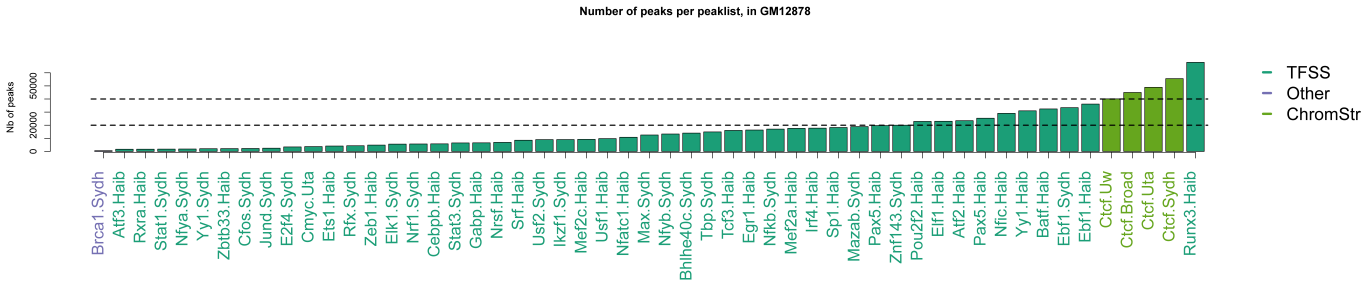


Figure 3.1 – Number of peaks in GM12878 called by ENCODE for each ChIP-seq experiment. The different TFs are colored by type, as defined by (Cheng et al., 2012) : sequence specific TF (TFSS), chromatin structure (ChromStr) and others. The horizontal dashed lines indicate 20'000 and 40'000 peaks respectively. The datasets are named using the TF and the laboratory which produced the data.

figure out whether there exists several different types of nucleosome organizations.

The GM12878 cell line was retained by the ENCODE Consortium, as one of the highest priority cell line. In consequence, their genome has been sequenced and many assays, including TF ChIP-seq, DNase-seq and MNase-seq, were performed on them. Interestingly, GM12878 retained the ability to divide but show a normal karyotype - unlike HeLa cells. All together, these features make of GM12878 cells a good model for genomic studies. For this, I decided selected these data to conduct genomic studies.

3.1 Data

During its production phase in 2012, the ENCODE Consortium released ChIP-seq data for 53 different TFs, nucleosome occupancy data (MNase-seq, Kundaje et al. (2012)) and chromatin accessiblity data (DNase-seq, (Thurman et al., 2012)) that were generated with a high depth of coverage in GM12878 cells. The ENCODE Consortium also released ChIP-seq peaks called using a uniform processing pipeline (Gerstein et al., 2012). These peaks account for i) technical variability as they are called from technical replicates and ii) inter peak caller discrepancies as several peak callers results were integrated together as part of the peak calling pipeline. These peaks are thus reproducible and robust to software related biases and can be considered as an excellent standard.

All data were taken from the MGA repository. The ChIP-seq peaks can be found at https://ccg.epfl.ch/mga/hg19/encode/Uniform-TFBS/Uniform-TFBS.html, the MNase-seq data at https://ccg.epfl.ch/mga/hg19/encode/GSE35586/GSE35586.html and the DNase-seq at https://ccg.epfl.ch/mga/hg19/encode/UW-DNaseI-HS/UW-DNaseI-HS.html.

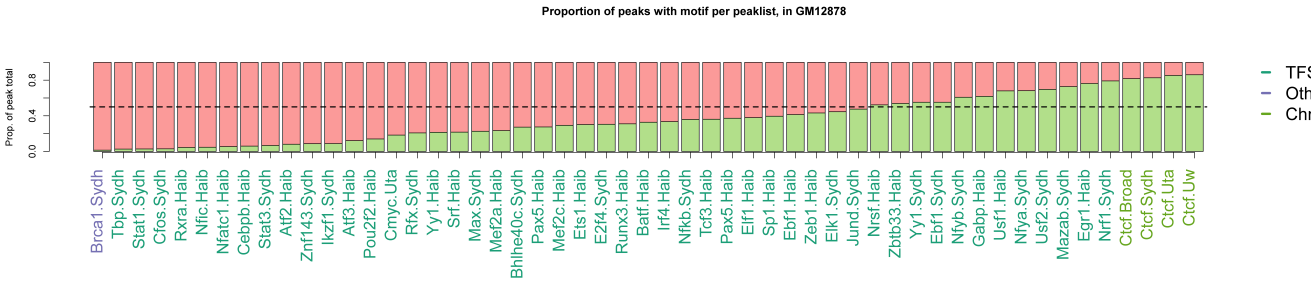


Figure 3.2 – Proportion of peaks with a motif occurrence in GM12878, for each ChIP-seq experiment, in green. Assuming that a TF binds to DNA through its motif, a motif occurrence should be nearby the peak center. Thus the center of each peak was scanned using a PWM modeling the TF binding specificity. Each TF was associated to a log-odd PWM contained either from JASPAR Core vertebrate 2014 (Mathelier et al., 2014), HOCOMOCO v10 (Kulakovskiy et al., 2016) or Jolma (Jolma et al., 2013) collection. If a motif occurrence (with a score corresponding to a pvalue higher or equal to $1 \cdot 10^{-4}$) could be found, the peak was considered bearing a motif occurrence. The different TFs are colored by type, as defined by (Cheng et al., 2012) : sequence specific TF (TFSS), chromatin structure (ChromStr) and others. The horizontal dashed line indicates 0.5. The datasets are named using the TF and the laboratory which produced the data.

The number of peaks called for each TF was highly variable and likely reflects each factor activity in this cell line (Figure 3.1). The most abundant factor in terms of peaks was RUNX3 followed by CTCF. This observation fits to BioGPS (Wu et al., 2016) data which indicates that both RUNX3 and CTCF have a higher expression in lymphoblast and in B cells compared to other tissues. Moreover, the propensity of each TF to bind through their motifs was also variable, with again CTCF being showing the highest values (Figure 3.2).

3.2 ChIPPartitioning : an algorithm to identify chromatin architectures

As discussed in section 1.6, pattern discovery is a long standing bioinformatic problem and several algorithms have been proposed to solve it. ChIPPartitioning (Nair et al., 2014) is probably the best of them. It is a probabilistic partitioning algorithm that softly clusters a set of genomic regions based on their signal shape (as opposed to the absolute values) resemblance. To ensure proper comparisons between the regions, the algorithm allows to offset one region compare to the other to retrieve a similar signal at different offsets and to flip the signal orientation. Finally, it has been demonstrated to be really robust to sparse data.

ChIPPartitioning (a graphic representation of the algorithm can be found further below in Figure 7.1) models the signal over N region of length L as having being sampled from a mixture

of K different read density models (classes), using L independent Poisson distributions for each region. The number of reads sequenced over this region is then the result of this sampling process. Each class model is represented by a vector C_k of size $L' \leq L$ that represent the expected number of reads at each position for class k. These values are thus the Poisson distribution parameters. The number of reads $r_{i,j}$ at position j, in a region i is :

$$r_{i,j} = \sum_{k=1}^{K} p_k \times X_{i,j,k} \tag{3.1}$$

where p_k is the probability of the class k and $X_{i,j,k}$ the number of reads sampled from $Poisson(\lambda = c_{k,j})$.

In order to discover the K different class models - that are the chromatin signatures to find - in the data, the algorithm proceeds to a maximum likelihood estimation of the Poisson distribution parameters $c_1, c_2, ..., c_k$ and the class probabilities $p_1, p_2, ..., p_k$ using an expectation-maximization (EM) framework. During the E-step, the likelihood $P(r_i|c_k)$ of each region i, given each class k and a posterior probability $P(c_k|r_i)$ are computed. The posterior probabilities are interpreted as the probability that r_i belongs to class k. Eventually, during the M-step, the class models $c_1, c_2, ..., c_k$ are updated using :

$$c_{k,j} = \sum_{i=1}^{N} p_k \times r_{i,j} \tag{3.2}$$

This procedure is actually a weighted and ungaped data alignment in which the posterior probabilities are the weights with the class models containing the average number of reads at each position of the alignment.

Since each region is assigned a probability of belonging to each class, it participates to the update of all the class models, with different weights.

If the length of the chromatin signature searched $L' < L$, then the algorithm slides a window of length $L' = L - S + 1$ along the regions, at each possible offset $1, 2, ..., S$, and searched for this signature at each possible offset. This is how it deals with alignment issue. The signal orientation issue is tackled by also performing a searched with the flipped model.

At the end of the process, this algorithm returns a posterior probability matrix of dimensions $NxKxSx2$ with $S = L - L' + 1$ corresponding to regions, classes, shift states (S) and flip states (forward and reverse). In other words, ChIPPartitioning computes a probability of belonging to each class, to each window (in both orientation) in each region.

Because the estimation of the class model parameters is done using an EM framework, ChIP-Partitioning is a heuristic algorithm. The final parameter estimates depend on the starting state (which is set randomly) and on the number of iterations run. Finally, it is worth mention-

ing that this algorithm is close to the MEME algorithm (Bailey and Elkan, 1994) that models DNA sequences as being sampled from a two class mixture model that represents the DNA motif to find and the noise.

Regarding implementation details, ChIPPartitioning was implemented in R programming language, as it was proposed in the supplemental material of (Nair et al., 2014). Nonetheless, ChIPPartitioning turned out to be relatively slow due to the quite heavy computations it has to carry out (logarithms, exponentials and probability computations), the intrinsic limitations of the R programming language and the lack of optimization in the implementation.

3.2.1 Data realignment

ChIPPartitioning computes a set of posterior probabilities and uses them to perform the class model updates. As illustrated in Figure 7.1, this procedure is actually a weighted and ungaped data alignment in which the posterior probabilities are the weights.

It is absolutely feasible to run a partitioning on a given matrix A, for instance MNase-seq read counts, using ChIPPartitioning, and to subsequently use the obtained posterior probabilities to compute the class models, using another data matrix, let us say B of DNase-seq reads.

This procedure allows to realign a dataset B as A in order to co-visualize different types of signals. The only things that should be taken care of is that matrices A and B should have the same dimensions and that the genomic positions inside both matrices are strictly identical.

In the following sections, this is the procedure that will be used to overlay different types of data for a given partition.

3.3 Nucleosome organization around transcription factor binding sites

For each dataset, the peak coordinates were reassigned to the best TF motif occurrence, if any in the peak. However dealing with unaligned signal was still necessary. Indeed, it could not be excluded that the differents TFs would not be the anchor of the chromatin organization around them and have nucleosome arrays at variable distances from their binding sites. Furthermore, dealing with the region orientation was also needed because i) all peaks did not contain a motif occurrence indicating the directionality of the binding site (Figure 3.2) and ii) as before, the TF binding site may not be the main driving force of the neighboring chromatin organization. However, this pre-processing step, even if it could not resolve entirely this issue, could at least soften it.

To uncover the different nucleosome architectures around TF binding sites, one partition per peaklist based on the MNase-seq signal was performed using ChIPPartitioning. Because the time required to run the partitioning procedure is long and is a linear function of the

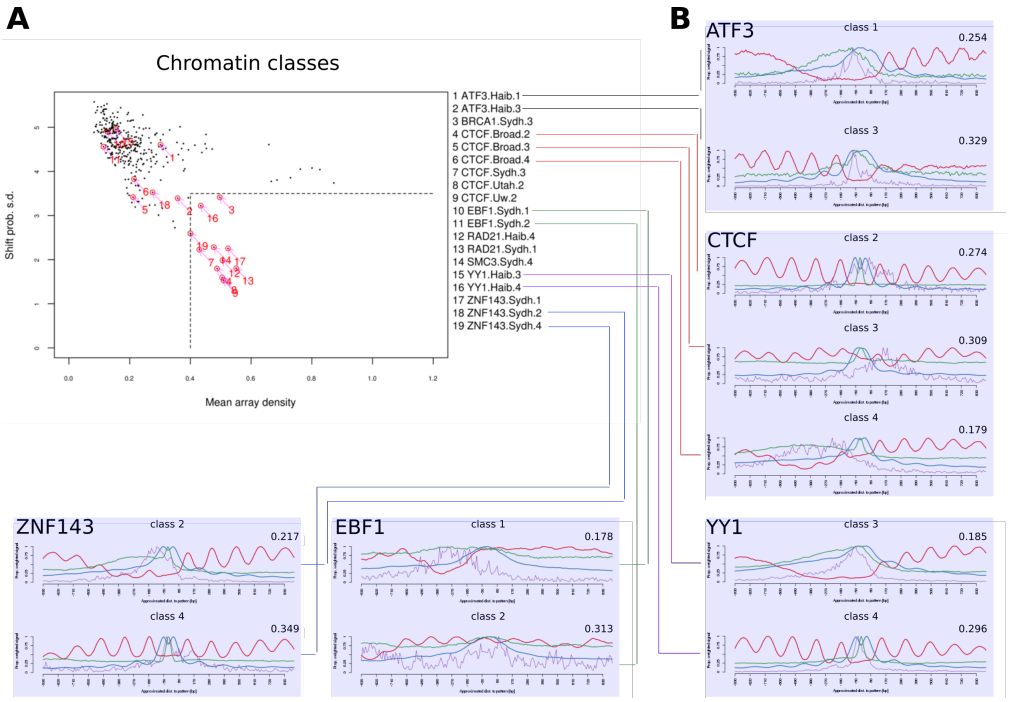


Figure 3.3 – Chromatin pattern around TF binding sites in GM12878 : A For each peaklist, nucleosome occupancy was measured +/- 1kb around each individual TF binding site using 10bp bins. The TF binding site were then classified into 4 classes according to their nucleosome patterns using ChIPPartitioning, allowing the patterns to be flipped and shifted. Each TF binding site was assigned a probability to belong to each of the 4 classes with a given values of shift and flip. To assess the extent of a given TF to i) display nucleosomes arrays on its flank and ii) to have nucleosome positioned with respect to its binding sites, array density and shift probability standard deviation have been measured for each class. Classes having a mean array density above 0.4 and a shift probability standard deviation under 3.5 and other custom classes are highlighted. Classes are named using the TF, the laboratory which produced the data and the class number (from 1 to 4). **B** Examples of class patterns corresponding to some of the highlighted classes for CTCF, ATF3, YY1, EBF1 and ZNF143. MNase profiles (red) were allowed to be shifted and flipped and DNaseI (blue), TSS density (violet) and sequence conservation (green) were overlaid according to MNase classification (taking into account both shift and flip). The y-axis scale represents the proportion of the highest signal for each chromatin pattern.

number of classes, the choice of four classes was a compromise allowing to discover several chromatin architectures while not being computationally to intense. ChIPPartitioning was also given a freedom of shifting of 15 bins (corresponding to -70bp, -60bp, ..., 0bp, ..., +60bp, +70bp) and of flipping. A visual inspection of the results revealed that all classes, for all TFs, show a nucleosome array on at least one of the side of the TF binding site (examples are displayed in Figures A.1, A.2, A.3 and A.4). Additionally, it was also possible to see an increased chromatin accessibility and sequence conservation at the level of the binding site. The enhanced chromatin accessibility is compatible with the current view of TFs binding nucleosome depleted regions (Kundaje et al., 2012). However, the absence of a footprint like signal is explained by the shifting. By shifting and flipping the regions, ChIPPartitioning realigns the signal over these regions, at the cost of unphasing the binding sites.

A noticeable exception to this rule was the early Early B-cell factor 1 (EBF1) that seemed to have nucleosome arrays spanning its binding sites (Figure 3.3B).

In order to explore more carefully to what extent nucleosome arrays may be organized with respect to each TF binding sites, I used the mean array density measure developed by (Zhang et al., 2014). A class pattern showing well positioned nucleosomes is typically showing sharp regions of strong signal separated by signal depleted regions reflecting of the alternance of nucleosome presence/absence. The method developed by Zhang and colleagues basically searches for strong variations of signal. The higher the score, the most the pattern contains well positioned nucleosomes. On the other hand, the ability of a TF to act as an anchor for arrays organization was measured as the standard deviation of the shift used by ChIPPartitioning. Briefly, it is possible to compute the probability density of the usage of each shift state. Assessing how much the different shift states were used is indicative of how much the individual patterns were aligned at the beginning. A low standard deviation value indicates that the shifting tends to be the same for all binding sites and thus that the nucleosome arrays occur at a fixed - unspecified - distance from the binding site. In this case, the binding site could be the array anchor.

Both values were measured for all classes discovered, for all TFs. The results are displayed in Figure 3.3. First, it was possible to identify a sub-population of classes in which the TF binding site seemed to act as an anchor for the nucleosomes. This represented binding sites for CTCF, RAD21, SMC3, YY1 and ZNF143 (see Figure 3.3A, points 6,8,10,13,14,15,18 and 19). A closer inspection of these class patterns showed a strong DNaseI footprint and a peak of sequence conservation. A DNaseI footprint is a typical pattern - composed of a signal depletion in between two signal enriched regions - revealing a region protected against the action of DNaseI by the binding of a factor. The presence of a clear footprint indicated that the underlying binding sites were aligned, supporting the fact that the binding sites were anchors for the nucleosome organization. This was further supported by the sharp peak of sequence conservation indicating, most likely, the TF motif. All other classes showed a wide and fuzzy chromatin accessibility pattern, as illustrated by ATF3 in Figure3.3B, indicating miss-aligned binding sites.

Breast cancer type 1 susceptibility protein (BRCA1) was also identified using this method. The identified class (class 3, see Figure A.5) indeed showed well positioned nucleosomes. However, I decided not to consider this hit for two reasons : i) there was no footprint in the nucleosome depleted region indicating that the sites are not aligned and ii) the ENCODE consortium labeled this peak list as problematic (low reproducibility read coverage).

Finally, it should be noted that noisy MNase-seq patterns were attributed high nucleosome array density scores. Because the nucleosome signal is noisy, it varies a lot and gets a good score. Such classes were found in the cloud of points just above the horizontal line on the right of the plot (mostly RNAPIII peak classes). Second, some CTCF binding sites displayed strongly positioned nucleosome, confirming previous reports (Kundaje et al., 2012; Fu et al., 2008).

Thus even if all classes showed at least one nucleosome array, it seems that most of the TFs are not the force driving the array organization, with the noticeable exceptions of CTCF, RAD21, SMC3, YY1 and ZNF143.

3.4 The case of CTCF, RAD21, SMC3, YY1 and ZNF143

Two possible alternative hypotheses could explain the presence of these strong nucleosome arrays around these TFs binding sites. First, each TF has the ability to drive the formation of well spaced nucleosome arrays in their vicinity. Second, all the classes detected contains the same set of genomic regions.

Two obsevations strongly support the second hypothesis. First CTCF is known to interact with the cohesin complex (Stedman et al., 2008) - composed of SMC1, SMC3, RAD21 and either STAG1 or STAG2 (Losada, 2014) -, with YY1 (Donohoe et al., 2007) and with ZNF143 (Bailey et al., 2015). Second, the YY1 and ZNF143 showed ~50% and ~10% of direct binding respectively (Figure 3.2), leaving the possibility of an indirect binding mechanism, for instance through CTCF.

To further confirm this hypothesis, I measured the extent to which CTCF and the other TF peaks co-localized. To do so, each RAD21, SMC1, YY1 and ZNF143 peak was checked for the presence of a CTCF peak. The results, shown in Figure3.4A, supported the four already known interactions between CTCF and the cohesin complex members RAD21 and SMC3, between CTCF and YY1 and to a lesser extent and between CTCF and ZNF143. Additionally, for YY1 and ZNF143, the presence of CTCF and of a canonical motif occurrence happen at separated peak subsets, as shown in Figure 3.4B, suggesting two different binding strategies : i) through a direct recognition of the motif or ii) through another mechanism leading to a co-localization with CTCF - most likely through binding to CTCF.

Peaks are represented by the maximum read density position, as defined by ENCODE. Thus, the effective binding site of these TF can by anywhere in the peak. As a matter of fact, ZNF143 and YY1 may bind close but without direct interaction with CTCF. If SMC3, RAD21, YY1 or

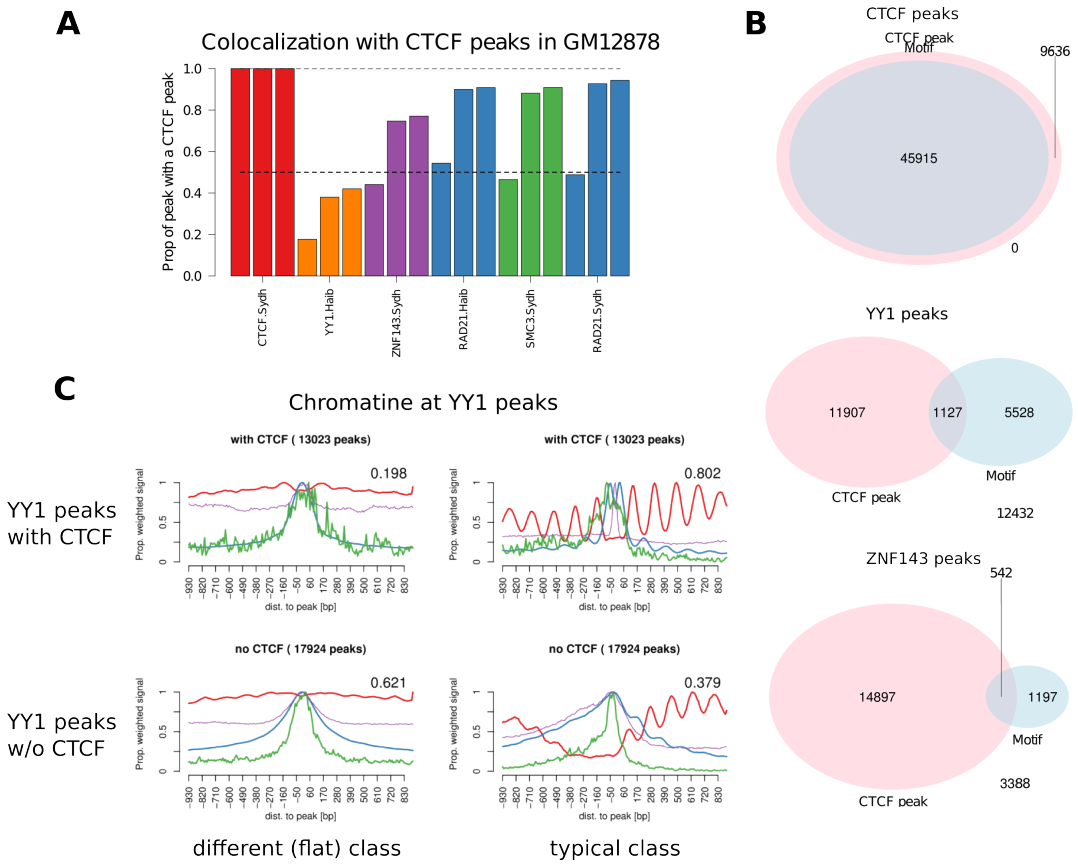


Figure 3.4 – Colocalization with CTCF peaks in GM12878 cells : **A** Proportion of peaks for different TFs having a CTCF peak within 10bp, 50bp and 100bp. The colours indicate different TFs. The CTCF peaklist used as reference to assess CTCF presence was CTCF.Sydh (in red), the two RAD21 peaklists are RAD21.Haib and RAD21.Sydh respectively (in blue), the SMC3 peaklist is SMC3.Sydh (in green), the YY1 peaklist is YY1.Haib (in orange) and the ZNF143 peaklist is ZNF143.Sydh (in violet). **B** Venn diagrams showing the proportion of peaks for each TF with i) an occurrence of its own motif, ii) a CTCF.Sydh peak within 100bp, iii) both or iv) neither of them. RAD21 and SMC3 are not represented as there is no PWM available to describe their sequence specificity. **C** ChIPPartitioning classification with shift and flip of MNase patterns +/- 1kb of YY1.Haib peaks using 10bp bins. YY1 peaks with (upper row) and without (lower row) a CTCF peak within 100bp. Two classes were used to account for "typical" and "non-typical" looking MNase patterns. DNaseI (blue), TSS density (violet) and sequence conservation (green) were overlaid according to MNase classification (taking into account both shift and flip). The number at the upper right corner of each plot indicate the overall class probability. The number of YY1 peaks is slightly smaller than in B) because peaks showing no MNase reads were not included in the classification analysis. Peaklists are named using the TF together with the laboratory which produced the data.

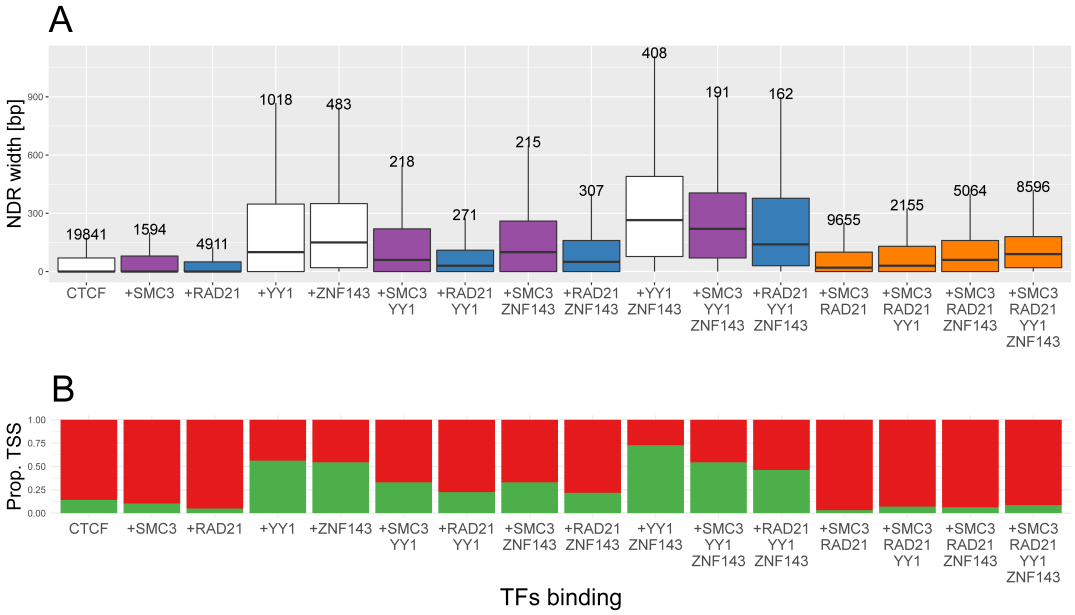


Figure 3.5 – Nucleosome free regions at CTCF binding sites A The NDR lengths are represented as boxplots. The CTCF binding sites are divided into subgroups according to additional presence of SCM3, RAD21, YY1 or ZNF143. The number of binding sites in each subgroup is indicated above the boxplots. The presence of SMC3 only, RAD21 only and SMC3 and RAD21 together are indicated in violet, blue and orange respectively. **B** The proportion of peaks (in green), in each subgroup, having a TSS within a 1kb.

ZNF143 physically interact with CTCF and bind as a complex, one prediction would be that an extended NDR should be observed to allow these complexes to bind.

In order to verify this hypothesis, I set up a classification method that assigns either a "nucleosome" or a "free" label to each position, in a given region based the MNase-seq signal. Assuming that the center of the CTCF peaks is in a NDR, these positions were labeled as 'free'. From there, the neighboring positions on the left and on the right were classified, until finding the first position labeled 'nucleosome' (see Figure A.6). The size spanned by the regions labeled as 'free' were then measured for each CTCF binding site. The NDR lengths were finally grouped according to the presence of RAD21, SCM3, YY1 or ZNF143 (Figure 3.5).

First, it seems that CTCF binding sites are distributed in two functional groups of regions based on the presence of other interactors : i) promoter distant regions with both RAD21 and SMC3 (the cohesin complex), ii) promoters together with YY1 and/or ZNF143. This segregation likely reflects different functions of CTCF : i) looping related functions with the cohesin complex and ii) a regulator of transcription with other partners. The fact that promoter enriched groups show an increased NDR, can be explained by an enhanced chromatin opening to accommodate for the presence of other TFs and of the RNAPII.

Interestingly the subgroups containing the cohesin complex (in orange in Figure 3.5A) show a NDR length that is function of the number of TFs present (cohesin < cohesin + YY1/ZNF143 < cohesin + YY1 + ZNF143). Because these sites are away from promoters, it is really likely that the increased NDR size is only caused by the binding of a larger CTCF complex. Furthermore, their reduced NDR size measured is compatible with the classes of binding sites showing strong nucleosome arrays.

Finally, in order to reveal the nucleosome organization around each subset of peaks, I performed a ChIPPartitioning classification method using two classes, with one of them set to represent a flat signal (and to act as a "waste" class). The aim was to make a clear difference between "typical" and "non-typical" nucleosome organizations. For RAD12, SMC3, YY1 and ZNF143 the results showed that strong nucleosome arrays on both sides and a clear DNaseI footprint were only present when CTCF was also present, as illustrated for YY1 in Figure 3.4C.

Together, these results support the hypothesis that CTCF forms a complex with YY1 and/or ZNF143, additionally than with the cohesin complex. They also support the fact that only CTCF has the property of positioning nucleosomes into regular arrays in its vicinity and that any other TF showing such a behaviour is likely binding with CTCF. As important, the apparent seggregation in terms of regions bounds by the different CTCF complexes is consistent with the hypothesis that the different functions of CTCF depends on its interactors (Ong and Corces, 2014; Ghirlando and Felsenfeld, 2016).

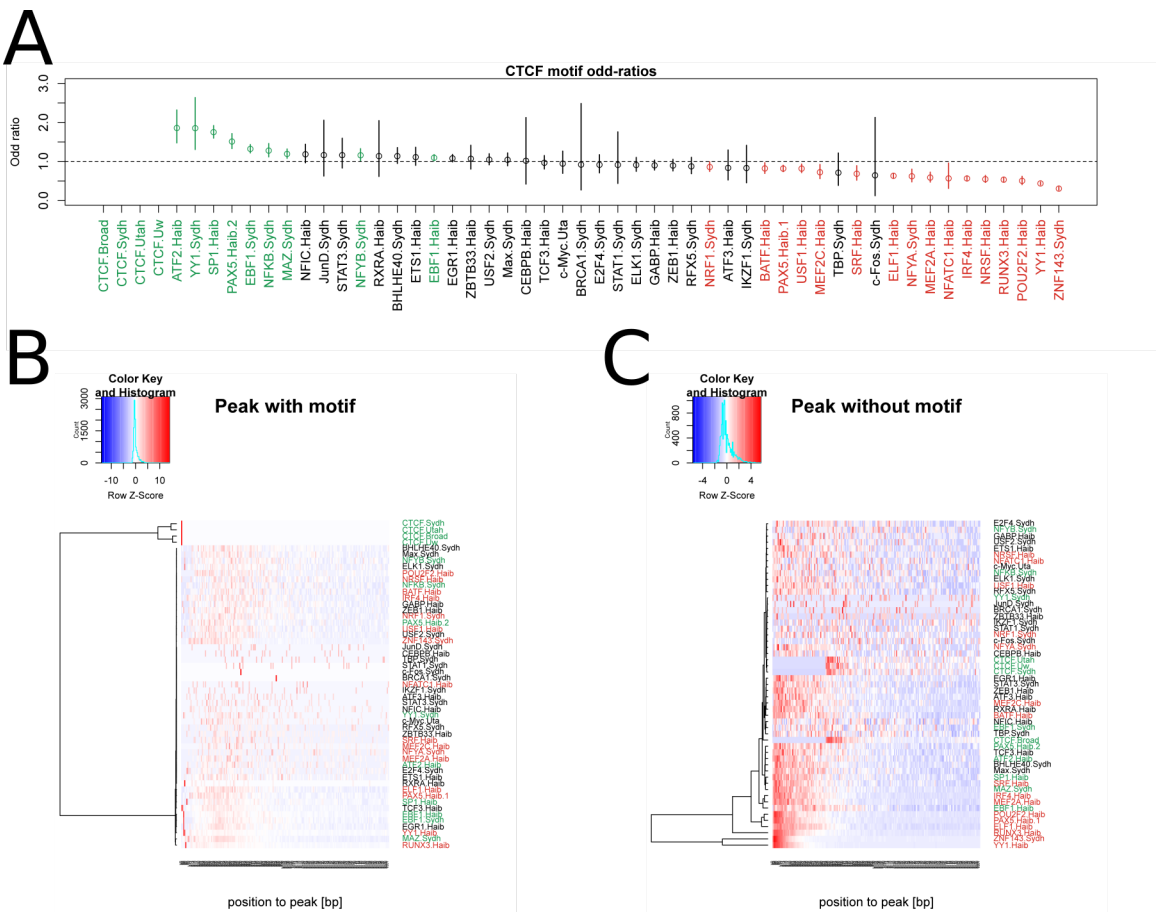


Figure 3.6 – CTCF motif association measured around the binding sites of different TFs. For each TF, its binding sites, +/- 500bp, were searched for the presence of i) the TF motif and ii) CTCF motif. For each TF, a 2x2 contingency table was created with the number of peaks having i) both motifs, ii) the TF motif only, iii) CTCF motif only and iv) no motif. **A** Odd ratio (OR) of the exact Fisher test performed on each TF contingency table. The ORs are displayed with their 95% confidence interval (CI). ORs > 1 - that is, with 1 not part of the 95%CI - are labeled in green and indicate an association of both motifs more frequent than expected by chance. ORs < 1 are labeled in red and indicate a repulsion of both motifs more frequence than expected by chance. The CTCF dataset ORs are too high to be represented in this plot. **B** Density of CTCF motif occurrences at the absolute distance of different TF binding sites (peak centers) which also have their own motif present (at distance 0). The rows were standardized and aggregated using the Euclidean distance. **C** Same as in (B) but for TF binding sites that does not have their own motif. The absence of CTCF motif occurrence within the first 70bp around CTCF binding sites is explained by the peak processing (see section 3.8.1).

Curated associations						
TF_A	TF_B	Motif ass.	Type	Binder	Reported	Validated
CTCF	ATF2	pos	indep.co-bind		no	no
CTCF	EBF1	pos	indep.co-bind		yes	no
CTCF	MAZ	pos	indep.co-bind		yes	no
CTCF	NFYb	pos	indep.co-bind		yes	no
CTCF	NFkB	pos	indep.co-bind		yes	no
CTCF	PAX5	pos	indep.co-bind		yes	no
CTCF	SP1	pos	indep.co-bind		yes	no
CTCF	BATF	neg	indir.co-bind	BATF	yes	no
CTCF	ELF1	neg	indir.co-bind	ELF1	yes	no
CTCF	IRF4	neg	indir.co-bind	CTCF	yes	no
CTCF	MEF2a	neg	indir.co-bind	both	yes	no
CTCF	MEF2c	neg	indir.co-bind	both	yes	no
CTCF	NFATc	neg	indir.co-bind	CTCF	no	no
CTCF	NFYa	neg	indir.co-bind	CTCF	yes	no
CTCF	NRF1	neg	indir.co-bind	CTCF	yes	no
CTCF	NRSF	neg	indir.co-bind	CTCF	yes	no
CTCF	PAX5	neg	indir.co-bind	both	yes	no
CTCF	POU2f	neg	indir.co-bind	POU2f	yes	no
CTCF	RUNX3	neg	indir.co-bind	both	no	no
CTCF	SRF	neg	indir.co-bind	CTCF	yes	no
CTCF	USF1	neg	indir.co-bind	both	yes	no
CTCF	YY1	neg	indir.co-bind	CTCF	yes	yes
CTCF	ZNF143	neg	indir.co-bind	CTCF	yes	no
JunD	BHLHE40	neg	indir.co-bind	BHLHE40	yes	no
JunD	CTCF	neg	indir.co-bind	CTCF	yes	no
JunD	EBF1	neg	indir.co-bind	EBF1	yes	no
JunD	EGR1	neg	indir.co-bind	EGR1	yes	yes
JunD	ELK1	neg	unknown		no	no
JunD	IRF4	neg	indir.co-bind	JunD	yes	yes
JunD	MAZ	neg	indir.co-bind	MAZ	no	no
JunD	PAX5	neg	indir.co-bind	PAX5	yes	no
JunD	SP1	neg	indir.co-bind	SP1	yes	yes
JunD	USF2	neg	indir.co-bind	USF2	yes	no
JunD	YY1	neg	indir.co-bind		yes	yes
JunD	ZBTB33	neg	unknown		yes	no

Table 3.1 – Identified associations : Details of all the TF associations identified, as well as the possible molecular mechanisms explaining them. The columns 'TF_A' and 'TF_B' refer to the TF involved in the association, 'Motif.ass.' to whether both motif are associated together ('positive') or repel each other ('negative'), as measured by the Fisher test, 'Type' to the proposed interaction mechanism between both TFs, 'Binder' to the TF binding DNA in case of an indirect co-binding, the value 'both' means that both tethering complexes may exist, 'Reported' to whether this interaction has already been reported in one of the following study Wang et al. (2012); Neph et al. (2012); Consortium (2012); Guo et al. (2012) and 'Validated' to whether this physical association is experimentally validated and reported in BioGRID v.3.4.145 (Chatr-aryamontri et al., 2017).

3.5 CTCF and JunD interactomes

The study of co-binding with CTCF showed that it was possible to detect global associations. I already detected that the cohesin complex members SMC3 and RAD21 form a complex with CTCF, as expected from literature (Ghirlando and Felsenfeld, 2016). Additionally, I detected that YY1 and ZNF143 are also frequently associated with CTCF, which has also been reported (Ong and Corces, 2014).

Thus, I decided to push forward in this direction. To this end, I set up a method based on motif co-occurrences to i) relieve the necessity of observing similar chromatin architectures, as in the previous section and ii) be able to functionally characterize the detected interactions.

As previously discussed (see section 1.2.1), several types of functional interactions between two TFs A and B exist : direct co-binding, indirect co-binding, independent co-binding and interference. Because the binding mechanisms are different from each other, different observations are expected. In the case of direct co-binding, both TF motifs are expected to occur in close vicinity, more often than by chance. Moreover, a spatial constrain (both spacing and orientation) reflecting the complex structure is also expected to occur. In the case of indirect co-binding, if TF_A is the factor binding its motif and TF_B is the tethering factor, both motif occurrences are expected to repel (avoid) each other at TF_A binding sites. In the case of independent co-binding, both $motif_A$ and $motif_B$ are expected to be enriched at both TF_A and TF_B binding sites. However, no spatial constrain is expected between the motif occurrences. Finally, in the case of interference, both motifs occurrences are expected to overlap. However, this may be difficult to detect.

In order to collect more evidence about functional connections between TFs, I developed a simple analysis pipeline able to detect the expected patterns of motif occurrences described above. Briefly, given a set of binding sites for a TF_A, it is possible to construct a contingency matrix containing the number of binding site with i) an occurrence of $motif_A$ and $motif_B$, ii) an occurrence of $motif_A$ only, iii) an occurrence of $motif_B$ only or iv) no motif occurrence and assess whether both motifs are associated or avoid each other using an exact Fisher test. Then, for pairs of motifs showing an association, displaying the spatial distribution of the motif occurrence may help to discriminate whether or not there is a spacing constrain or a motif overlap.

I investigated the association of 47 TFs for which 53 datasetes were available in GM12878 cells with CTCF or JunD. CTCF was chosen because i) most of its binding sites have a short nucleosome depleted region and show a sharp peak of sequence conservation at the binding site leaving a restricted space for other motifs to co-occur (Figure A.1) and ii) I already collected several observation regarding CTCF. JunD was chosen as a complementary example to CTCF in the sense that i) contrarily to CTCF, it is only a trancriptional regulator, ii) it is expected to bind to regulatory regions mostly, thus to open chromatin regions where other motifs are expected to occur, iii) ~50% of the peaks have a motif occurrence versus ~80% to ~90% for CTCF peaklists (Figure 3.2).

The motif co-occurrence analysis suggested several interactions. Regarding CTCF motif (Figure 3.6A), 8 positive motif association (ATF2, EBF1, MAZ, NFYb, NFkB, PAX5, SP1, YY1) and 16 negative motif associations (BATF, ELF1, IRF4, MEF2a, MEF2c, NFATc, NFYa, NRF1, NRSF/REST, PAX5, POU2F2/OCT2, RUNX3, SRF, USF1, YY1 and ZNF143) were found. Regarding JunD (Figure A.7A), positive motif association with 2 others TF motifs (BATF, cFos) and 12 negative associations with others TF motifs (ATF2, BHLHE40, CTCF, EBF1, EGR1, ELK1, IRF4, MAZ, PAX5, SP1, USF2, YY1 and ZBTB33) were found. cFos and one of the YY1-Sydh peaklists displayed evidence of poor quality (not shown and annotated as such by the ENCODE Consortium). Additionally, ATF2 belongs to the members of the AP1 family that have a 2bp spacer (TGANNTCA) while JunD has a 1bp spacer (TGANTCA). Thus the strong negative interaction may simply be due to the fact that both motifs are simply mutually exclusive. In consequence, the positive associations CTCF-YY1 and JunD-cFos and the negative association JunD-ATF2 should be ignored. Additionally, JunD and BATF motifs are the same as both these TFs belong to the AP1 family. In consequence, it is impossible to say whether BATF peaks harbour a JunD or a BATF site. Thus this association should be ignored as well, leaving no positive association left with JunD motif.

The analysis of CTCF and JunD motif occurrence densities (Figures 3.6B and C and Figure A.7B and C) revealed further interesting details regarding possible association mechanisms. First, positive associations showed CTCF density patterns mostly compatible with the direct co-binding and the independent co-binding scenarios (see Figure 3.6B). However, making a clear distinction between both is often really difficult. For instance, both EBF1 peaklists showed a decreased in CTCF motif occurrence density ~10bp after the peak, followed by an increase which could represent the spacer between CTCF and EBF1. However this is followed by a rather wide CTCF motif occurrence presence, mostly suggesting an independent co-binding scenario. An interesting candidate for a direct co-binding with CTCF is RXRa (Figure 3.6B). Even though the motif association was not significant, a focused co-localization of both motif appears. Second, negative associations showed CTCF and JunD density patterns compatible with the indirect co-binding scenario where the TFs would tether through CTFC or JunD, i.e. the CTCF or JunD motif occurrences do not show a spacing constrain with the binding sites but are rather spread over 100bp around binding sites without their own motif (Figure 3.6C and Figure A.7C). Interestingly, CTCF motif occurrence around YY1 and ZNF143 binding sites lacking their own motifs (see bottom of Figure 3.6C) showed really focused densities, indicating that for some reason, the occurrence of CTCF motif is well localized. Even if unexpected, this observation is not incompatible with the indirect co-binding scenario and further supports the results from section 3.4.

To summarize, the motif association statistics allowed me to identify 35 associations of TFs with either CTCF or JunD (Table 3.1). The strongest negative interactions for CTCF were ZNF143 and YY1, supporting the results found in the previous sections. The analysis of CTCF and JunD motif occurrence distributions around peaks and a closer examination of the contingency matrices allowed to suggest details about the interacting mechanisms, including which TF binds DNA. The only two exceptions were JunD-ELK1 and JunD-ZBTB33 for which

the motif occurrence densities were uninformative. Finally, out of these 35 associations, 5 were supported by experimental evidence and 5 were not reported in previous studies or databases (Wang et al., 2012; Neph et al., 2012; Consortium, 2012; Guo et al., 2012; Chatr-aryamontri et al., 2017).

3.6 EBF1 binds nucleosomes

EBF1 is a crucial factor for B cell development. It is necessary in the early steps, for a proper lineage commitment as well as later on during the entire B cell development (Boller et al., 2018). Since many years, EBF1 has been though to be able to "pioneer early changes in the target gene chromatin necessary for transcriptional activation" and proper B cell development (Hagman and Lukin, 2005). Experimental evidence supported that EBF1 could be able to bind compacted naive chromatin (without noticeable mark/modification), leading to a local chromatin opening, H3K4me2 deposition, DNA demethylation and gene activation (Maier et al., 2004; Boller et al., 2016). If such features makes a lot of sense during lineage commitment, the some underlying mechanisms remained mysterious, especially how EBF1 primarily binds to closed chromatin. With regard to this, the results of section 3.2, suggesting that EBF1 binding sites may be covered by nucleosome arrays, rose my attention. In order to collect evidence that may shed light on this, I conducted a deeper exploration of the EBF1 binding sites.

First, the distribution of nucleosome dyads - from two independent experiments - around EBF1 binding sites revealed a landscape that is compatible with a nucleosome positioned 70bp apart from the binding sites (Figures 3.7A). This configuration would position the EBF1 binding sites at the edge of the nucleosome. The 10bp periodicity visible suggested that other positioning of the EBF1 binding site exist but always at integer numbers of helix turn, such that the EBF1 binding site would always be positioned the same compared to the nucleosome surface. Surprisingly, the distribution of EBF1 motif occurrence remained the same, whether the nucleosome was containing an EBF1 bound site or not (Figure A.8).

Second, to support the fact that these EBF1 binding sites are indeed functional sites, I compared some of their chromatin features with the entire nucleosome pool. As expected, the presence of EBF1 binding sites was correlated with an increased accessibility (Figure A.10A), even though the opening was spread rather than narrow. Furthermore, this increased opening was concomitant with an enriched H3K4me2 deposition (Figure A.10B), in line with the literature. Last, it was also possible to highlight a higher sequence conservation at the nucleosome edges when they had an EBF1 binding site (Figure A.10C), suggesting a functional difference between both nucleosome pools.

Third, a further inspection of the sequence composition of the nucleosomes bearing an EBF1 binding site revealed i) a periodic occurrence of antiphased WW (W=A/T) and SS (S=C/G) dinucleotides and ii) a periodic occurrence of the YRRRRRYYYYYR (R=A/G, Y=C/T) nucleosome positioning motif described by Trifonov (Trifonov, 2011). Together, these observations

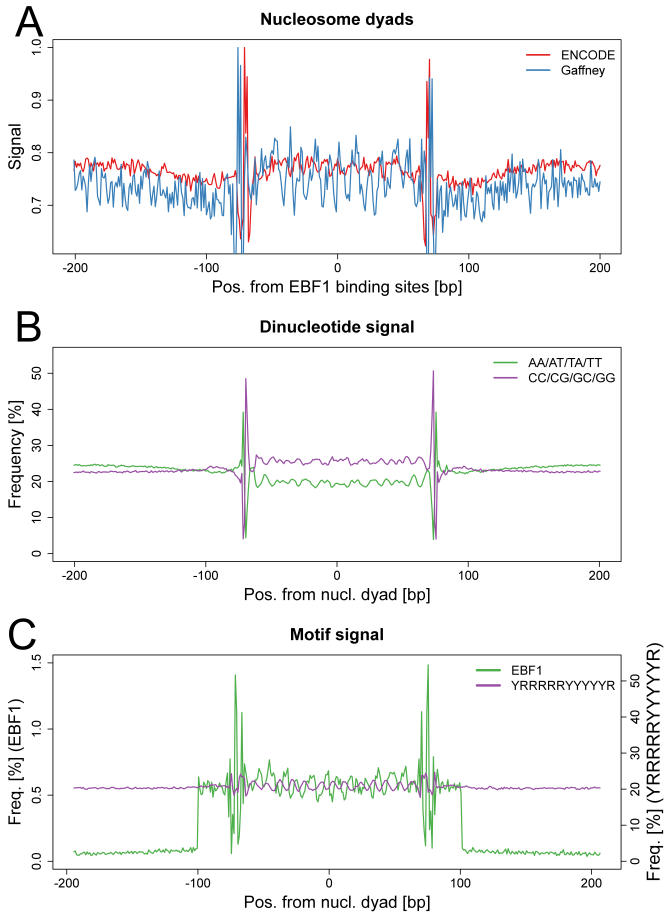


Figure 3.7 – EBF1 binding sites stand on the edge of a nucleosome. **A** Nucleosome dyad distributions around the EBF1 binding sites (from the Haib dataset). The dyad distributions have been measured from two independent datasets : i) MNase-seq data released by the ENCODE Consortium (in red) and by Gaffney et al. (in blue) (Gaffney et al., 2012). **B** Dinucleotide frequencies around the nucleosome dyads from the Gaffney dataset that have an EBF1 binding site within 100bp. **C** Motif occurrence frequency around the nucleosome dyads from the Gaffney dataset that have an EBF1 binding site within 100bp. The abrupt decrease of EBF1 motif occurrence frequency at +/- 100bp reflects the nucleosome selection process.

suggest that EBF1 binding sites are located on the edge of a rotationally positioned nucleosome (Ioshikhes et al., 2011; Trifonov, 2011; Gaffney et al., 2012). Interestingly, Trifonov's motif occurs in counter phased with EBF1 motif occurrences. A closer look at both motifs (see Figure A.9 for EBF1 logo) revealed that half of Trifonov's motif (RRRRR or YYYYY) matches one half of the EBF1 motif ({A/C}CCC{A/C} or {A/G}GGG{A/G}) at the cost of 2 or 0 missmatches.

These results suggest that EBF1 can indeed bind nucleosomal DNA. The bound motif occurrences were predominantly located at the edges of the nucleosomes. Yet, this was also the fact for nucleosome that are not bound by EBF1. This suggests that nucleosomes are already in this position before EBF1 binding, which may be the case given the presence of favorable nucleosome positioning sequences.

The reason why the EBF1 motif occurrences are already on the edges of nucleosome, even without EBF1 binding, remains unknown. One explanation could be that such sites have a double function. The first function would be to recruit EBF1 to open up the region. The second, would be that EBF1 binding sequence (together with other positioning sequences) can act as a barrier - a potential well - avoiding the nucleosome to roll over in this direction. Such a system would have the advantage of promoting a suited chromatin structure in developmentally important regions. Constraining nucleosome movement could serve to hide regulatory elements. At the same time, these regions would remain responsive to differentiation signals through the exposition of EBF1 sites on the periphery of nucleosomes.

3.7 Discussion

Overall, the results presented in this section complement and support the observations made by other research groups worldwide.

The systematic study of the nucleosome landscape in the vicinity of TFs binding sites highlighted that nucleosome arrays are always present on the flanking regions. However, all the TFs, with the exception of CTCF, do not act as a barrier and thus are not major determinants of the chromatin architecture. Instead, an alternative mechanism, probably involving chromatin remodelers, is likely to be responsible. Furthermore, all TFs were found to bind in NDRs with the noticeable exception of EBF1.

Surprisingly, a large fraction of EBF1 binding sites was found to be occupied by what seemed to be a rotationally positioned nucleosome, which edges were bound by EBF1. Furthermore, it appeared that EBF1 binding motif resembles a nucleosome positioning sequence and could be involved in the positioning of the nucleosome. However, at least two alternative scenarios could explain the presence of an EBF1 binding site at the entry of a nucleosome. First, EBF1 genuinely binds to such "pre-positioned" nucleosomes, in which case I am observing EBF1 true binding mechanism. Alternatively, EBF1 binding - to either nucleosomal or naked DNA - results in the positioning of a nucleosome right beside. To my opinion, the previous results suggesting a pioneer function for EBF1 (Boller et al., 2016) makes the second hypothesis more

likely. EBF1 would directly engage a nucleosome and somehow trigger its displacement such that EBF1 binding site will eventually reside at the nucleosome edge. Testing this hypothesis could be performed by assaying in vitro binding of EBF1 to assembled nucleosome arrays.

The study of CTCF binding sites revealed that they can be grouped in i) promoter distal and ii) promoter proximal binding sites. In each of the subsets, CTCF was observed to bind with a different group of interactors, suggesting different functions. At promoter distal binding sites CTCF is associated the cohesin complex while at promoter proximal regions, CTCF seems to be associated with ZNF143 and YY1.

Finally the study of the motif co-localization, even if simple, seemed quite powerful as it allowed to identify 35 interactions with CTCF or junD. Out of these, 25 have already been proposed but without experimental support, 5 have been proposed and experimentally validated and 5 were new. These 5 new interactions are proposed to be indirect co-binding event and thus imply a physical interaction that can be tested.

3.8 Methods

3.8.1 Data and data processing

All the GM12878 ENCODE data used were mapped against hg19 genome and can be found on the MGA repository (Dreos et al., 2018).

Peaks called by the ENCODE Consortium using their uniform processing pipeline Gerstein et al. (2012) were used. These peaks can be found at https://ccg.epfl.ch/mga/hg19/encode/Uniform-TFBS/Uniform-TFBS.html. Assuming that a TF binds to DNA through motif recognition, the peak center should be localized on a motif occurrence center. Thus the center of each peak was moved to the closest motif occurrence within 60bp. To do so, each TF was associated to a log-odd PWM contained either in JASPAR Core vertebrate 2014 Mathelier et al. (2014), HOCOMOCO v10 Kulakovskiy et al. (2016) or Jolma Jolma et al. (2013) collection. Using the corresponding log-odd PWM, peak sequences were scanned to find motif occurrence with a score corresponding to a pvalue higher or equal to 1e-4. If such a motif occurrence was found, the peak position was shifted to the center of the motif occurrence and mapped to the corresponding strand. Otherwise, the peak position remained unchanged without strand information.

In GM12878 cells, nucleosome occupancy was assessed using MNase-seq data released by the ENCODE Consortium (GSE35586). These data can be found at https://ccg.epfl.ch/mga/hg19/encode/GSE35586/GSE35586.html. To increase sequencing depth, all replicates available for this cell line were pooled together, resulting in 789 mio reads, and used as a single dataset. The resulting dataset is available and has the description "GM12878|Nucleosome|all (SLOW!)". Because each read was represented as a single point coordinate corresponding to their 5' edges, these coordinates were centered by 70bp in order to indicate the nucleosome dyads. Finally,

another dataset was used for one analysis only. These data were released by Gaffney and colleagues Gaffney et al. (2012) and can be found at https://ccg.epfl.ch/mga/hg19/gaffney12/gaffney12.html and were not centered as the coordinates already represent the center of paired-end sequenced fragments. The dataset is labeled "All Paired-end samples - 147bp fragments".

Chromatin accessibility was assessed using DNaseI-seq data released by the ENCODE Consortium Boyle et al. (2008) (GSE32970). To increase sequencing depth, all replicates available for GM12878 cells were pooled together, resulting in 144 mio reads, and used as a single dataset. The individual replicates can found at https://ccg.epfl.ch/mga/hg19/encode/Duke-DNaseI-HS/Duke-DNaseI-HS.html. The reads were represented as a single point coordinate corresponding the their 5' edges but were not centered as this corresponds to the exact DNaseI nick location.

The EPDnew release 003 was used as TSS annotation Dreos et al. (2017) and genome sequence conservation was assessed using Phastcons Siepel et al. (2005). Both datasets can be found at https://ccg.epfl.ch/mga/hg19/epd/epd.html and https://ccg.epfl.ch/mga/hg19/phastcons/phastcons.html respectively.

3.8.2 Classification of MNase patterns

For each TF peaklist MNase, DNase, sequence conservation and TSS density around TF binding site were assessed independently by counting the number of read mapped from -999bp to +1000bp around each peak, using 10bp bins. For each TF, 4 matrices having one row per binding site (peak) and 199 columns were created using ChIP-extract program (Ambrosini et al., 2016a).

Probabilistic pattern classification was achieved using the ChIPPartitioning (see section 3.2). The algorithm was implemented as described in the supplemental materials of Nair et al. (2014).

Two different procedures were used to classify MNase patterns. Both were run for 10 iterations allowing flip and a value of shift of 15 bins.

The first procedure aimed to discover 4 different pattern classes, allowing flip and a shift of 15 bins. The procedure was initialized with 4 classes. The class patterns were initialized by assigning each peak a random probability to belong to each of the 4 classes. The patterns were then computed as the weighted average of the signal given the peak class probabilities as weights. Then the prior class probabilities were initialized as $p_{k,s,f} = 1/K * S * 2$ where k is the class index, s is the shift value in bins (here 15), f is an indicative variable for the flip state (1 for "normal", 2 for "reverse"), K is the number of classes (here 4) and S is the maximum allowed shift in bins. The classification was run for 10 iterations. At the end, it returned a matrix of dimensions $NxKxSx2$ containing the probabilities for each of the N region to belong to each of the K class, for each possible shift state S and for both flip states ("normal" or "reversed").

The second procedure aimed to discriminate between 2 classes : i) the binding sites describing the "average" binding sites as opposed to ii) those differing from this. To do so, class patterns were initialized to i) the aggregation over all peaks (the average pattern) and ii) a flat pattern being the mean number of counts of the input matrix. Flip and 15 bins of shift were allowed. The prior class probabilities were initialized as $p_{k,s,f} = \mathcal{N}(s, floor(S/2)+1, 1)$ where the second and third parameters are the mean and the standard deviation, giving a higher prior probability to states with shift equal to 0bp.

3.8.3 Quantifying nucleosome array intensity from classification results

Nucleosome array intensity was quantified using a method developed by Zhang and colleagues (Zhang et al., 2014). Briefly, nucleosome signal is represented in 2 dimensions as a set of signal intensities for a given set of positions. Data are structured as vector Y containing the nucleosome occupancy signal (for instance an EM classification class profile) for n bins (for EM class profiles, 199 bins of 10bp). First, the 1^{st} order derivative D_1 of Y is computed. Then the 1^{st} order derivative D_2 of the absolute value of D_1 is computed. Local maxima in D_2 are searched using a windows of 15 bins (corresponding to 150bp, a nucleosome width). Maxima can be interpreted as strong drop or enrichment of signal, corresponding to a pattern expected from a well positioned nucleosome array. Finally, all D_2 maxima are joint by a line and the nucleosome array intensity at each given position is the height of the line at this position. The nucleosome array density for the first and last position of Y were set to 0. The average nucleosome array intensity of Y was used as the nucleosome array value of the input data.

The classification of a matrix of counts having N rows (regions), with K classes, allowing a maximum of S shift states and two flip states ("normal" and "reverse") outputs a probability matrix P of dimension $[N, K, S, 2]$ containing the probability for each region to belong to each class, given a shift state and a flip state. This matrix can be used to compute a vector D_k of length S containing the probability density of the shift states for a class k using :

$$D_{k,s} = \frac{\sum_{i=1}^{N}(P_{i,k,s,1} + P_{i,k,s',2})}{\sum_{i=1}^{N}\sum_{s=1}^{S}(P_{i,k,s,1} + P_{i,k,s',2})} \quad \text{with} \quad s' = S - s + 1 \tag{3.3}$$

(Ambrosini et al., 2016a) where s' represents the index of the reverse orientation and with the constrain that all the elements of P sum to 1. Given the shift probability density vector D_k of

one class, computing its standard deviation was done using :

$$\sigma_k = \sqrt{\sum_{i=1}^{S}(X_i^2 \cdot D_{k,i}) - \mu_k^2}$$
$$\text{with} \tag{3.4}$$
$$\mu_k = \sum_{i=1}^{S}(X_i \cdot D_{k,i})$$

where X is a vector containing the position changes in bp for every shift state, i.g. for a maximum number of shift states of 15 ($S = 15$) with bins of 10bp, X would contain [-70, -60, ..., 0, ..., +60, +70].

3.8.4 Peak colocalization

To measure the extent of colocalization between CTCF, YY1, ZNF143, SMC3 and RAD21, the occurrence of YY1, ZNF143, SMC3 and RAD21 peaks around CTCF peaks was computed using ChIP-extract (Ambrosini et al., 2016a). The CTCF peak list used as reference was "wgEncodeAwgTfbsSydhGm12878Ctcfsc15914sc20UniPk" because it was the CTCF peak list containing i) the most CTCF peaks and ii) the highest proportion of peaks with a motif occurrence. Chip-extract was run separately for YY1, ZNF143, SMC3 and RAD21 using the following parameters : from -99, to 100, window size 1. Then, the propotion of CTCF peak having at least one other peak within +/-10 bp, 50bp or 100bp was computed.

3.8.5 NDR detection

Let us consider a matrix of MNase-seq counts R of dimensions NxL containing N vectors of read counts $r_1, r_2, ..., r_n$ of length L. Because MNase-seq reads are a direct indication of the nucleosome occupancy, detecting NDRs is about finding low signal regions, flanked by two high signal regions.

The signal in each vector X_i (region) is assumed to have been sampled from a 2 class mixture of high (nucleosome) and low (nucleosome-free) signal, using a Poisson distribution. Both classes are expected to occur with a given probability p_i^{nucl} and p_i^{free}. The rows are considered individually to lessen technical biases such as region specific sequencing depth.

The class probabilities and their mean parameters are estimated using an EM algorithm. First, during the E-step, for each position inside a region, the posterior probability of the

nucleosome given the data is computed using :

$$P(nucl|r_{i,l}) = \frac{p_i^{nucl} \times Poisson(r_{i,l}, \lambda = m_i^{nucl})}{p_i^{nucl} \times Poisson(r_{i,l}, \lambda = m_i^{nucl}) + p_i^{free} \times Poisson(r_{i,l}, \lambda = m_i^{free})} \tag{3.5}$$

where $r_{i,l}$ is the number of reads at position l in the i-th row of R, m_i^{nucl} and m_i^{free} are the mean parameters of the nucleosome and nucleosome-free classes respectively. Obviously, the nucleosome-free class posterior probability is

$$P(free|r_{i,l}) = 1 - P(nucl|r_{i,l}) \tag{3.6}$$

Then, during the M-step, the class mean parameters are updated using

$$\begin{aligned} m_i^{nucl} &= \sum_{l=1}^{L} r_{i,l} \times P(nucl|r_{i,l}) \\ m_i^{free} &= \sum_{l=1}^{L} r_{i,l} \times P(free|r_{i,l}) \end{aligned} \tag{3.7}$$

and the class probabilities :

$$\begin{aligned} p_i^{nucl} &= \frac{1}{L} \times \sum_{l=1}^{L} P(nucl|r_{i,l}) \\ p_i^{free} &= 1 - p_i^{nucl} \end{aligned} \tag{3.8}$$

The EM optimization of the parameter estimates was repeated for 10 iterations. At the end of the parameter estimation process, each of the L positions in a region R_i were assigned two posterior probabilities $P(nucl|r_{i,l})$ and $P(free|r_{i,l})$ to belong to each class. In all cases, the nucleosome class was the class having the highest mean parameter and the nucleosome free class the class with the smallest ($m_i^{nucl} > m_i^{free}$).

The binding sites - located in the center of the regions, at position $s = L/2$ - were assumed to

be within the NDR. From that point, the NDR was extended using the following procedure :

Algorithm 1: Searches the coordinates of the NDR using the posterior nucleosome and nucleosome free class probabilities, for a region R_i, from its central position.

```
1  float NDRextend(){
       Data: The posterior probabilities obtained for each position of r_i.
       Result: the left and right coordinates of the NDR
2      // NDR only covers the central location
3      left = s;
4      right = s;
5      while left ≠ 2 and right ≠ L − 1 do
6          p.free.l = P(free|r_{i,left});
7          p.free.r = P(free|r_{i,right});
8          p.nucl.l = P(nucl|r_{i,left});
9          p.nucl.r = P(nucl|r_{i,right});
10         // bidirectional extension
11         if prob.free.l > p.nucl.l and p.prob.free.r > p.nucl.r then
12             left −= 1;
13             right += 1;
14         end
15         // extension to left
16         else if prob.free.l > p.nucl.l then
17             left −= 1;
18         end
19         // extension to right
20         else if p.prob.free.r > p.nucl.r then
21             right += 1;
22         end
23         // no more extension possible
24         else
25             break;
26         end
27     end
28     return left, right
```

The nucleosome occupancy around CTCF binding sites was measured using ChIP-extract with "wgEncodeAwgTfbsSydhGm12878Ctcfsc15914sc20UniPk" peak list as reference - because it was the CTCF peak list with the most peaks and with the highest proportion of peaks with a CTCF motif -, the ENCODE MNase-seq data described in section 3.8.1 as targets and the following parameters : from -999bp, to 1000bp and window size 10bp.

This matrix was subjected to a ChIPPartitioning partitioning, as described in section 3.8.2, to find 4 nucleosome architectures, using shifting and flipping. The resulting posterior prob-

abilities were used to re-orient the data. If the major shift state - that is the shift state with the highest overall probability - for a given region was the "reverse" state, then the row was reversed. The re-oriented matrix was then subjected to the NDR detection. The re-orientation was done for aesthetic purposes only. Because the NDR detection was performed starting from the center position in each region - and given that reverting a vector did not change its central position - this operation had no influence on the NDR detection.

3.8.6 CTCF and JunD interactors

To enumerate the occurrences of CTCF and JunD motifs, the hg19 genome assembly was scanned using CTCF (MA0139.1 from JASPAR Core Vertebrate 2014 (Mathelier et al., 2014)) and JunD (JUND_HUMAN.H10MO.A from HOCOMOCOv10 (Kulakovskiy et al., 2016)) matrices to produce lists of potential binding sites. A limit score threshold was set as the score corresponding to a pvalue of 1e-5 for each matrix, respectively. This was done using matrix_scan program from PWMScan (Ambrosini et al., 2018). Eventually, any motif occurrence falling inside a region classified as being a repeated element and blacklisted by the ENCODE Consortium was filtered out using count_filter program from the ChIP-seq tools (Ambrosini et al., 2016b).

Then, for each TF peak list independently, the number of i) the TF and ii) CTCF/JunD occurrences +/- 1kb of each peak was measured, in bins of 1bp, using ChIP-extract program from the ChIP-seq tools (Ambrosini et al., 2016b). The association were measured as follows : using the ChIP-extract results for the given peak list versus i) the TF and ii) CTCF/Jund motif occurrences, the number of peaks having i) at least one TF and one CTCF/Jund motif occurrences, ii) only TF motif occurrences, iii) only CTCF/JunD motif occurrences or iv) no motif occurrences. These numbers were used to build a contingency table and a two-sided Fisher exact test for association was performed. The motif relationship was considered significant if the test OR was bigger than 1 and the 95% CI of the OR did not contain 1 or as a significant motif exclusion if the OR was smaller than 1 and the 95% CI of the OR did not contain 1.

The motif occurence densities were computed from the ChiP-extract result matrices. Out of each matrix, a vector containing the number of motif occurrences at each possible absolute distance was computed. This was done as follows : first each non-null cell neighbours were incremented (+/- 5 columns on each side) to turn motif occurrences into non point-like representation. A given cell value could be incremented several times. Second for each row, the column corresponding to the same absolute distances from the peak were summed together (i.g. +1bp with -1bp, +2bp with -2bp, +999bp with -999bp). The first column of the resulting matrix should contain the number of motif occurrences present at the peak center (distance of 0bp), the second column at an absolute distance of 1bp and so one. Eventually, the rows were summed up and the resulting vector was considered as the motif occurrence density vector for the given peak list. The vectors were used to create a matrix for CTCF motif and Jund motif occurrences (a vector corresponds to a row), separately, and the matrix was displayed as a heatmap. The row values were standardized and the rows hierarchically clustered using the

euclidean distance.

3.8.7 EBF1 and nucleosome

The correlation between EBF1 binding sites and nucleosome dyads was made using ChIP-cor (Ambrosini et al., 2016b), from the web (https://ccg.epfl.ch/chipseq/chip_cor.php). The references were the corrected EBF1 peaks (wgEncodeAwgTfbsHaibGm12878Ebf1sc137065Pcr1xUniPk dataset, for more details see section 3.8.1) and the targets either i) the MNase-seq data released by Gaffney et al. (Gaffney et al., 2012) (hg19 / DNase FAIRE etc / Gaffney 2012 ... / All Paired-end samples - 147bp fragments) or ii) the ENCODE MNase-seq data (hg19 / ENCODE DNase FAIRE etc / GSE35586 ... / GM12878 Nucleosome all (SLOW!)). In both cases, "any" strand was selected. Because Gaffney data are paired-ended and represent the fragment midpoint (the dyad), no centering was done. The ENCODE data are single-ended and a centering of 70bp (half a nucleosome) was applied to approximate the fragment midpoint. The count cut-off was set to 1 and the range to -399 to +400bp.

To isolate nucleosomes with an EBF1 binding site, the opposite ChIP-cor analysis was run : Gaffney data as references versus EBF1 binding sites as targets with count cut-off set to 1 and the range to -399 to +400bp. In the results page the "Feature Selection Tool" was used to select dyads with at least 1 EBF1 binding site (threshold parameter) located "From" -99bp "To" 100bp. The count cut-off was set to 9999 and both "Switch to depleted feature" and "Reference feature oriented" set to "Off".

These nucleosome dyads were uploaded to OProf (https://ccg.epfl.ch/ssa/oprof.php) on the SSA server (Ambrosini et al., 2003). Four individual analyses were run to measure the "WW", "SS", "YRRRRRYYYYYR" and EBF1 motif occurrences. In all cases, the 5' and 3' borders were set to -399bp and 400bp, the window shift to 1bp and the search mode to "bidirectional". For "SS" and "WW", the motif to search was entered as a "Consensus sequence", the window size was set to 2bp, the reference position to 1 and the number of allowed mismatches to 0. For "YRRRRRYYYYYR", the motif was also entered as a "Consensus sequence", the window size was set to 12bp, the reference position to 6 and the number of allowed mismatches to 4. For the EBF1 motif, the JASPAR CORE Vertebrate 2018 "EBF1 MA0154.3 (length=14)" was used with a window size of 14bp, a reference position of 7 and a p-value threshold of 1e-4.

To investigate the chromatin architecture around nucleosome dyads, ChIP-cor was used. Two references were used : i) the nucleosomes with an EBF1 binding site (see above) and ii) the entire Gaffney dataset (hg19 / DNase FAIRE etc / Gaffney 2012 ... / All Paired-end samples - 147bp fragments). For each reference, three analyses were run against different target features : i) DNase-seq data to monitor chromatin accessibility (hg19 / ENCODE DNase FAIRE etc / Boyle 2008 ... DNaseI HS - GM12878 - Rep 1) with "any" strand and no centering, ii) H3K4me2 ChIP-seq data (hg19 / ENCODE ChIP-seq / GSE29611 ... / GM12878 H3k4me2) with "any" strand and a centering of 70bp (half the nucleosome) and iii) positional sequence conservation scores (hg19 / Sequence derived / Vertebrate Conservation (phastCons46way)

... / PHASTCONS VERT46) with "any" strand an no centering. For DNase-seq and sequence conservation, the range was set to -399bp to 400bp with a window with of 1bp. For H3K3me2 data, the range was set to -3999bp to 4000bp with a window width of 10bp. For the DNase-seq and the H3K4me2 data, the count cut-off were set to 1, for the sequence conservation to 9999.

4 SPar-K

This chapter describes SPar-K (Signal Partitioning with K-means), a modification of the K-means algorithm to cluster genomic regions based on their chromatin organization, defined by their sequencing profiles.

Due to the wealth of sequencing data, it is common to analyze positional correlations between chromatin features, e.g. the position of nucleosomes (revealed by MNase-seq) relative to transcription factor binding regions (mapped by ChIP-seq) in order to shed light on their functional relationship. A typical analysis workflow usually contains a partitioning step which objective is to creates groups (clusters) of regions sharing common features.

The K-means algorithm is typically used to partition genomic regions based on their sequencing densities (the number of reads found at each position). However, it is intrinsically limited in the sense that K-means will only perform position-wise comparisons of the regions. Thus, if two regions contains the same signal but no aligned - not at the identical offsets - the regions will be defined as dissimilar.

Several methods and software have been developed for discovering chromatin patterns by clustering and/or realigning read density profiles over genomic regions (see section 1.6), including ChromaSig (Hon et al., 2008), ArchAlign (Lai and Buck, 2010), CATCHProfiles (Nielsen et al., 2012), CAGT (Kundaje et al., 2012) and ChIPPartitioning(Nair et al., 2014). However, these programs have some limitations. Some do not perform a realignment, others are restricted to count data or lack an runtime efficient implementation, such as ChIPParititioning. To fill this gap, I developed SPar-K. SPar-K allows to partition genomic regions while aligning them properly to ensure that common stretches of signal, even if located at different offsets, are compared together.

I developed, implemented and benchmarked this algorithm and produced all the figures that are shown in this chapter. The content of this section is taken and adapted from the original article (Groux and Bucher, 2019).

4.1 Algorithm

SPar-K algorithm (see Algorithm 3) is a modified version of the regular K-means algorithm during which a set of N regions of size L are partitioned into K clusters, using an iterative optimization procedure. Each cluster is composed of an alignment of regions sub-parts of length $L'a$ assigned to this cluster and the cluster is summarized as a vector of length $L \geq L'$ that contains the average signal at each position in the alignment.

The input data are stored as a N rows and L columns matrix R. The signal resolution may be at single-base or at a larger bin size. The regions are typically defined by relative positions to an anchor point, e.g. a ChIP-seq peak summit. If the signal is noisy, a data smoothing step can be performed to average out outlier values (see Algorithm 4) and ease the partitioning procedure.

SPar-K optimizes the alignments by minimizing the sum of squares errors. That is, the sum of the squared distances of each point to the cluster aggregation they are assigned to.

The distance between any two regions is computed using a modified correlation distance. Let us assume two regions X and Y of length L and a shifting freedom S. X and Y will be sub-divided in S slices each. Each slice has a length of L'=$L - S + 1$ and starts at all possible offsets $s = 1, 2, ..., S$. All S^2 pairwise comparisons between any slices of X and Y are computed using $1 - cor(X_i, Y_j)$ where X_i and Y_j are the slices starting at offsets $i, j \in s$. If flipping is allowed, another set of S^2 comparisons is performed by flipping Y_j (that is, the 1st position in Y_j becomes the last and vice-versa), resulting in $2 \times S^2$ comparisons. Eventually, the distance between X and Y is the minimum of the S^2 (without flipping) or $2 \times S^2$ (with flipping) values. For each distance, the indices i and j and whether Y_j was flipped in the best comparison are remembered as they allow to rebuilt the optimal alignment between X and Y. The naive algorithm to do this is $\Theta(S^2 \times L')$ in time however I could design a faster algorithm which is $\Theta(S \times L')$ by using a dynamic programming approach (see algorithm 5).

SPar-K is initialized by choosing K regions to become the initial cluster aggregations of length L either i) randomly (Algorithm 10) or ii) using the K-means++ (Arthur and Vassilvitskii, 2007) sampling procedure (see Algorithm 11). Then, each regions is aligned against each cluster aggregation an assigned to the cluster to which it has the smallest distance with. Once all N regions have been aligned to a cluster, the cluster aggregations are updated by computing the average signal at each position in the alignments.

This procedure and is repeated until i) reaching the maximum number of iterations or ii) achieving convergence, that is when the alignments in each cluster do not change from one iteration to the next.

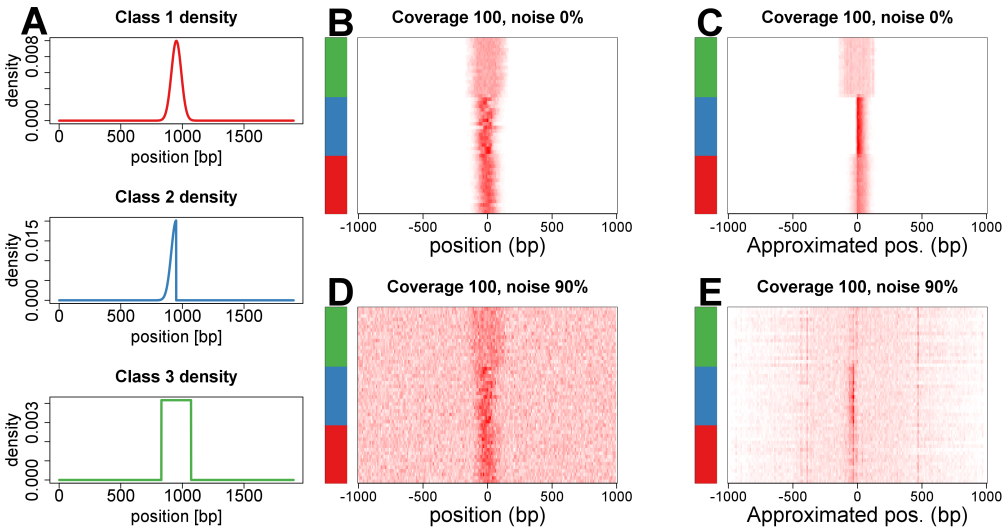


Figure 4.1 – Synthethic datasets : **A** The class signal densities. **B** A synthetic dataset with a mean coverage of a 100 reads per region in average (c=100) and 0% noise (p_s=1, p_b=0) and **C** one of the corresponding SPar-K partition, with shifting and flipping. The color ribbons on the side indicate the cluster assignments. **D** A synthetic dataset with a mean coverage of a 100 reads per region in average (c=100) and 90% noise (p_s=0.1, p_b=0.9) and **E** one of the corresponding SPar-K partition, with shifting and flipping. Figure and legend taken and adapted from (Groux and Bucher, 2019).

.

4.2 Implementation

SPar-K algorithm has been implemented as a stand-alone, fully multithreaded, C++ program. Regarding the parallellization, the computations at each step are independent of each other, leading to an "embarrassingly parallel" situation. Thus, at each step, the computations are split into equal amounts and distributed over a pool of worker threads. Eventually, the program returns a table listing for each region the cluster assignment, the shift state and the orientation. The software distribution also includes R scripts for visualizing the data as heatmaps as shown in Figure 4.5. The software source code is available from Github https://github.com/romaingroux/SPar-K and as Docker container https://hub.docker.com/r/rgroux/spar-k.

4.3 Benchmarking

First I compared SPar-K, regular K-means and ChIPPartitioning on synthetic datasets exhibiting properties that are plausible for ChIP-seq profiles for genomic regions.

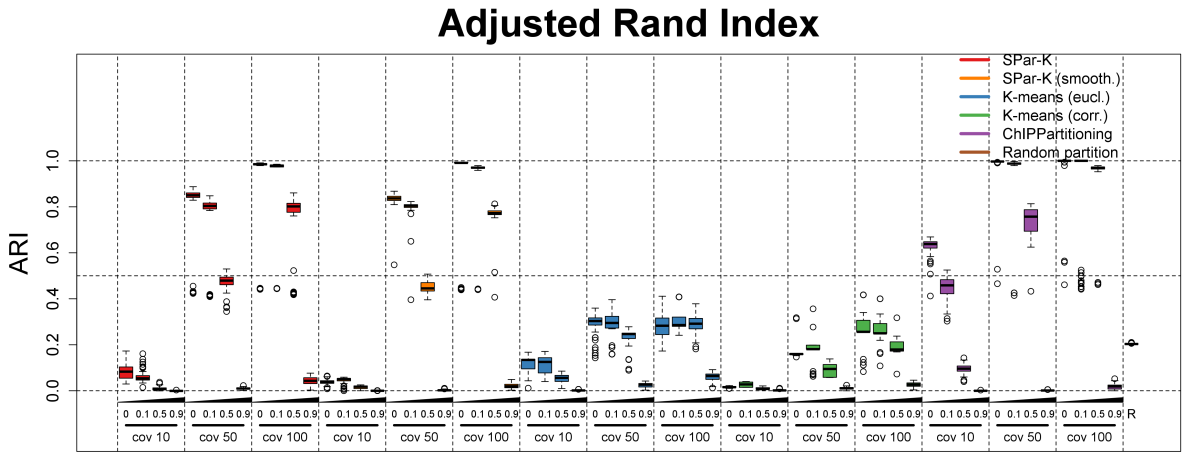


Figure 4.2 – Clustering accuracy using random seeding : to compare the clustering accuracies of the different methods, several simulated dataset containing 3 classes, different coverages (10, 50 and 100 reads per region indicated as "cov10", "cov50" and "cov100") and noise proportions (no noise, 10% noise, 50% noise and 90% noise indicated as "0.0", "0.1", "0.5" and "0.9") were generated. Each dataset was clustered 50 times with each method. The Adjusted Rand Index (ARI) was computed for each partition. The ARI values are displayed as boxplots. SPar-K and ChIPPartitioning were run allowing flipping and shifting. The ARI was measured on each of the resulting data partitions. For SPar-K, "smooth" indicates outlier smoothing. For the regular K-means, "eucl." and "corr." refer to the euclidean and correlation distances. "R" stands for "random" and indicates the ARI values obtained when comparing the true cluster labels with a randomly shuffled version of it, 100 times. Figure and legend taken and adapted from (Groux and Bucher, 2019).

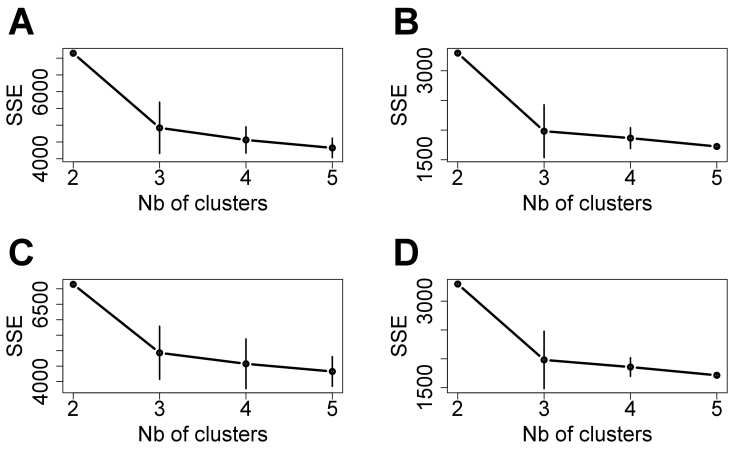


Figure 4.3 – Median SSE : for the simulated ChIP-seq dataset containing 3 classes, with coverage 100 and no noise, partitioned into 2 to 5 clusters. To judge whether the elbow method could be used to estimate the optimal number of clusters, this dataset was partitioned with SPar-K, allowing flip and shifting, into 2 to 5 clusters, 50 times for each set of parameters. For each number of clusters, the median SSE is shown, +/- 1 standard deviation (bars). **A** Seeding done at random, **B** seeding done at random and outlier smoothing **C** seeding done with the K-means++ method **D** seeding done with the K-means++ method and outlier smoothing. In all cases, the optimal number of clusters seemed to be 3 (which was the expected value). Figure and legend taken and adapted from (Groux and Bucher, 2019).

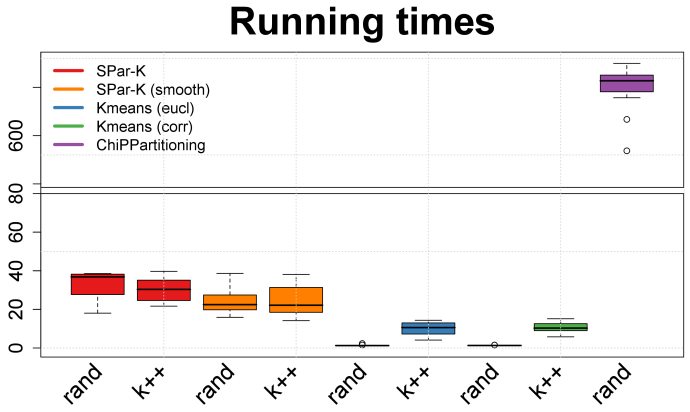


Figure 4.4 – Running times : to compare the run times of each program, the synthetic dataset with coverage 100 and no noise was partitioned 20 times with each program. The run times (wall clock) in second were measured. For all SPar-K and the regular K-means, the partitions were initialized using a random and K-means++ (indicated as "k++"). For ChIPPartitioning, only a random seeding was used. The partitions were then optimized for 30 iterations at most. For SPar-K and ChIPPartitioning, a shifting of 71 bins and flipping were allowed. For SPar-K, only one thread was used and "smooth" indicates outlier smoothing. For the regular K-means, "eucl." and "corr." refer to the euclidean and correlation distances. Figure and legend taken and adapted from (Groux and Bucher, 2019).

4.3.1 K-means

For the regular K-means, the "kccaFamily" function from the "flexclust" R package (Leisch, 2006) was used. Calls to kccaFamily(dist=distEuclidean, cent=centMean) or kccaFamily(dist=distCor, cent=centMean) were employed to partition the data using the euclidean distance and the correlation distance respectively. "distEuclidean" is a package defined function and "distCor", a custom function computing $1 - cor(x, y)$ for any two x and y vectors. If the correlation between x and y could not be computed (for instance the standard deviation of x or y is equal to 0), the correlation was assumed to be 0 (and the distance 1). The initial centers were chosen using one of the two following seeding strategies : i) a random sampling of K points or ii) K-means++, a strategy aiming at sampling K initial points as far as possible from each other.

4.3.2 ChIPPartitioning

The implementation was done in R programming language. The "em_shape", "em_shape_shift" and "em_shape_shift_flip" functions present in the supplemental material of (Nair et al., 2014) were taken as such and incorporated in a R wrapper (as in Chapter 3). For this method, the partitioning could only be initialized using a random procedure, as described in (Nair et al., 2014).

4.3.3 Data

I generated several synthetic datasets. Each dataset contained 1000 regions of 2001bp (+/- 1kb around a central position), equally distributed over 3 classes. The signal over a region was modeled as a mixture of class specific signal and of background signal. The class specific signal was modeled by a 1902 element density vector. The background signal was modeled using a second 1902 element density vector containing a uniform density. The first class density vector contained a Gaussian density with mean 951 and standard deviation 40 (Figure 4.1A upper panel). The second class density was a Gaussian density of mean 950 and standard deviation 40. To create an asymmetric signal class, the values at positions 950 to 1902 (comprised) were set to the minimal value found in the original density (Figure 4.1A middle panel). The last class contained a rectangular function with a step corresponding to the elements 830 to 1070 (Figure 4.1A lower panel). Finally, all the densities were normalized such that the sum of each vector was 1. From these densities, the λ values for a class k were computed using the following formula :

$$lambdas_k = signal_k * c * p_s + background * c * p_b \tag{4.1}$$

where $signal_k$ is the class characteristic signal density, $background$ a uniform density, c the coverage factor, p_s the overall signal proportion and p_b the overall background proportion, with the constraint $p_s + p_b = 1$.

For each region, a read signal of 1902bp long was randomly sampled from Poisson distributions with the $lambdas$ values as function parameters. Then, the signal vector was introduced, in a 2001 element long vector filled of 0's, at a given offset, in a given orientation. The offset was randomly sampled from 1 to 100 . The orientation was randomly sampled with a probability of 0.3 to be in the reversed orientation. Finally, the resulting 2001bp vectors were binned using a 10bp window, that is, the signal was summed up every 10 columns leading to the creation of 201 bin long vectors. At the end of the process, a dataset was stored as a matrix of 1000 rows and 201 columns. Two examples of synthetic datasets are shown in Figure 4.1B and D.

4.3.4 Performances

Performances were assessed using the Adjusted Rand index (see Figure 4.2 and Supplemental Figure 3 in (Groux and Bucher, 2019)) and the optimal number of classes was estimated using the elbow method (Figure 4.3). As expected, regular K-means performed poorly. On the contrary, SPar-K was equally accurate as ChIPPartitioning except for the lowest coverage class. Considering speed, Spar-K outperformed ChIPPartitioning by a factor of at least 20 (Figure 4.4).

4.4 Partition of DNase and MNase data

I applied SPar-K with $K = 3$ to DNaseI accessibility profiles (2bp resolution) around 7'206 ChIP-seq SP1-binding peaks (+/-300bp relative to peak summit) in K562 cells (Figure 4.5A). The results revealed the presence of clear footprints in all the clusters (Figure 4.5B). To validate these footprints, I checked whether they were consistent with the location of nucleosomes (Figure 4.5C) and SP1 binding motif occurrences (Figure 4.5D), which was indeed the case. A *de novo* motif analysis of the narrow footprints seen in Figure 4.5B with MEME-ChIP and Tomtom (Bailey et al., 2009) revealed SP1-related, NFYA/B and GATA motifs (Figure 4.5G) the latter two reportedly being interaction partners of SP1. Taken together, these results suggest that SPar-K is able to precisely refocus initially misaligned DNaseI profiles around SP1 binding sites.

The partitioning of SP1 binding regions revealed distinct chromatin landscapes. Cluster 1 (red) groups binding sites lying between two closely spaced nucleosomes. Cluster 2 (blue) showed a strong asymmetry suggestive of promoter regions, an interpretation supported by the presence of TSSs indicative of promoters and of CAGE tags (Figure 4.5E and F). Finally, the symmetrical cluster 3 (green) contained binding sites located on a large nucleosome-free regions reminiscent of enhancer regions.

As a second example, I ran the same type of analysis on nucleosome profiles around CTCF binding sites (Figure 4.6). Overall, the results confirm observations from Chapter 3 and published in (Kundaje et al., 2012). Strong nucleosome arrays became visible in all classes

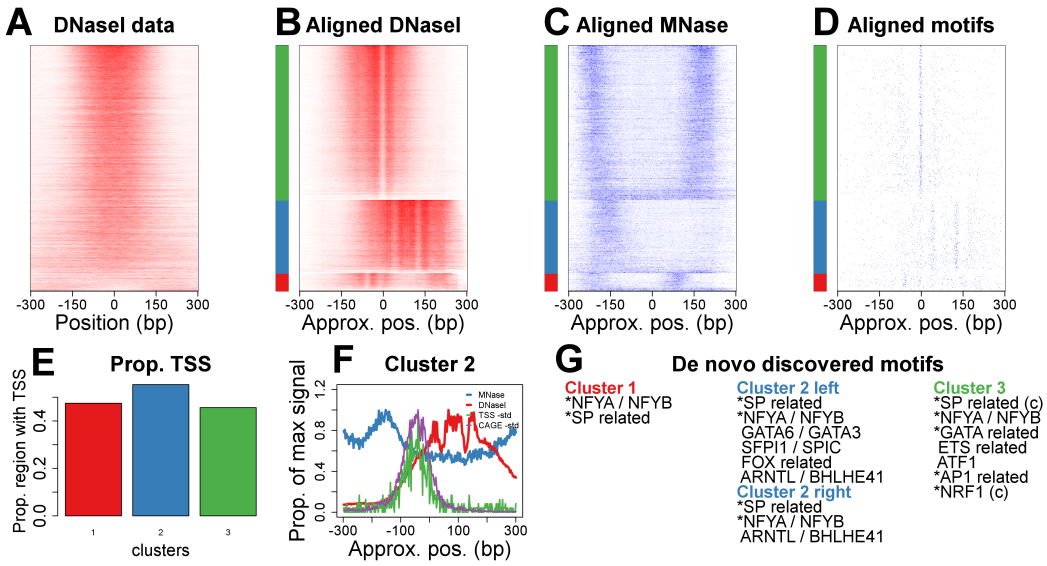


Figure 4.5 – Partitioning of DNaseI hypersensitivity profiles around SP1 binding sites in K562 cells. The optimal number of clusters was determined using the elbow method. **A.** Input data based on peak summits provided by ENCODE. **B.** Same regions clustered, re-aligned and oriented by SPar-K. Clusters 1, 2 and 3 are indicated by colored bars in red, blue, and green, respectively. **C.** MNase-seq read densities for the same regions, ordered, aligned and oriented as in B. **D.** Predicted SP1 motif occurrences for the same regions, ordered, aligned and oriented as in B. **E.** Proportion of binding sites within each cluster having a confirmed promoter-associated TSS within +/- 300bp. **F.** Aggregations profiles for DNase-seq (red), MNase-seq (blue), promoter TSS (green) and CAGE-seq data (violet) for cluster 2 (aligned and oriented as in B). **G.** Motifs found by MEME-ChIP and Tomtom in the narrow footprints of each cluster. (*) known SP1 interactor, (c) central enrichment. Cluster 2 left and right refer to the left and right footprints seen in **B**. Figure and legend taken and adapted from (Groux and Bucher, 2019).

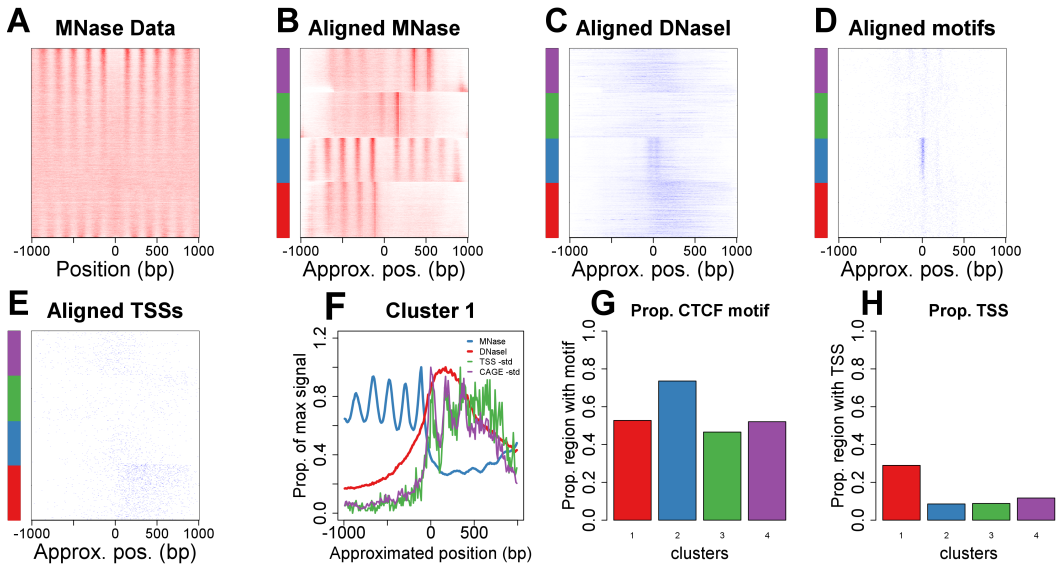


Figure 4.6 – Nucleosome occupancy, determined by MNase-seq, in bins of 10bp, +/- 1000bp around 79'957 CTCF binding sites in GM12878 cells. **A** MNaseI-seq read density around the CTCF binding sites. ChIP-seq peak summits are aligned at position 0. The regions (rows) are ordered according the their resemblance (correlation) to the overall aggregation pattern. **B** SPar-K data partition. The number of clusters (4) was determined using the elbow method. The cluster labels are indicated by the color ribbons on the left. Within each cluster, the data have been realigned according to the shift and flip informations returned by SPar-K and the regions have been ordered according the their resemblance (correlation) to the cluster aggregation pattern. Because of the realignment, ChIP-seq peak summits are not anymore aligned at position 0. **C** Corresponding DNaseI hypersensitivity measured by DNaseI-seq at the same loci and realigned as in B. **D** CTCF motif occurrences predicted using a motif scan, at the same loci and realigned as in B. Each predicted binding site, +/- 1kb around a peak, is represented as a point. **E** Transcription start site (TSS) density at the same loci and realigned as in B. **F** Cluster 1 (red) aggregation profiles. The original peak coordinates were modified accordingly to the shift and flip values returned by SPar-K and the read densities the different data types were measured using ChIP-Cor (Ambrosini et al., 2016a). For the TSSs and the transcription initiation (CAGE), only the data mapping on the negative strand were used to monitor transcription firing towards the nucleosome array (towards the left). **G** Proportions of regions having at least one CTCF motif occurrence +/- 1kb (same motifs as in D), for each cluster. **H** Proportions of regions having at least one TSS +/- 1kb (same TSSs as in E), for each cluster. Figure and legend taken and adapted from (Groux and Bucher, 2019).

after realignment, with three out of four showing strong asymmetry in addition.

4.5 Conclusions

SPar-K is a useful partitioning method for moderately misaligned and randomly oriented chromatin regions. Compared to existing methods, it is competitive in terms of accuracy, superior in speed, applicable to a wider range of input signals (not restricted to count data) and easy to use.

5 SMiLE-seq data analysis

The following section contains work made in collaboration with Alina Isakova from Prof. Bart Deplancke research group at EPFL and published in Isakova et al. (2017). I personally made the presented figures and analyses, with the exception of Figure 5.1.

5.1 Introduction

Deciphering TF binding specificity is key to understand the regulation of gene expression. Several technologies exist to study TF specificity in vitro such as mechanically induced trapping of molecular interactions (MITOMI Maerkl and Quake (2007)), protein binding-microarray (PBM Berger and Bulyk (2009)) or hight throughput systematic enrichment of ligangs by exponential enrichment (HT-SELEX Zhao et al. (2009); Jolma et al. (2010)).

Because TFs can bind as monomers, homodimers, heterodimers and as higher order complexes, it is critical to have suited technologies to interrogate their binding specificity. Nonetheless, because of combinatorials, this undertaking represents a staggering amount of work. In order to allow robust and easy measurement of TF monomers and dimers sequence specificity, over a wide range of binding affinities, Prof. Bart Deplancke research group at EPFL developed the selective microfluidics-based ligand enrichment followed by sequence (SMiLE-seq, Isakova et al. (2017)). An overview of the SMiLE-seq data production and processing procedures is shown in Figure 5.1. Overall, three major conceptual features should be highlighted. First, to tackle the scalability issue, the SMILe-seq plateform core is microfluidic device that contains hundreds of individual wells allowing to run as many different TF-DNA interaction assays at the same time. Second, the SMiLE-seq assay does not perform an iterative enrichment, as HT-SELEX does. SMiLE-seq is a one round selection system. Third, because TF-DNA interactions happens over a wide range of affinities, the weakest interaction can be lost during washing. In order to prevent this, complexes are protected under a membrane during the washing step.

In order to assess the quality of this new technology and its ability to produce relevant data to

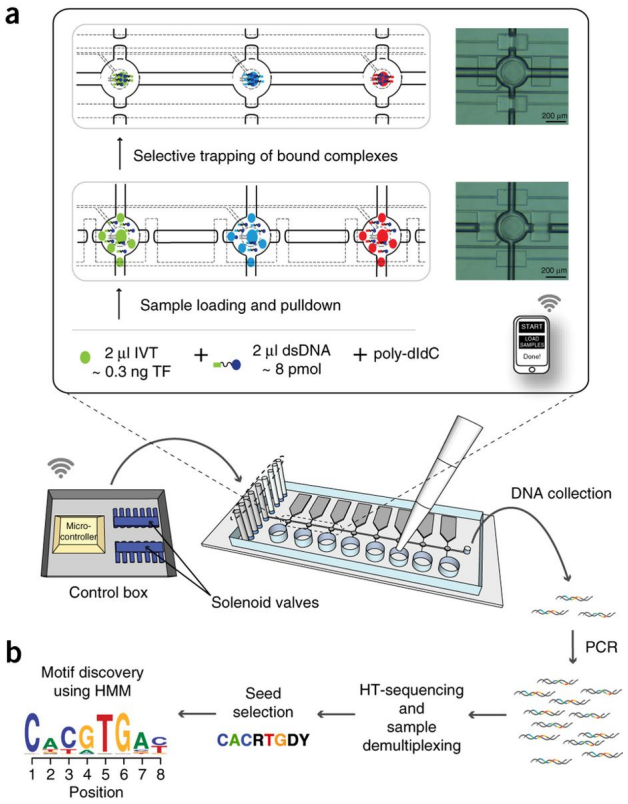


Figure 5.1 – SMiLE-seq pipeline : a Schematic representation of the experimental setup. A snapshot of three units of the microfluidic device is shown. In vitro transcribed and translated bait TF, target double-stranded DNA, and a nonspecific competitor poly-dIdC are mixed and pipetted in one of the wells of the microfluidic device. The mixtures are then passively pumped in the device (bottom panel). Newly formed TF–DNA complexes are trapped under a flexible polydimethylsiloxane membrane, and unbound molecules as well as molecular complexes are washed away (upper panel). Left, schematic representation of three individual chambers. Right, corresponding snapshots of an individual chamber taken before and after mechanical trapping. **b** Data processing pipeline. The bound DNA is eluted from all the units of the device simultaneously and collected in one tube. Recovered DNA is amplified and sequenced. The sequencing reads are then demultiplexed, and a seed sequence is identified for each sample. This seed is then used to initialize a probability matrix representing the sequence specificity model for the given TF. The model parameters are then optimized using a Hidden Markov Model-based motif discovery pipeline. Figure and legend taken and adapted from (Isakova et al., 2017).

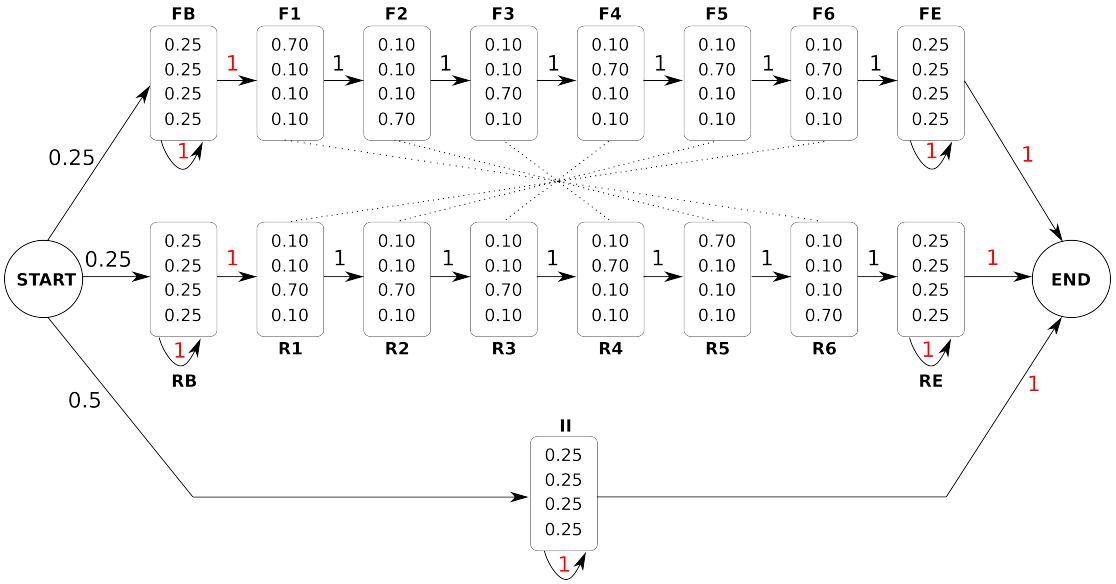


Figure 5.2 – Example of a Hidden Markov model : initial HMM representation with a seed sequence 'ATGCC'. The upper Markov chain models + strand motif containing sequences, the middle one - strand motif containing sequences and the lower zero motif occurrence sequences. The FB, FE, RB and RE positions represents positions in the sequence that occur before and after the binding site on the forward and reverse strand. For these nodes, a self transition exist to allow the binding site to occur at a variable distance from the beginning and the end of the sequence. Once transiting toward the 1st position of the binding site, the next transition is forced toward the 2nd position in the binding site, and so on until the end of the binding site. The + strand and - strand Markov chains emission parameters are paired together (they have the same values), as represented by the grey dashed lines. The transition probabilities in red are not subjected to the Baum-Welch training. Finally, a binding model represented as a probability matrix is composed of the emission probabilities at the binding site positions. Figure and legend taken and adapted from (Isakova et al., 2017)

model TF binding, I ran *de novo* motif discovery analyses on these data and compared their predictive performances with similar models derived from other in vitro and in vivo assays.

Our laboratory developed a Hidden Markov Model (HMM) based motif discovery method and a binding motif evaluation tool. Both are available on the laboratory web portal http://ccg.vital-it.ch/pwmtools/. My involvment in this project was to develop a scalable and efficient pipeline, at the backend of our server, to run the *de novo* motif discovery and benchmark steps and to analyze the results.

5.2 Hidden Markov Model Motif discovery

The *de novo* motif discovery method is based on an initial work performed by Philipp Bucher and Giovanna Ambrosini. This method was considered reliable and accurate. It participated

to the DREAM5 TF-DNA Motif Recognition Challenge (Weirauch et al., 2013) and was awarded the 3^{rd} place. Additionally, this method this method was already implemented and available in the lab. In consequence, it required a minimal setup effort. Also, the algorithm is somewhat similar to MEME, as described below. For these reasons, this method was selected for the analyses. Finally, the choice of training mono- rather than di-nucleotide models was motivated by the fact that we wanted to asses the SMiLE-seq technology by comparing the performances of the SMiLE-seq derived models with models derived from competitor assays. Because the competitor models were mono-nucleotide models, training more complex models from the SMiLE-seq data would have resulted in biased comparisons.

This motif discovery method models the DNA sequences using an HMM. The input is composed of N sequences of length L and the motif is represented as a LPM $\mathbf{M}$ of dimensions $4 \times L'$ where $L' \leq L$ and with the constrain $\sum_{i=1}^{4} m_{i,j} = 1$.

The sequences are modeled as a mixture of a set of consecutive positions belonging to the binding site (to which the TF binds) and of positions outside the binding site, which, is close to what MEME does (Bailey and Elkan, 1994). Since the position of the binding site in each sequence is unknown, they have to be guessed in order to align the binding sites and compute the LPM. To this end, we used an HMM that can handle the hidden information about the position of the binding site within a longer sequence. The emission parameters are the base probabilities at each position inside the binding site.

Additionally, to account for experimental biases, such as unspecific binding, a sequence can bear zero binding site. The entire model is composed of three Markov chains representing each path. Because a motif occurrence can appear on the + or the - strand of the DNA sequence, the modeling of the binding site is done by two paired Markov chains. The modification of a parameter in one of these two paired chains is propagated to the equivalent parameters in the other paired chain. An example of HMM is displayed in Figure 5.2.

The parameter estimation given the data is performed using the Baum-Welch algorithm. Mamot (Schütz and Delorenzi, 2008), a dedicated computational framework for HMMs, is used to handle all the computations.

Because, the Baum-Welch algorithm performs an iterative optimization of the model parameters, it needs a starting state. The motif length and the starting emission parameters are estimated using an over-represented kmer analysis. The motif length was set to the best kmer length and the starting emission probabilities are set to 0.7 for the base present in the kmer and 0.1 for the three others.

5.3 Binding motif evaluation

To evaluate and compare different binding models originating from different libraries - computed with different algorithms and data - with the SMiLE-seq data derived binding models,

I used PWMEval-ChIP-peak (formerly named PWMEval), a program using a methodology proposed by Orenstein and Shamir (Orenstein and Shamir, 2014) that has been developed in-house and is available at https://ccg.epfl.ch/pwmtools/pwmeval_chippeak.php. In order to run massive batch analyses, I developed a dedicated pipeline that was run at the backend of our servers.

PWMEval takes as input a set S^{pos} of N' experimentally validated binding sites sequences of length L'' - extracted from ChIP-seq peaks - and a set S^{neg} of N' presumably non-binding-sites of length L'' - extracted 300bp downstream of the corresponding peak sequences - and a probability matrix $\mathbf{M}$ describing the binding motif.

Each positive set sequence S_i^{pos} is scored using:

$$score_i^{pos} = \sum_{l=1}^{L''-L'+1} \prod_{k=1}^{L'} \frac{m_{b,k}}{p_b}$$
$$\text{with } b = \begin{cases} 1 & \text{if } s_{i,l+k-1}^{pos} = \text{A.} \\ 2 & \text{if } s_{i,l+k-1}^{pos} = \text{C.} \\ 3 & \text{if } s_{i,l+k-1}^{pos} = \text{G.} \\ 4 & \text{if } s_{i,l+k-1}^{pos} = \text{T.} \end{cases} \tag{5.1}$$

The same is done for each negative set sequence S_i^{neg}, resulting in the creation of two vectors of scores of length N' - one for each sequence set. Both vectors are concatenated into a unique vector of scores. Two vector of size N', containing the class labels (N' times 0 or 1) are created and concatenated as well. Then the vector of labels is sorted according to the decreasing sorting order of the score vector.

If the binding model allows to perfectly segregate the positive from the negative sequences, then we expect the positive sequence scores to be larger than the negative sequence scores. Thus the positive labels should all be at the beginning of the label vector and the negative labels at the end. The propensity of the model to recognize true binding sites from other sequences can be measured by computing the area under the curve (AUC) of the receiver operating characteristic (ROC). Given the list of sorted labels, the true positive and true negative discovery rates can be computed and the AUC-ROC can be computed as U/N^2 where U is the Mann-Whitney U statistic for the true positive and true negative discovery rates.

5.4 Results

In order to assess the robustness of SMiLE-seq, I derived binding models using the HMM motif discovery procedure for all TFs for which ChIP-seq peaks have been released by the ENCODE Consortium. For each ChIP-seq peak list, the 500 best peaks were selected based on their signal enrichment score (attributed by ENCODE). The peak score reflects the TF

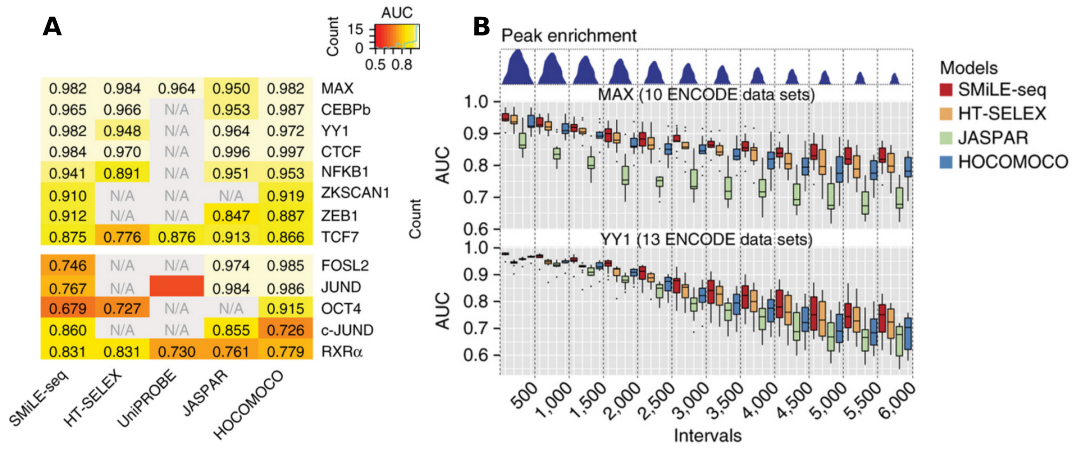


Figure 5.3 – Predictive power of SMiLE-seq : A SMiLE-seq detived motifs compared to that of previously reported motifs that are retrievable from the indicated databases. For each motif, the AUC-ROC values on the 500 top peaks of the ENCODE ChIP-seq data sets for the corresponding TF was computed. The heatmap represents the AUC values computed for each method on the respective ChIP-seq data sets that were selected based on the highest mean AUC values among all five models. **B** the predictive performances of MAX and YY1 binding models were assessed using subsets of binding sites of decreasing affinities. Inside each peak list, the peaks were ranked by score and subsets of 500 peaks were selected. Peaks 1-500 have the highest affinity, then peaks 501-1000, and so on. The boxplots indicate the distribution of AUC-ROC obtained over all available peak-lists. Figures and legends taken and adapted from (Isakova et al., 2017)

occupancy and likely the TF binding affinity to this site. Whenever several ChIP-seq peak lists were available for a given TF, the peak list achieving the highest mean AUC value over all models available for that TF was considered.

For each TF, LPMs equivalent to the SMiLE-seq derived LPM were selected from the HT-SELEX Jolma (Jolma et al., 2013), JASPAR 2014 (Mathelier et al., 2014) and HOCOMOCOv10 (Kulakovskiy et al., 2016) motif collections and were compared to the SMiLE-seq derived LPM using the AUC-ROC procedure procedure (Figure 5.3 A). This analysis reveled that, in the majority of the cases, the SMiLE-seq derived binding models were doing at least as good (MAX, CEBPb, CTCF, NFkB, ZSCAN1 TCF7, JunD, RXRa) or better (YY1, ZEB1) than the available models.

To verify that SMiLE-seq was indeed able to measure binding over a wide functional range of affinities, I computed AUC-ROC using decreasing affinity subsets of ChIP-seq peaks (Figure 5.3 B and Figure A.11B). For high affinity peak subsets, SMiLE-seq derived models scored as well as others but tend to become better than other models for lower affinity peaks subsets.

To further emphasize the quality of the SMiLE-seq technology and support conceptual differences with HT-SELEX, binding models were trained from HT-SELEX data using the HMM motif discovery procedure. The rational was that if the model performances could be attributed to different motif discovery methods, running the HMM motif discovery on these data should get us ride of this confounding factor. Because SMiLE-seq does not perform iterative enrichment of high affinity binders, 1st cycle HT-SELEX data were used. Here again, SMiLE-seq derived models were at least as good as HT-SELEX models (Figure A.11 A).

5.5 Conclusions

These results demonstrated that SMiLE-seq is a valuable plateform to run in-vitro TF-DNA binding assays. More specifically, this was demonstrated by showing that SMiLE-seq derived models are as good or better than the models derived from ChIP-seq data (such as HOCOMOCO or JASPAR models) or from in vitro data (HT-SELEX). Moreover, it was also possible to support the fact that SMiLE-seq was able to capture interaction over a wide range of affinities. This was shown through the use of differential binding affinity sites in the AUC-ROC analyses.

All together, these results support that SMiLE-seq is a valuable technology and a strong competitor for other in vitro TF-DNA interaction assays such as HT-SELEX.

6 PWMScan

In this chapter, I describe PWMScan, a software that has been developed to predict TF binding sites in a genome, given a binding specificity model. The problem of pattern-matching using a specificity model is considered an important problem and many algorithms and programs have been developed. The challenge comes from the fact that this problem should be solved in a time and memory efficient manner in order to allow large eukaryotic genomes to be scanned with complex patterns.

PWMScan is a web-server for rapid scanning of large genomes for high-scoring matches to a user-supplied or server-resident PWM. Compared to other web-based PWM scanning tools, PWMScan is unique in that it scans server-resident whole genomes rather than user-uploaded DNA sequences. Other key features are: i) menu-driven access to genomes of >30 model organisms, ii) menu-driven access to >300 public PWM libraries, iii) support of various PWM representations and formats, iv) cut-off values can be specified as match scores or P-values, v) output in BEDdetail format with match scores and P-values, vi) links to UCSC genome browser for visualization of results, and vii) action buttons to transfer match lists to analysis tools.

This work has been conducted in close collaboration with Giovana Ambrosini, a senior scientist of the laboratory. Some parts of this chapter have been taken and adapted - with the author permission - from the original article that presents this work (Ambrosini et al., 2018). For this project, I contributed to the development of the matrix_scan program, the development of the necessary Python code for an efficient parallelization of matrix_scan on the server backend and I set up and executed the entire benchmarking procedure. Furthermore, I produced all the figures presented in this chapter.

6.1 Algorithms

From an algorithmic perspective, PWMScan is dual. It implements a regular genome scanner and also implements a novel approach that uses short read alignment to find PWM matches.

6.1.1 Scanner algorithm

The program matrix_scan implements a scanner (or sliding window) searching algorithm. It takes as input a DNA sequence of length L, a PWM of dimensions $L' \times 4$ where $L' \leq L$ and a threshold score t. For each possible offset $i = 1, 2, ..., L - L' + 1$, the sub-sequence $S_i = S_{i,i+1,...,i+L'-1}$ is given a score $score(S_i)$ using equation 1.5. If $score(S_i) \geq t$ then the sub-sequence S_i is predicted to be a binding site. This naive approach is $\theta((L - L' + 1) * L')$ in time. Even though this is not bad in terms of asymptotics analysis, many unnecessary computations are performed. Indeed, during the score computations for a sub-sequence S_i, it is possible to define a point after which the current score has become so bad that it will automatically turn to be $< t$. From that moment, the computations for the current sub-sequence S_i can be dropped and the next sub-sequence S_{i+1} can be tested.

This only requires to modify the PWM such that the maximum possible score achieved at each position is 0. To do so, the values on each column should be substracted the maximum for that column. The threshold score t should modified into t' by subtracting the sum of the column maxima. This scheme is called 'lookahead' (LA, Pizzi and Ukkonen (2008)). LA can further be improved by redefining the position scoring order. The optimal position scoring order is to start first by the position in the PWM which can achieve the worst score, then by the PWM position achieving the second worst score and so one until the last PWM position that achieves the best possible score. By doing this, the score can drop under t' faster. This scheme is called permutated LA (PLA, Pizzi and Ukkonen (2008)). Even though PLA does not modify the algorithm complexity, in average this algorithm is faster than the naive one as it can avoid unnecessary computations.

6.1.2 Matches enumeration and mapping

The second approach used by PWMScan is drastically different. It also takes as input a DNA sequence S of length L, a PWM of dimensions $L' \times 4$ where $L' \leq L$ and a threshold score t. Then, all the k-mer of length L' achieving a score higher than t are enumerated and mapped against S using a short read mapper - in this case Bowtie (Langmead et al., 2009).

Enumerating all possible L-mer using a naive algorithm is $\theta(4^{L'})$ in time, which really quickly becomes cumbersome to handle. However, here again, many unnecessary computations can be avoided.

Let us assume a LA scheme as before. When scoring a sub-sequence S_i from left to right, the score monotonically decreases. If at some position j the score falls under the threshold score t' then the sub-sequence $S_i = S_{i,i+1,...,i+L'-1}$ is guaranteed to be rejected. By extension, this means that any sub-sequence that starts with a prefix $S_{i,i+1,...,i+j-1}$ are also guaranteed to be rejected. If the k-mers are stored in a trie, then it would be sufficient to ignore the sub-tree of $S_{i,i+1,...,i+j-1}$.

The program mba (for branch and bound) implements thisk-mer enumeration strategy. The

k-mers are enumerated by alphabetical order which is the same as exploring the k-mer trie in breadth-first order. If a sub-sequence S_i is rejected by dropping the computations at step j, the program jumps to the next sub-sequence $S_{i'}$ that does not start with the same prefix and continues the enumeration here. This is equivalent to ignore the entire prefix sub-tree in the trie.

Eventually, the enumerated k-mers are all the possible different matches that can be achieved with the given PWM and threshold score. Their locations in the genome are identified by a read mapping procedure using Bowtie, with S as the reference genome.

6.2 PMWScan architecture

PWMScan architecture and design is presented in Figure 6.1.

Several genomes are supported by PWMScan and are offered to the user on the web interface. The different specie genome sequences were downloaded from the NCBI in FASTA format and indexed for rapid scanning using Bowtie (Langmead et al., 2009).

PWMScan web interface offers several PWM collections to scan the available genomes. Each collection was downloaded from the MEME Suite website (Bailey et al., 2009) and all matrices, of any type, were converted to integer PWMs (see Supplementary Material in (Ambrosini et al., 2018)).

For all matrix formats, the cut-off value can be specified as a PWM score, as a percentage of the score range (0% = minimal score and 100% = maximal score) or as a p-value. The p-value of a PWM score X is defined as the probability that a random k-mer sequence of the same length as the PWM has a binding score $\geq X$ given the base composition of the genome. In the two latter cases, the corresponding cut-off value is computed.

It should be noted that IUPAC consensus sequences are also accepted instead of a matrix. In that case, they are converted into binary matrices consisting of 0 and 1. However, in this case, the threshold score is specified as the maximal number of mismatches allowed.

Once the appropriate format conversions have been applied to the inputs, the integer PWM and the corresponding cut-off score are given to the search engine. The choice between the scanner approach and the read mapping approach depends on the length of the PWM and the cut-off. As a matter of fact, the read mapping strategy is more efficient for short PWMs and high cut-off values. Indeed, in this case, the k-mers enumeration step is really fast. Empirically, we found that the limit was around 10^5 kmers. Above this limit, the scanner strategy is used. In this case, the individual chromosomes are processed in parallel with each chromosome sequence being distributed to a core by a Python script.

Eventually, the output is a list of sequence regions that match the PWM with a score higher or equal to the cut-off value. Post-processing of this list involves computation of the correspond-

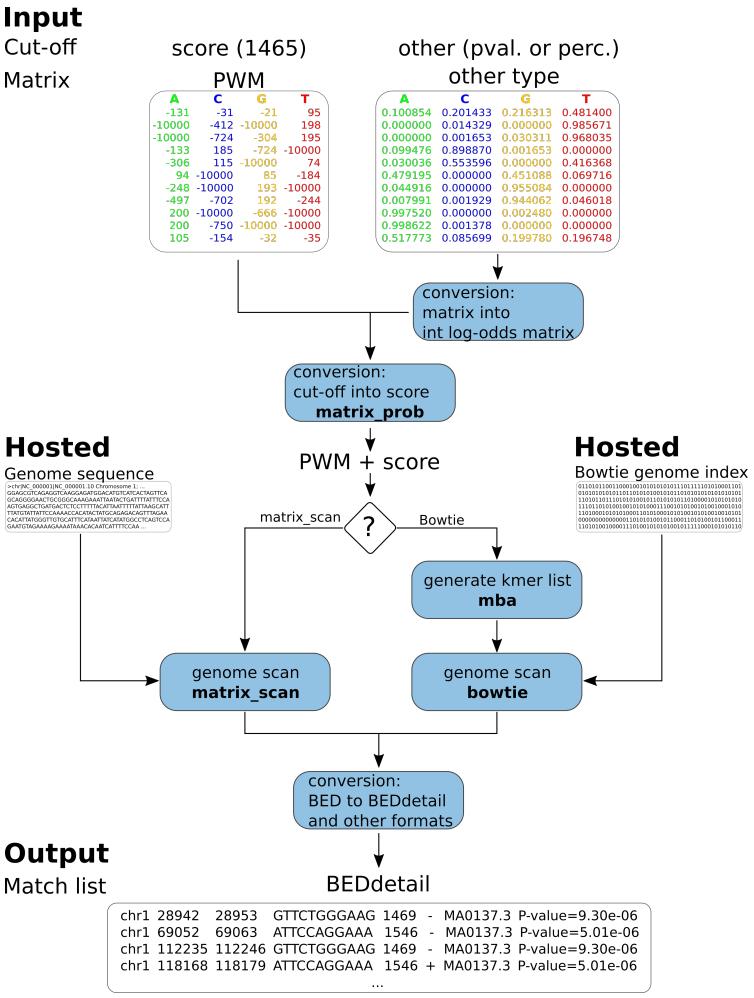


Figure 6.1 – PWMScan workflow : the input is composed of a PWM and a score threshold specifying the minimum score for a sequence to achieve to be considered as a match. LPMs or LFMs are also accepted and are converted into PWMs. The score threshold can also be given as a p-value or a percentage of the maximum score, in which case it is converted into a threshold score. Based on the length of the PWM, Bowtie or pwm_scan can be used to find the matches on the genome. If Bowtie is used, the set of k-mers achieving a better score than the threshold score is computed using branch-and-bound algorithm (mba) and mapped on the genome. On the other hand, if matrix_scan is used, the PWM is used to score every possible sub-sequence in the genome. The regions corresponding to the sequences achieving a score at least as good as the threshold score are then returned under BED format. Figure and legend taken and adapted from (Ambrosini et al., 2018).

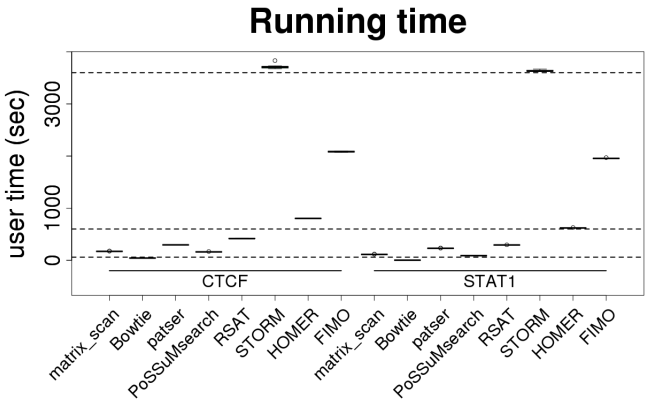


Figure 6.2 – Benchmark : PWMScan speed performances were measured and compared with 6 other well known genome scanners. In all cases, the h19 genome sequence was scanned with a 19bp CTCF matrix and a 11bp STAT1 matrix, 10 times. The run times are represented as boxplots. For PWMScan, both pwm_scan and Bowtie strategies were run. Figure and legend taken and adapted from (Ambrosini et al., 2018).

ing p-values, addition of the matrix name and, optionally, elimination of overlapping matches. The final match list is provided in BEDdetail format.

On the web interface, PWMScan is interconnected with i) the ChIP-seq (Ambrosini et al., 2016a) and ii) the SSA (Ambrosini et al., 2003) servers which allows to proceed to downstream analyses. Additionally, different action buttons are present on the result page and allows to i) extract the DNA sequences around the matches, ii) send the match coordinates to UCSC genome browser for visualization and iii) liftover the match coordinates to other assemblies of the same or related species.

PWMScan is also available as a standalone software from SourceForge. This includes a master script scheduling all computational steps running during a web job.

6.3 Benchmark

Many algorithms have been developed to find matches in genome using matrices. The most straight forward approach to this problem is the scanner approach. This way of doing is implemented by matrix_scan, Patser (Hertz et al., 1990), PossuMSearch (Beckstette et al., 2006), RSAT matrix-scan (later on simply referred to as RSAT, (Turatsinze et al., 2008)), HOMER (Heinz et al., 2010) or FIMO (Grant et al., 2011). PoSSuMSearch implements both the use of LA scheme and the use of enhanced suffix arrays (ESA) to store the indexed genome and search matches in sub-linear time.

These programs also generally compute a p-value associated with each score. FIMO also performs an expensive false discovery rate (FDR) estimate in order to correct the p-values

Program	CTCF			STAT1		
	Nb of matches	Perc. overlap	Score corr.	Nb of matches	Perc. overlap	Score corr .
matrix_scan	70607	100	1	138531	100	1
Patser	70607	100	1	138531	100	1
PossumSearch	70607	100	1	138531	100	1
FIMO	70131	99.16	0.9989	134901	100	1
RSAT	70701	99.85	0.9999	134901	100	1
HOMER	70701	99.95	0.9999	134901	100	1
STORM	70704	99.83	0.9999	134901	100	0.9999

Table 6.1 – Motif scanning software comparison. The performances of matrix_scan were assessed by comparing how many of the regions returned by matrix_scan were also returned by other programs and if the region scores were comparable. For the percentage of overlap with the match list returned by matrix_scan, the shortest of the two lists always served as the reference (100%). For the score correlations with matrix_scan scores, the Spearman correlation was used. Table and legend taken and adapted from (Ambrosini et al., 2018).

for multiple testing. On its side, STORM (Schones et al., 2007) uses a database of regulatory sequences to count the expected number of occurrences of kmers to compute a p-value. STORM creates a so called (g,k) table in order to store the number of occurrences of all possible kmers of size k with a maximum gap of size g present in a sequence database. Thus, it can handled gaped matches.

Also, for completness, it is worth mentioning that other programs have been released, such as MotifScanner (Aerts et al., 2003), MotifViz (Fu et al., 2004) or TRED (Zhao et al., 2005). However they could not be included in the benchmark for diverse reasons. MotifScanner could only be used through a GUI client connected to a (offline?) server, making it unsable for batch analyses. MotifViz relied on deprecated 32bits libraries and TRED input sequence size was limited to 10kb.

To assess how these programs perform compared to each other, I considered the run-time efficiency aspect but also, as important, the match consistency between programs. In order to perform meaningful comparisons, each program options were set such that each program returned a number of matches as close as possible to PWMScan. A match was considered common between two programs if the starting positions reported were equal. FIMO, Patser, RSAT and HOMER positions were modified by '-1' to account for a constitutive shift with PWMScan reported positions. The proportion of overlap was then computed by dividing the number of common match by the length of the longest match list from either programs. Then, a Spearman correlation was computed between the scores of the matches common to both programs to assess the scoring consistency.

The runtimes of PWMScan and of other competitor programs were measured by scanning

the human genome (UCSC assembly hg19) with two different PWMs from JASPAR, STAT1 with length 11 bp and CTCF with length 19 bp, and different cut-off values expressed as p-values. Results are shown in Figure 6.2 and and Table 6.1. PoSSuMSearch was used with its sliding window strategy. To our surprise, PWMScan was the fastest of the method. On the other hand, STORM was by far the slowest. This may be caused by STORM's ability to handle gaped matches. PWMScan also achieved a high similarity in terms of matches with the other programs both in terms of overlap between matches than in terms of score similarity. The differences observed cannot be explained with certainty. They could be explained, for instance, by differences between integer and double arithmetic or by differences in the handling of corner cases between different programs.

6.4 Conclusions

We developed PWMScan, a program that searches matches inside a genome using a PWM. We implemented two alternative search strategies : i) a traditional sliding window approach and ii) an innovative match mapping using Bowtie. Both strategies are competitive in terms of speed and score similarity compared to other existing programs. Furthermore, our window sliding approach turned out to be fastest of all.

PWMScan is meant to support many types of genomic data analysis and designed to be interoperable with other tools from our group and elsewhere. An example of a typical workflow involving ChIP-seq data is presented in Supplementary Material in (Ambrosini et al., 2018). Additionally, PWMScan is available as a standalone distribution.

In short, PWMScan is an asset for the research community because of its simplicity of use and of its high performances.

7 Chromatin accessibility of monocytes

The chapter contains ongoing work. I present the bases of a computational framework to analyse chromatin organization around TF binding sites from ATAC-seq data. As a matter of fact, the results presented here are quite preliminary. However, in the best case, this may shape a basis for other projects.

7.1 Monitoring TF binding

As discussed above (see section 1.4.4), DGF assays are able to highlight active regulatory elements from an entire genome, at once. However, this comes at the price of an information loss. First, even if we can identify active loci likely to be bound by TFs, we have no direct information about the identity of the bound TF(s). Second, we have no idea about the function of those regions. These regions may act as transcriptional activator or repressor. This activity is ultimately bared by the TF and other complexes bound. Thus delineating a region function necessitates to identify the TFs bound here.

This task, even if difficult, can be undertaken by implementing dedicated strategies. First, it is possible to collect evidence about the identity of TF likely to bind at a given location through a motif analysis. TFs can bind DNA directly through their own DNA binding domain or indirectly, through an interaction with at least one other partner TF which binds DNA directly (Neph et al., 2012). For a given TF, direct binding events can be detected by monitoring the presence of a motif occurrence if a specificity model is available. Thus a footprint bearing a motif occurrence is likely to reflect a direct binding event. However, this method has two important limitations : i) related TF often share a common DNA specificity and ii) this does not detect indirect binding events. However, evidence about the presence of large complexes can be collected by studying the size of the footprint. Large complexes should leave large footprints. This approach, even if limited is able to pinpoint a handful of candidate TFs.

Second, it had been suggested that the regulatory function of a TF can be deciphered by looking at its footprint. Indeed, previous studies have shown that activator and repressor TFs

tend to produce different types of footprints (Berest et al., 2018). Also, the spatial positioning of TF motif occurrences within the footprint seemed to be linked with the factor functions (Grossman et al., 2018). For instance, factors associated with the regulation of transcription tend to have a motif occurrence in the middle of the footprint whereas factors known to interact with chromatin remodeling factors tend to have a footprint at the edge of the footprint, in contact with the surrounding nucleosomes.

7.2 The advent of single cell DGF

Recently, the advent of single-cell (sc) sequencing technologies have been a real game changer in the field of life science. These technological advances allowed to measure gene expression and chromatin accessibility (scATAC-seq) at a yet unprecedented resolution. As bulk sequencing was providing an average overview of what was going on, single-cell sequencing allows to monitor what is happening in each cell of a population. This advance had a profound impact on genomics for two reasons.

First, for the really first time, the heterogeneity of a cell population became accessible and could be studied at the chromatin, transcriptional and protein levels. Second, the possibility of collecting high dimensionality data from tenths of thousands of individual cells allows genomics to fully enter in the modern big data era, making commonly used machine learning methods usable as the number of parameters to estimate in the models became smaller than the number of individuals, in this case cells, in the data (Angerer et al., 2017).

7.3 Open issues

I identified two interesting question with regard to ATAC-seq data. First, in the previous chapters, I studied how chromatin is organized in the vicinity of TF binding sites using a pretty standard combination of ChIP-seq, DNase-seq and MNase-seq data. However, I wanted to assess to what extent the same could be done from ATAC-seq data which are cheaper and easier to produce. Second, I wonder to what extent single-cell data could be pooled together and used as a bulk sequencing experiment.

7.4 Data

To this end, I chose to work with a publicly available sc-ATAC-seq dataset from 5'000 human blood monocytes from a healthy donor. These data have been produced by 10xGenomics (https://www.10xgenomics.com).

10xGenomics is a company active in the field of sequencing technologies and data analysis softwares. To demonstrate the capabilities of their sequencing and bioinformatics analysis technologies, 10xGenomics offers a free access to several high quality single cell datasets

together with their analysis results. Thus pre-processing steps such as mapping, cell demultiplexing, sequencing adapters trimming, quality control checks have already been performed. Thus working with these data require minimum handling. Additionally, some downstream analyses such as peak calling or clustering have already been performed. For these reasons, their datasets offer all the conditions to be used as a standard to develop and benchmark new analyses methods.

Hg19 mapped reads were downloaded in bam format as well as the corresponding peaks, in bed format, called on the aggregated data.

7.5 Identifying over-represented signals

The study of signal shape (distribution) has been a quite active field for bulk sequencing experiments during the last decade. Dedicated algorithms (Hon et al., 2008) (Nielsen et al., 2012) (Kundaje et al., 2012) (Nair et al., 2014) (Groux and Bucher, 2019) have been developed to cluster genomic regions based on their distribution of reads. As discussed in section 1.6, the major issue faced are that i) to assess whether two regions are similar in terms of sequencing signal, they have to be properly aligned and ii) oriented and iii) there may be region-specific sequencing depth differences. Nonetheless, these algorithms allow to capture important trends in the data and to collect evidence about the underlying biological mechanisms at play.

7.5.1 ChIPPartitioning algorithm

ChIPPartitioning is an algorithm that has been developed by Nair et al. (2014) to classify regions based on their sequencing read densities and to identify archetypical sequencing densities (or models). Because the algorithm is already presented in details in section 3.2, it will not be discussed further here. Nonetheless, the reader is invited to read the above mentioned section in order to properly understand the points discussed below.

7.5.2 EMSequence algorithm

ChIPPartitioning algorithm presented an interesting feature : it explicitly models the sequencing signal. Thus it can be adapted to search for different types of signals, for instance nucleosomes architectures or footprints. But because footprints reflect the binding of TFs, it is also critical to be able to identify the motifs within. To this end, I modified ChIPPartitioning in order to discover over-represented sequence motifs. Let us called this new algorithm EMSequence. The following modifications have been applied to ChIPPartitioning i) how the class signal is modeled, ii) the way data likelihood are computed and iii) the update of the class models. This is illustrated in Figure 7.1B. With these modifications, EMSequence is *de facto* a *de novo* motif discovery algorithm.

Conceptually, this algorithm proposed is close to the EM algorithm proposed by Lawrence

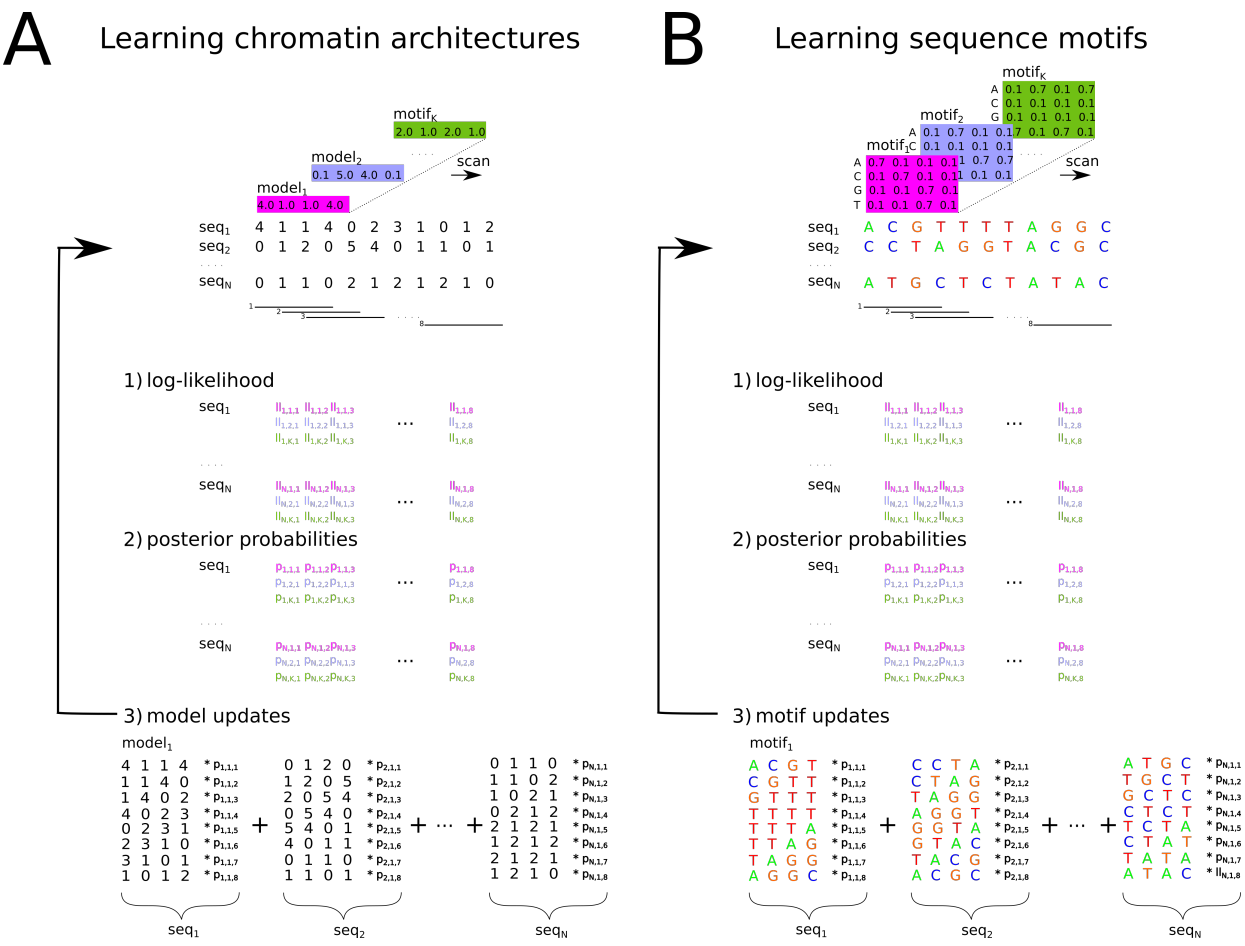


Figure 7.1 – Illustration of the expectation-maximization algorithms A illustration of ChIP-Partitioning, an algorithm dedicated to the discovery of over-represented chromatin patterns, as described in (Nair et al., 2014). **B** illustration of EMSequence, an algorithm to discover over-represented DNA motifs. The overall design is the same. Both algorithms model the data as having being sampled from a distribution and perform a maximum-likelihood estimation of the distribution parameters from the data through an iterative procedure.
EMJoint algorithm is the combination of both ChIPPartitioning and EMSequence at the same time.

and Reilly (Lawrence and Reilly, 1990) and to MEME (Bailey and Elkan, 1994), that are both designed for *de novo* motif discovery. The difference with the algorithm of Lawrence and Reilly is that EMSequence is generalized to find several classes instead of one. The difference with MEME is that EMSequence searches all of the different classes at once instead of using an iterative procedure.

The input is composed of an integer matrix D of dimensions $N \times L$ containing N DNA sequences $d_1, d_2, ..., d_N$ of length L. Each sequence $d_i = (d_{i1}, d_{i2}, ..., d_{il})$ is a vector of integers encoding it (A=1, C=2, G=3, T=4).

The K classes profiles from which the data originate, instead of being modeled as signal profile, are modeled as sequence motifs $M_1, M_2, ..., M_K$ of expected base probabilities (LPMs). A class motif M_j is a matrix of dimensions $4 \times L'$ with the constrain $\sum_{b=1}^{4} m_{jb,l} = 1$.

without shift and flip

For the case where $L' = L$, the original equation (1) from (Nair et al., 2014) to compute the probability of a sequence d_i given a class M_j is replaced by :

$$\begin{aligned} P(d_i|m_j) &= \prod_{l=1}^{L} m_{jb,l} \\ \text{where } b &= d_{il} \end{aligned} \tag{7.1}$$

Once the posterior probabilities $P(M_j|d_j)$ have been computed using equation 1.7, the original equation (3) in (Nair et al., 2014), to update the class models, is modified as follows :

$$\begin{aligned} m^*_{jb,l} &= \frac{\sum_{i=1}^{N}(P(M_j|d_{il}) \times z}{\sum_{k=1}^{N}(P(M_j|d_{il})} \\ \text{with } z &= \begin{cases} 1, & \text{if } b = d_{il}. \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{7.2}$$

where M^*_j is updated model of class j and b takes the values $1, 2, 3, 4$ for A, C, G and T respectively.

with shift and flip

For the sake of generality, I present the case with shift and flip because cases with shift only or flip only are special cases with shift and flip.

For the case with shifting ($L' < L$) and flipping, the original equation (9) from (Nair et al., 2014) to compute the probability of a sub-sequence of length L' starting at offset s in sequence d_i

given class M_j is replaced by :

$$P(d_i|M_j,s,inv) = \prod_{l=1}^{L'} m_{jb,l}^{inv} \quad \text{with } b = \begin{cases} d_{i,s+l-1} & \text{if } inv = 1. \\ 4 - d_{i,s+l-1} + 1 & \text{if } inv = 2. \end{cases} \tag{7.3}$$

where inv is a notation indicating the orientation. If $inv = 1$ we are searching in forward orientation (the forward strand) and $M_j^1 = M_j$. If $inv = 2$, we are searching in flipped orientation (reverse strand) and M_j^2 is the reverse complement motif of M_j. Computing the reverse complement motif M_j^2 of a class motif M_j is done using :

$$m_{ji,l}^2 = m_{j4-i+1,L'-l+1} \tag{7.4}$$

The computation of the posterior probabilities $P(M_j,s,inv|d_j)$ remains the same as in (Nair et al., 2014). With the posterior probabilities, the model update can be undertaken. The original equation (12) in (Nair et al., 2014) should be modified. The update of the model is made in 2 steps: i) by creating an intermediate motif for each strand separately and then ii) by combining them, as follows :

$$\begin{aligned} m_{jb,l}^{*1} &= \sum_{s=1}^{S}\sum_{i=1}^{N} P(M_j,s,inv=1|d_{il+s-1}) \times z^1 \quad \text{with } z^1 = \begin{cases} 1, & \text{if } b = d_{il+s-1}. \\ 0, & \text{otherwise.} \end{cases} \\ m_{jb,l}^{*2} &= \sum_{s=1}^{S}\sum_{i=1}^{N} P(M_j,s,inv=2|d_{il+s-1}) \times z^2 \quad \text{with } z^2 = \begin{cases} 1, & \text{if } 4-b+1 = d_{il+s-1}. \\ 0, & \text{otherwise.} \end{cases} \\ m_{jb,l}^{*} &= \frac{m_{jb,l}^{*1}}{\sum_{b'=1}^{4} m_{jb',l}^{*1}} + \frac{m_{j4-b+1,L'-l+1}^{*2}}{\sum_{b'=1}^{4} m_{j4-b'+1,L'-l+1}^{*2}} \end{aligned} \tag{7.5}$$

where m^{*1} is the partial motif update for the forward strand, m^{*2} is the partial motif update for the reverse strand and b takes the values $1,2,3,4$ for A, C, G and T respectively.

As in the original algorithm, the optimization process is then carried on for a given number of iterations.

7.5.3 EMJoint algorithm

For completeness, I also describe a generalized EM algorithm that performs the classification of a set of regions using several different signal layers over these regions, at once. This algorithm has been implemented but was not tested as time did not permit it.

Because ChIPPartitioning and EMSequence algorithm computations are strictly identical with the exception of the likelihood computations and the model update, it is possible to design a third algorithm, called EMJoint, that models at the same time one or more read coverage signals over a region and its sequence composition. To do so, I simply mixed both previous algorithms and applied the following modifications. For the sake of simplicity, I only expose the version with shift and flip, for one read coverage signal and the DNA sequence, as it is the most general.

The input is composed of two matrices of integers, D and R, of dimensions $N \times L$. N DNA sequences $d_1, d_2, ..., d_N$ of length L and of N vectors of read counts $r_1, r_2, ..., r_N$ of length L are contained inside each matrix respectively. Each DNA sequence $d_i = (d_{i1}, d_{i2}, ..., d_{il})$ is a vector of integers encoding the DNA sequence (A=1, C=2, G=3, T=4) and each read count vector $r_i = (r_{i1}, r_{i2}, ..., r_{il})$ is a vector of integers containing the number of reads mapping over the sequences contained in D.

The positions represented by each cell of the matrices R and D must be strictly the same. That is, if $D_{10,74}$ represents the base at position 12'342'457 of chromosome 9, $R_{10,74}$ must contain the density of reads that are mapped at position 12'342'457 of chromosome 9.

Each class is modeled by a vector of length L' of expected number of reads $C_j = (c_{j1}, c_{j2}, ..., c_{jL'})$ and by a sequence motif M_j of expected base probabilities M_j of dimensions $4 \times L'$ with the constrain $\sum_{b=1}^{4} m_{jb,j} = 1$.

To compute the likelihood $P(r_i, d_i, s, inv | C_j^{inv}, M_j^{inv})$ of a region, equation 7.1 is modified as follows :

$$P(r_i, d_i, s, inv | C_j^{inv}, M_j^{inv}) = \prod_{l=1}^{L} Poisson(r_{i,l}, \lambda = c_{j,l}^{inv}) \times m_{jb,l}^{inv}$$
$$\text{with } b = \begin{cases} d_{i,s+l-1} & \text{if } inv = 1. \\ 4 - d_{i,s+l-1} + 1 & \text{if } inv = 2. \end{cases} \tag{7.6}$$

where λ is the mean parameter of the $Poisson$ probability mass function.

The posterior probability $P(C_j, M_j | r_j, d_j)$ computation remain unchanged. Once these values have been computed, it is possible to update both part C_j and M_j of a class using the original equation (11) from (Nair et al., 2014) and equation 7.5 respectively.

It is possible to further generalize this algorithm in order for it to accept Z different input

matrices (called layers, at most 1 DNA sequence matrix and any number of read density matrices) of dimensions $N \times L$ containing different types of signal (for instance DNA sequences, TF_1 ChIP-seq, TF_2 ChIP-seq, DNase-seq, ...) for a set of N regions.

This only requires to adapt how the classes are modeled and equation 7.6 to sum over the Z different layers instead of only two. Additionally, care should be taken to use equation 7.3 for DNA sequence layer and equation (6) from (Nair et al., 2014) for read count layers.

7.5.4 Data realignment

As for ChIPPartitioning, these algorithms compute a set of posterior probabilities and use them to perform the class model updates. Thus, each one of them can be used to partition a dataset A and relign another dataset B, using the same procedure as described in section 3.2.1 for ChIPPartitioning.

Furthermore, it is absolutely feasible to run a partitioning on a given matrix A, let us say of DNA sequences using EMSequence, and to subsequently use the obtained posterior probabilities to compute the class models using another data matrix B of ATAC-seq read counts. Care should only be taken to use the appropriate data model computation equation.

In the following sections, this is the procedure that will be used to overlay different types of data for a given partition.

7.5.5 Soft aggregation plots

Given a read density matrix R containing the signal of N regions of length L, a standard aggregation plot can be computed by averaging the signal present at each position over the rows. This results in the creation of a vector of length L that represents the average signal in R.

In essence, ChIPPartitioning class models are aggregation plots. A data partition into K classes created K models that are as many aggregation vectors. Each model contains the average expected signal of a class. It only differs from the above definition by the fact that a *weighted* average is computed. For a given class, each row is assigned a weight that is the (posterior) probability that this region belongs to this class. If shifting is enabled, then the aggregation vectors have a length $L' = L - S + 1$ where S is the shifting freedom.

Conceptually, EMSequence does exactly the same excepted that aggregating DNA sequences results in the creation of a LPM instead of a vector. Obviously, at a given position, one cannot average 'A' and 'G' together. Instead one can say that there are 50%A, 0%C, 50%G and 0%T.

In the following sections, soft aggregation plots simply correspond to ChIPPartitioning/EMSequence models.

7.6 Data processing

Prior undertaking the chromatin organization study several pre-processing steps and checks have to been taken in order to ensure a proper treatment of the data.

If a TF can protect a stretch of DNA against transposition and create a footprint, so can a nucleosome. As a matter of fact, both cases are biologically drastically different. Nucleosome compete with TFs to bind on DNA (Voss and Hager, 2014). Thus nucleosome footprints represent regions of the genome that cannot be bound by TFs, if we except pioneering factors (Cirillo et al., 2002; Zaret and Carroll, 2011). Mixing nucleosome and TF footprints could bias downstream analyses.

The sequenced fragments were split in two classes based on their sizes, following the method described in Buenrostro et al. (2013) (more details are available in section A.4.1). One class contained the small sized fragment that originate from accessible chromatin regions - later referred to as "open chromatin" - and the second class contained longer fragments mapping to individual nucleosomes - later referred to as "nucleosomes".

Then, I used the edges of the open chromatin reads to reveal chromatin accessibility and TF footprints and the middle position of the nucleosome fragments to reveal the histone octamer occupancy (more details in section A.4.2).

7.7 Results

To create a catalog of chromatin architectures around TF binding sites in monocytes, it was necessary to be able to align the regions of interest properly (with respect to the binding sites) or to have methods able to deal with this issue.

7.7.1 Aligning the binding sites

The list of active regulatory regions was assumed to correspond to regions of high ATAC-seq signal. Consequently, I chose to used the peak list generated by 10xGenomics for this dataset as the list of regulatory regions of interest. The regulatory regions overlapping known repeated elements were filtered out leaving 70'462 regulatory regions.

An assessment of ChIPPartitioning indicated that ChIPPartitioning did not seem to be able to realign regulatory elements - and reveal footprints - using their chromatin accessibility profiles at the single base resolution (see section A.4.3). However, EMSequence showed good performances to retrieve sequence motifs (see section A.4.3). Because footprints are expected to be located over TF binding motif occurrences, I decided to use EMSequence to realign the regions based on their sequence motif occurrences to reveal TF footprints.

In order to limit the scope of the investigation, I decided to focus on a priori important TFs

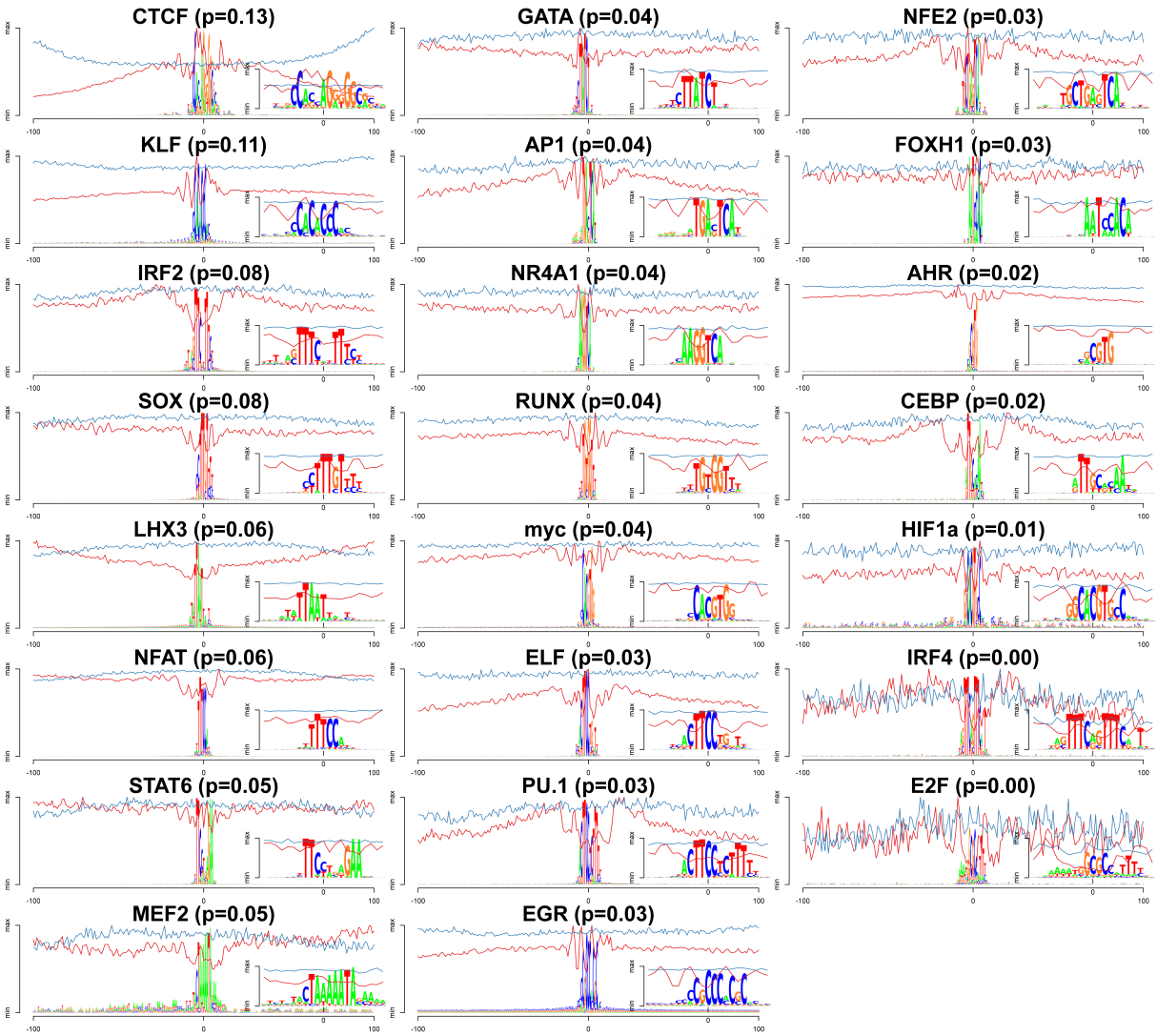


Figure 7.2 – Central parts of the extended sequence and chromatin models found in monocytes regulatory regions. Each read density and sequence pattern is a soft aggregation plot. The displayed logos correspond to the sequence class models found by EMSequence. The corresponding chromatin accessibility (red) and nucleosome occupancy (blue) are displayed atop of the logos. The classes are displayed by overall decreasing probability. A zoom over the central part of each class aggregation is shown in the top right inlet.

for the monocyte biology (Kurotaki et al., 2017; Rico et al., 2017). These TFs were : jun, HIF1a, myc, PU.1, CEBPB, IRF2, IRF4, IRF8, LHX3, FOXH1, SOX, MEF2c, ELF5, STAT6, NFE2, AHR, E2F2, E2F3, KLF2, KLF4 and NR4A1. Additionally, CTCF was added together with the EGR, GATA, NFAT and RUNX families to widen the spectrum of TF included in the analysis. Because TFs within a given family tend to bind the same motif (for instance IRF4 and IRF8 or E2F2 and E2F3), binding models representative for sets of TFs were selected from the JASPAR database motif clustering (Castro-Mondragon et al., 2017). In total, 23 LPMs were downloaded from JASPAR database clustering.

In order to realign the regions and reveal the TF footprints, the 70'462 sequences of 1'001bp centered on the regulatory element mid position were subjected to a special EMSequence partionining setup. EMSequence model parameters were initialized using the downloaded LPMs. Then EMSequence was run for 1 iteration. The rational was that EMSequence was not required to learn the models but instead to partition and align the sequences using these known models. A shifting freedom of 971 was allowed, resulting in the alignment of 30bp long sub-regions, together with flipping freedom. Based on the alignment and data, the resulting 30bp ATAC-seq signal and sequence models were extended of 500bp on each side to reveal the organization of regulatory sequences. This created 23 soft aggregation plots revealing the sequence and chromatin architecture around each TF binding sites (Figures A.23 and 7.2).

First, from the class aggregations, footprints are clearly visible over the TFs binding motif. This is a strong evidence that the region realignment worked properly. Second the 23 different classes showed various types of footprints. For instance, CTCF shows its usual strongly positioned nucleosome arrays together with a clear chromatin opening over the motif occurrences, supporting CTCF binding. The important monocyte TF PU.1 also shows an increased chromatin accessibility at its binding sites. However, the footprint drastically differ from CTCF in the sense that a clear a wide signal drop - larger than PU.1 motif only - is visible. It is also concomitant with an increased nucleosome occupancy. Conversly, LHX3 shows a pattern that rather suggest a modest chromatin opening. Finally, KLF's family binding sites have a strong chromatin accessibility rather than a protection of the bound sequences, which may be compatible with a transcriptional repressive activity (Berest et al., 2018).

Third overall class probabilities gives an indication of the regulatory element content in term of motif occurrences. Its seems that CTCF motif occurrences are the most common even though it does not mean that each occurrence is bound or even functional.

7.7.2 Exploring individual TF classes

The results shown in the previous section are per TF aggregation profiles. Thus a further exploration of each class is required to investigate whether several different footprint classes can be isolated per TF. To do so, I extracted the data assigned to each class and run ChIPPartitioning on these data. Because the regions have already been aligned such that the motif occurrences were in their centers, ChIPPartitioning was not allowed shifting nor flipping.

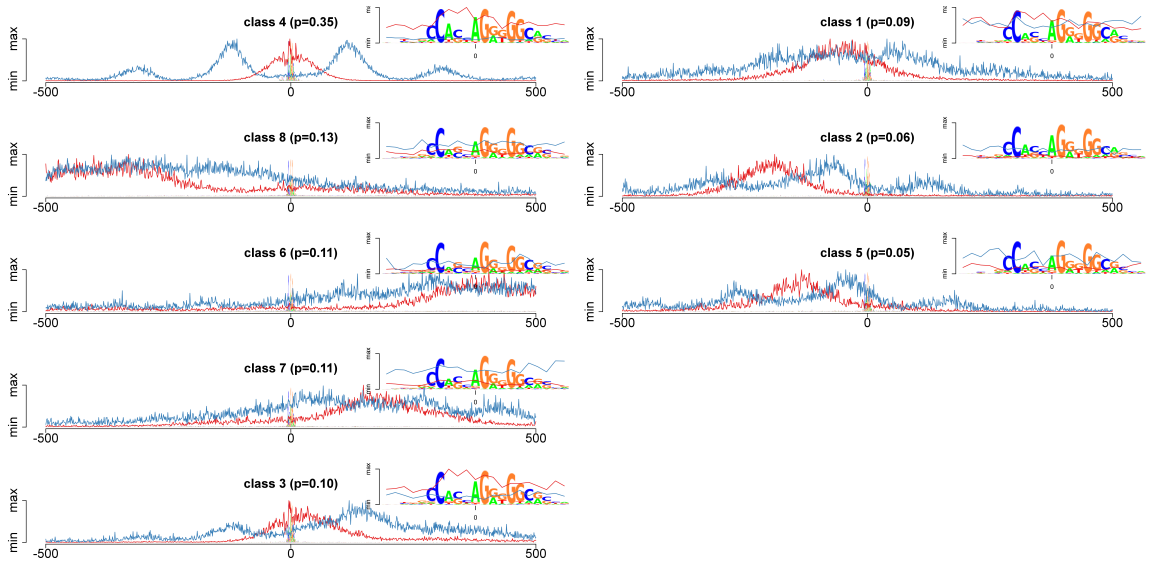


Figure 7.3 – Soft aggregation plots of CTCF sub-classes obtained by extracting CTCF class data and subjecting them to a ChIPPartitioning classification into 8 classes. The displayed logos correspond to each class sequence aggregation. The corresponding chromatin accessibility (red) and nucleosome occupancy (blue) are displayed atop of the logos. The classes are displayed by overall decreasing probability. A zoom over the central part of each class aggregation is shown in the top right inlet.

As expected, applying this method refined the results. For instance, the CTCF class data classification (Figure 7.3) showed sub-classes in which CTCF motif occurrences were likely not bound (sub-classes 8, 6, 7, 2 and 5) as well as sub-classes in which they were likely bound (sub-classes 4, 3 and 1). In the latter group, several chromatin organizations could be revealed, with approximately 35% of the motif occurrences showing the canonical CTCF chromatin organization (class 4).

The same is illustrated for PU.1 and AP1 classes (Figure A.24 and A.25). In both cases, it was possible to identify likely bound and unbound motif occurrence sub-classes. Also, for these two TFs, the nucleosome are not visible, in line with my previous results showing that only CTCF has nucleosome arrays organized with respect to its binding sites (see Chapter 3).

7.8 Discussions

Even though preliminary, these results showed that this computational framework can turn useful to analysis the chromatin organization around TF binding sites using ATAC-seq data.

First, not much of a surprise, applying population level analyses to the pool of single cell data gave meaningful results.

Second, ChIPPartitioning turned out to be useless to properly phase unaligned regions based on their chromatin accessibility patterns. Instead, the newly proposed EMSequence algorithm turned out to be usable for this task, in a special setting, and was able to produce a meaningful per TF data realignment. As a reminder, short models were searched (and thus large shifting freedom was set). This alignment was then used to realign larger regions and revealed footprints. Also, a priori knowledge was fed in under the form of the initial sequence model values taken from TF binding model databases.

Third, I presented a method to extract data assigned to a class, from a probabilistic partition, without using any hard assignment shortcut. Running ChIPPartitioning on these data then turned out to revealed different chromatin organization for each TF, allowing to distinguish between likely bound and unbound motif occurrences.

As a fact, in its current form, this framework is incomplete and many things could be modified or included.

As an immediate algorithm improvement, I propose to modify EMSequence. Currently, it models the DNA sequences using LPMs. LPMs have many advantages but on the other side they imply a strict order of occurrence. For instance, a 50bp long matrix can represent two sub-motifs of 10bp each, separated by 30bp. Consequently, it cannot handle a case where the order of occurrence of the sub-motifs would be inverted or have a different spacing differently than by dedicating a class to this. To circumvent this, I propose to handle the DNA sequence content using a list of k-mers. K-mers would have two immediate benefits. First, it would alleviate the ordering constrain. Any k-mer can occur before or after any other. The only

thing that matters is whether the k-mers are present or not in the sequence. Second, the use of k-mers would imply to "bin" the sequences. For instance, 10-mers would split the sequences in bins of 10bp. This point is of interest for EMJoint which use jointly EMSequence and ChIPPartitioning and currently require bins of 1bp if a DNA sequence matrix is used. The usage of bigger size bins has the following advantages : i) it will smooth the read densities for ChIPPartitioning leading to higher classification accuracy, ii) it will reduce the memory requirements (the data dimensions diminishes) and thus iii) it will reduce the runtime. On the other hand, storing k-mers can be a burden. Storing all possible kmers is $\Theta(K * 5^S)$ (including 'N's) in memory where K is the number of classes and S the kmer length. But this can be changed to $\mathcal{O}(N * (L - S + 1))$ where N is the number of sequences and L the sequence length, using hashtables, which scales better for high values of K, S, N and L.

Additionally, a method to estimate the fit of a given partition and choose the best one would be an asset. This would help choosing the appropriate number of classes to search. This could be implemented using the Akaike information criterion (Marsland, 2015).

7.9 Methods

7.9.1 Code availability

All the code used in this project is available on a c4science git repository (https://c4science.ch/source/scATAC-seq.git).

7.9.2 Data sources

The reads mapped to hg19 genome were downloaded, in bam format, from 10xGenomics website (http://s3-us-west-2.amazonaws.com/10x.files/samples/cell-atac/1.1.0/atac_v1_pbmc_5k/atac_v1_pbmc_5k_possorted_bam.bam). The corresponding peaks called on the aggregated data were downloaded, in bed format, from http://cf.10xgenomics.com/samples/cell-atac/1.1.0/atac_v1_pbmc_5k/atac_v1_pbmc_5k_peaks.bed. A file containing cell barcode related information was downloaded from http://cf.10xgenomics.com/samples/cell-atac/1.0.1/atac_v1_pbmc_5k/atac_v1_pbmc_5k_singlecell.csv and the barcode sequences extracted using "grep -E _cell_[0-9]+ <file_csv> | cut -d ',' -f 1" where <file_csv> is the downloaded file.

The hg19 genome sequence was downloaded from the Ensembl ftp at ftp://ftp.ensembl.org/pub/grch37/current/fasta/homo_sapiens/dna/. Chromosome 1, 2, ..., 22, X and Y sequences were downloaded in fasta format and concatenated together. The sequence headers were then formatted to fit a "chr<N>" format where <N> is 1, 2, ..., 22, X or Y to correspond to the sequence field values in the bed file containing the peaks.

The list of repeated elements for hg19 was downloaded from USCS Genome Browser Table Browser (http://genome.ucsc.edu/cgi-bin/hgTables). The parameters were set as follows : i) clade to mammal, ii) genome to human, iii) assembly to hg19, iv) group to repeat, v) track to

repeatMasker. Finally all the regions that were not mapped to chromosome 1 to 22, M, X or Y were filtered out.

7.9.3 Data post-processing

The reads that did not have a proper barcode were filtered out using "python3.6 filter_bam.py -i <reads_bam> –tag CB –values <barcodes_txt> -o <output_bam>" where <reads_bam> is the bam file containing the reads, <barcodes_txt> the file containing the barcodes created with the above grep command and <output_bam> is the output bam file. filter_bam.py is a in-house developed python program for this project.

The peak name field was modified such that the values corresponding to a chromosome name followed a "chrN" format using "sed -E s/^\([0-9XY])/chr\\1/". Then only the peaks mapping to chromosome contigs were selected using "grep -E ^chr".

The peaks that had at least 30% overlap with any repeated element were filtered out using "bedtools substract -A -f 0.3 -a <peaks_bed> -b <repeats_bed>" where <peak_bed> and <repeats_bed> are the peak and repeat files in bed format, from the bedtools suite (Quinlan and Hall, 2010). Finally, the peaks were sorted by position using "sort -K 1,1V -k2,2n -k3,3n".

7.9.4 Model extension

Let's assume that we have partitioned a read density matrix R of dimensions NxL using K classes, with a shifting freedom S and with flipping. The posterior matrix probability P has dimensions $NxKxSx2$ (region, class, shift, flip) and the K models have a length of $L' = L-S+1$.

Extending the models is obtained by computing a larger matrix R^{ext} of dimensions NxL'' where $L'' = L+E$. In this case E is the extra number of columns to add. Care should be taken to construct R^{ext} by adding exactly $E/2$ columns one each side of R such that R is contained in the central part of R^{ext}.

Once R^{ext} has been constructed, the model update step (equation 11 in Nair et al. (2014)) can be applied on it using the posterior probability matrix P. This will results in the creation of K models of length $L''-S+1$. The K original models will be contained in the central part of the K extended models.

Extending a sequence model is done using exactly the same procedure with the exception that equation 7.5 should be used to compute the models.

The read and sequence model extension procedures are implemented in C++ in the ReadModelExtender and SequenceModelExtender programs.

7.9.5 Extracting data assigned to a class

The aim of the manipulation described in this section conceptually corresponds to creating a matrix containing only the rows (region) assigned to a given class X. This is of interest to run downstream analysis on this specific data subset (in this case, further clustering). For hard clustering algorithms, this can be done quite easily. It only requires to select the regions (rows) that have been assigned to class X. However, for soft partitioning, things are different and each and every region is assigned to all classes, with varying degrees (probabilities) (Dalton et al., 2009).

Let's assume that a first matrix of read densities R of dimensions NxL containing N regions of length L has been partitioned in K classes by ChIPPartitioning, with shifting freedom $S < L$ and flip. This created a probability matrix P of dimensions $NxKxSx2$ (region, class, shift, flip).

As described in section 7.5.5, ChIPPartitioning models computed are equivalent to aggregation plots. The aim is thus to do the reverse operation : unfold a given class X model (a given aggregation vector) to create a matrix having N rows and $L-S+1$ columns where S is the shifting freedom. However, if S is high, then the number of columns $L-S+1$ is low and the created matrix does only provide a narrow vision on the regions of interest. Enlarging the matrix such that it has dimensions NxL' where $L' \leq L-S+1$ can be desirable.

Let us proceed in two steps.

First, let us construct an extended matrix of read densities R^{ext} of dimensions NxL' where $L' = L+S$. Constructing R^{ext} is equivalent to adding $S/2$ columns on each side of R (as described in section 7.9.4).

Second, let us construct the wanted final matrix R^{class} of dimensions NxL using a modified "class model update" procedure. This procedure, instead of aggregating the signal into a single vector to create the class model, unfold it and create a matrix (as described in algorithm 2).

Because some regions (rows) can be assigned with a really low probability to class X, the

corresponding rows in R^{class} will show a really weak signal or even no signal at all.

Algorithm 2: Computes a matrix containing the data assigned to a given class S.

```
1  matrix ExtractClassData(){
       Data: The matrices R' and P.
       Result: The class matrix R^class
2      // overall class probabilies
3      class.prob = vector of K 0's ;
4      tot = 0 ;
5      for ifrom1toN do
6          for jfrom1toK do
7              for kfrom1toS do
8                  for lfrom1to2 do
9                      class.prob_j += p_{i,j,k,l} ;
10                     tot +=+= p_{i,j,k,l} ;
11                 end
12             end
13         end
14     end
15     for jfrom1toK do
16         class.prob_j /= tot ;
17     end
18     // modified class model update
19     for iin1toN do
20         for sin1toS do
21             // forward orientation
22             from.dat2.fw = s ;
23             to.dat2.fw = from.dat2.fw + L − 1 ;
24             j.dat3.fw = 1 ;
25             for j.dat.2.fwfromfrom.dat2.fwtoto.dat.2.fw do
26                 R^class_{j,j.dat3.fw} += (P_{i,X,s,1} × R'_{i,j.dat.2.fw}) / class.prob_X ;
27                 j.dat3.fw += 1 ;
28             end
29             // reverse orientation
30             j.dat3.fw = 1 ;
31             from.dat2.rv = L' − 1 − s ;
32             to.dat2.rv = from.dat2.rv − (L − 1) ;
33             for j.dat.2.rvfromfrom.dat2.rvdowntoto.dat.2.fw do
34                 R^class_{j,j.dat3.fw} += (P_{i,X,s,2} × R'_{i,j.dat.2.rv}) / class.prob_X ;
35                 j.dat3.fw += 1 ;
36             end
37         end
38     end
39     return R^class
```

The same can be done for sequence data. The read density and sequence data extraction procedures are implemented in C++ in the ClassReadDataCreator and ClassSequenceDataCreator respectively.

7.9.6 Programs

In order to allow an easy handling and a quick treatment of the data, the algorithms and procedures described above have been implemented in C++ and some are fully multi-threaded. Here is a list of the relevant C++ implementations :

- SequenceMatrixCreator : creates a sequence matrix in which each row contains the sequence of a given region. It takes as input a fasta file containing the sequences of interest, a bed file that specifies regions of interest in the fasta sequences and a from/to range that will specify the sequence boundaries around the center position of the regions listed in the bed file.
- ReadMatrixCreator : creates a read density matrix in which each row contains the number of reads mapping at each position of a given region. It takes as input a bam file containing the sequencing reads of interest, the bam index file, a bed file that specifies regions of interest and a from/to range that will specify the region boundaries around the center position of the regions listed in the bed file. Additionally, it also takes a bin size that is used to aggregate the counts in each region. A bin size of 1 means a signal resolution at the base-pair.
- WhichNullRows : returns the indices of the rows that have no signal in a read density matrix. It takes a read density matrix file as input.
- EMRead : implementation of the ChIPPartitioning algorithm. Takes a read count data matrix as input, the number of classes, the shifting and flipping parameters and return the posterior probability matrix in binary format.
- EMSequence : implementation of the EMSequence algorithm. It takes a DNA sequence data matrix as input, the number of classes, the shifting and flipping parameters and return the posterior probability matrix in binary format.
- EMJoint : implementation of the generalized EMJoint algorithm. It takes any number of data matrix as input, the number of classes, the shifting and flipping parameters and return the posterior probability matrix in binary format. This program can be given 0 or 1 DNA sequence matrix and any number of read count matrices as input. If a DNA sequence matrix is given, the read density matrix/matrices given aside must be at the single base-pair resolution (bin size). If only read density matrices are given, any resolution is acceptable.
- ProbToModel : implementation of the data realignment procedure. It takes as input a data matrix (DNA sequence or read counts) and a matrix of posterior probabilities as

input and returns the corresponding (read or sequence) class models. These values can be used as they are to create soft aggregation plots.

- ClassReadDataCreator : implementation of the class data extraction procedure for read density data. It takes as input a peak file (which centers will be positioned in the center of each row), a bam file containing the reads of interest, the corresponding bam index file, a posterior probability matrix file returned by a partitioning program, a from/to range indicating the width of the final matrix and the number (the ID) of the class to extract.

- ClassSequenceDataCreator : implementation of the class data extraction procedure for sequence data. It takes as input a peak file (which centers will be positioned in the center of each row), a fasta file containing the sequences of interest, the posterior probability matrix file returned by a partitioning program, a from/to range indicating the width of the final matrix and the number (the ID) of the class to extract.

- ReadModelExtender : implementation of the read model extension procedure. This program takes as input a bam file containing the reads, a bam file index file, a bed file containing the regions of interest, a probability matrix created by a classification program, the from/to values that were used to create the matrix that was classified, the bin size that was used to create the matrix that was classified and the total number of columns (positions) to add to the model. The result of this program is the extended model.

- SequenceModelExtender : implementation of the sequence model extension procedure. This program takes as input a fasta file containing the genome sequence, a bed file containing the regions of interest, a probability matrix created by a classification program, the from/to values that were used to create the matrix that was classified and the total number of columns (positions) to add to the model.

7.9.7 Fragment classes

The distribution of fragment sizes was modeled as a mixture of three classes (open chromatin, mono-nucleosomes, di-nucleosomes), each following a Gaussian distribution. Each class fragment length distribution was modeled using :

$$f(x) = a \times \exp^{\frac{-(x-m)^2}{2\times s}} \tag{7.7}$$

where x is the fragment length, m the mean fragment length for this class, s the fragment length standard deviation and a an amplitude factor.

The mean parameters were initialized to 50, 200 and 300bp. The standard deviation parameters were initialized to 10, 10 and 30bp and the amplitude factors to 1. The parameters were fitted

to the data using the the nls() function in R using the Gauss-Newton algorithm.

The reads were sorted from the bam file using "python3.6 split_by_size.py -i <read_bam> -o <out_bam> –length <from>-<to>" where <read_bam> is the bam file containing the reads, <out_bam> the output bam and split_by_size.py is a program developed for this project. For the open chromatin fragments, <from> and <to> were set to 30 and 84. For the mono-nucleosome fragments <from> and <to> were set to 133 and 266. Finally, for the di-nucleosome fragments <from> and <to> were set to 341 and 500.

In order to create a nucleosome fragment set, the di-nucleosome fragments were cut in two at their center position using "python3.6 split_in_two.py -i <dinucl_bam> -o <split_bam>" where <dinucl_bam> is the bam file containing the di-nucleosome fragments and <split_bam> is the output bam file containing the the reads corresponding to the fragments cut in two. "split_in_two.py" is a in-house developed python program for this projet. This file was then sorted using "samtools sort <split_bam>" from the samtools suite (Li et al., 2009). The sorted file was then merged with the mono-nucleosome fragments using "samtools sort <mononucl_bam> <split_bam>" where <mononucl_bam> is the bam file containing the mono-nucleosome fragments and <split_bam> is the sorted bam file containing the di-nucleosome fragments split in two.

Finally, the open chromatin fragment and nucleosome fragment bam files were indexed using "samtools index <bam_file>" where <bam_file> is the bam file to sort.

7.9.8 Simulated sequences

2'000 synthetic DNA sequences of 100bp long were generated. Two equiprobable classes were created and each sequence was assigned to either of the two classes. Each class was defined by a 8bp sequence motif (Figure A.15). Each sequence had exactly one motif occurrence, anywhere in the sequence (with a uniform probability), on either strand (equiprobable). The motif occurrence sequence was sampled using the corresponding class model. Finally, the bases outside the sequence were sampled using a mono-nucleotide model with 0.25 probability for each base.

7.9.9 Binding site prediction

The hg19 genome was scanned on both strands using PWMScan (Ambrosini et al., 2018) from its web interface (https://ccg.epfl.ch/pwmtools/pwmscan.php) to predict the binding sites of CTCF, EBF1, myc and SP1 using PWMs from the JASPAR 2018 collection (Khan et al., 2018) (MA0139.1 CTCF, MA0154.3 EBF1, MA0147.3 MYC and MA0079.3 SP1 respectively). The reference positions were set to 10, 7, 6 and 6 respectively and the threshold was set to a pvalue of 1^{-6} for all TFs excepted for SP1 for which a pvalue of 1^{-7} was used. Non-overlapping matches option was enabled.

7.9.10 Realignment using JASPAR motifs

The peaks were filtered for repeated elements. The peaks that had at least 30% overlap with any repeated element were filtered out using "bedtools substract -A -f 0.3 -a <peaks_bed> -b <repeats_bed>" where <peak_bed> and <repeats_bed> are the peak and repeat files in bed format, from the bedtools suite (Quinlan and Hall, 2010).

70'642 DNA sequences of 1'0001bp corresponding to the central position of each peak +/- 500bp were extracted and used to create a sequence matrix using "SequenceMatrixCreator –bed <peak_rmsk_bed> –fasta <hg19_fasta> –from -500 –to 500" where <peak_rmsk_bed> is the bed file containing the repeat filtered peaks and <hg19_fasta> is a fasta file containing chromosome 1, 2, ..., 22, X and Y sequences of the hg19 genome assembly.

The corresponding open chromatin and nucleosome sequencing read density matrices were created using "CorrelationMatrixCreator –bed <peak_rmsk_bed> –bam <open_bam> –bai <open_bai> –from -500 –to 500 –binSize 1 –method read_atac" and "CorrelationMatrixCreator –bed <peak_rmsk_bed> –bam <nucl_bam> –bai <nucl_bai> –from -500 –to 500 –binSize 1 –method fragment_center". <open_bam> and <open_bai> are the bam file containing the ATAC-seq reads, as provided by 10xGenomics and <open_bai> the corresponding bam index file. The bin size of 1b ensured that the read density matrix regions exactly correspond the sequence matrix regions.

A total of 23 binding models were downloaded from the motif clustering of JASPAR (Castro-Mondragon et al., 2017). Briefly, the motif clustering is made of a forest of trees (each tree is a cluster in which the leaves are the individual TF binding models). Internal nodes binding models are also available. As a matter of fact, they represent a consensus over multiple individual TF binding models. In order to i) have models representing the binding specificity of the TFs of interest and ii) widen the analysis to other TFs if they were sufficiently related to one of the TFs of interest in terms of specificity, I manually selected binding motifs, in the different motif trees, that would fit these requirements.

The downloaded models were :

Binding models downloaded			
Cluster ID	Node ID	TFs covered	Name
1	74	ARID3b, LHX3	LHX3
2	12	ESRRG, NR4A1, ESRRB, NR2F2	NR4A1
3	23	FOSL1::JUNB, FOSL1::JUN, FOS::JUND, FOSL2::JUN, FOS::JUNB, JDP2, NFE2, FOSL1, FOS, JUND, FOSL2, JUNB, JUN::JUNB, FOSL1::JUND, FOS::JUN, FOSL2::JUND, FOSB::JUNB, FOSL2::JUNB, BATF::JUN, JUN	AP1
3	24	NFE2L2, BACH1::MAFK, MAF::NFE2, BACH2	NFE2
4	22	max::myc, MXI1, myc, mycn	myc
4	30	ARNT, AHR::ARNT	AHR
4	31	HIF1A, HES5, HES7	HIF1A
5	20	CEBPA, CEBPG, CEBPD, CEBPB, CEBPE	CEBP
7	13	SPIC, SPI1	PU.1
7	17	ELF5, ELF3, EHF, ELF1, ELF4	ELF
19	2	NFAT5,NFATC1,NFATC3	NFAT
20	4	MEF2C,MEF2B,MEF2A,MEF2D	MEF2
21	5	GATA3, GATA5, GATA4, GATA6, GATA1, GATA2	GATA
28	13	EGR2, EGR4, EGR1, EGR3	EGR
28	14	KLF4,KLF1,KLF9	KLF
31	4	IRF7, IRF9, IRF4, IRF8, IRF5	IRF4
31	5	STAT1::STAT2, IRF2	IRF2
32	STAT6	STAT6	STAT6
33	1	SOX3, SOX6	SOX
38	3	RUNX1, RUNX2, RUNX3	RUNX
39	1	E2F3, E2F2	E2F
48	CTCF	CTCF	CTCF
66	1	FOXH1	FOXH1

Table 7.1 – TF binding models from JASPAR matrix clustering. Each model can be retrieved within JASPAR matrix clustering (http://jaspar2018.genereg.net/matrix-clusters/vertebrates/?detail=true) using the cluster and node ID. "TFs covered" refers to all TF which models are children of the given node. "Name" refers to the label this model is referred to in the text and figures.

All the binding models were downloaded as LFMs, in JASPAR format and were then converted into LPMs.

The DNA sequence matrix was partitioned into 23 classes using EMSequence with the model initialized using the 23 LPMs from JASPAR using "EMSequence –seq <matrix_seq> –class 23 –motifs <LPM1,LPM2,...,LPM23> –shift 971 –flip –iter 1 –seed <seed> –out <file_out>" where

<matrix_seq> is a text file containing the sequence matrix created by SequenceMatrixCreator, <LPM1,LPM2>,...,LPM23> is a coma separated list of 23 files containing the 23 JASPAR LPMs, <seed> a randomly generated seed and <file_out> the output file containing the probability matrix.

This resulted in the creation of 23 31bp sequence models from which the corresponding read models (open chromatin and nucleosome) were computed using "ProbToModel –read <reads_mat> –prob <prob_mat>" where <reads_mat> is the file containing the read density matrix created using CorrelationMatrixCreator and <prob_mat> is the file containing the probability matrix returned by EMSequence. These models were then extended of 1000bp leading to the creation of 1031bp models that are displayed in Figures A.23 and 7.2.

The sequence models were extended using "SequenceModelExtender –bed <peak_rmsk_bed> –fasta <hg19_fasta> –prob <prob_mat> –from -500 –to 500 –ext 1000" where the values of the –bed, –fasta, –from and –to option were the values used for the creation of the sequence matrix using SequenceMatrixCreator and <prob_mat> is the probability matrix file returned by EMSequence.

The read models were extended using "ReadModelExtender –bed <peak_rmsk_bed> –bam <open_bam> –bai <open_bai> –prob <prob_mat> –from -500 –to 500 –ext 1000 –binSize 1 –method <m>" where the value of the –bed, –bam, –bai, –from, –to, –binSize and –method options were the values used for the creation of the read density matrices (open chromatin and nucleosomes) using CorrelationMatrixCreator and <prob_mat> is the probability matrix file returned by EMSequence.

7.9.11 Per TF sub-classes

For each of the 23 TF classes, a sequence matrix, an open chromatin read density matrix and a nucleosome read density matrix were created.

The sequence matrix was created using "ClassSequenceDataCreator –bed <peak_rmsk_bed> –fasta <hg19_fasta> –prob <prob_mat> –from -500 –to 500 –k <X> –out <matrix_out>" where the values of the –bed, –fasta, –from and –to option were the values used for the creation of the sequence matrix using SequenceMatrixCreator, <prob_mat> is the probability matrix file returned by EMSequence, <X> the number of the class to extract and <matrix_out> the output file.

The read density matrices were created using ""ClassReadDataCreator –bed <peak_rmsk_bed> –bam <open_bam> –bai <open_bai> –prob <prob_mat> –from -500 –to 500 –binSize 1 –method <m>" where the value of the –bed, –bam, –bai, –from, –to, –binSize and –method options were the values used for the creation of the read density matrices (open chromatin and nucleosomes) using CorrelationMatrixCreator and <prob_mat> is the probability matrix file returned by EMSequence.

The rows with no signal (0) in the open chromatin read density matrix were listed using "WhichNullRows –mat <reads_mat>" where <reads_mat> is the matrix created using ClassReadDataCreator.

The partitioning was performed on the open chromatin signal using ChIPPartitiong using "EMRead –read <reads_mat> –iter 20 –class <k> –shift 1 –filter <filter_txt> –seed <seed> –filter –out <prob_mat>" where <reads_mat> is the file containing the open chromatin read density matrix created using ClassReadDataCreator, <k> is the number of classes from 1 to 10, <filter_txt> the file created by WhichNullRows, <seed> a randomly generated seed and <prob_mat> is the output file containing the probability matrix. No shifting nor flipping was used because the previous step of TF motif occurrence alignment already solved it. The best partition was estimated by manual curation.

8 Discussion

In this chapter, I get back to some of the major aspects - in my opinion - that I presented in the previous chapters and discuss them in a larger scope and present some related perspectives.

About the chromatin organization

The systematic study of nucleosome organization arount TF binding sites revealed that all TFs showed a strong array on at least one of their flank. A possible explanation regarding this asymetry is that the upstream and downstream arrays are organized with respect to different anchors. The methodology I used to display them could only phase the arrays on one side, at the price of unphasing - to different extent - the array on the other side, rendering it hardly visible on the aggregations. Thus, it is reasonable to claim, as a general rule, that nucleosomes are organized into regular arrays on both sides of all TF binding sites.

The case of CTCF arrays - literally a school case when it comes to nucleosome arrays - was investigated further and the ISWI enzyme SNF2H has been shown to be necessary to maintain this typical nucleosome organization (Wiechens et al., 2016). The same study also showed that SNF2H and SNF2L are necessary to maintain the nucleosome organizations at RUNX5 and JUN binding sites.

This work shows that all TFs binding sites are flanked by nucleosome arrays. The involvement of chromatin remodeler is thus likely to be general, as well. Understanding these nucleosome arrays are maintained and how exactly they are delimited is crucial. The extent to which the chromatin is open defines the boundaries of each regulatory element and thus which TF binding sites are accessible and which should remain in the closed chromatin on the flanks. It is thus unsurprising that the dysfunction or deregulation of chromatin remodelers has been linked with cancer (Wilson and Roberts, 2011; Längst and Manelyte, 2015). Consequently, delineating how TF and chromatin remodeling complexes influence each other to regulate the expression of genes is crucial.

About pioneer factors

This work also provided supporting evidence about the pioneer role of EBF1. The results suggest that EBF1 binding sites tend to be located at the edges of rotationally positioned nucleosomes.

To my knowledge, no direct evidence of EBF1 ability to engage closed chromatin has ever been proposed. EBF1 pioneer function was rather based on its ability to drive cellular differentiation (Hagman and Lukin, 2005) and to trigger chromatin remodeling (Maier et al., 2004; Boller et al., 2016). This work suggests that EBF1 also exhibit the major characteristic of pioneer TFs, that is, the ability to engage DNA in inaccessible chromatin (Iwafuchi-Doi and Zaret, 2014). Based on this result, an *in vitro* assessment of this property, as performed in Soufi et al. (2015) for Oct4, Sox2, Klf4 and c-myc, could help to further strengthen this observation.

Finally, having found a pioneer behaving TF in the ENCODE data raises a question : why no more than one pioneer TF could be found? To my opinion, the answer can be declined in two parts. First, this dataset only contained a few dozens of TFs, likely not containing more pioneer TF besides EBF1. Second, upon binding, pioneer TFs are known to trigger chromatin opening. Thus, it is likely that the observation of the chromatin organization at pioneer TF binding sites will result in the observation of a steady-state : a TF binding in an open chromatin region. As a consequence, capturing TFs bound in closed chromatin seems difficult and identifying pioneer TF, in this way, unlikely. In the light of this assumption, the EBF1 results are puzzling and call for a further delineation of the events triggered upon EBF1 binding.

About assaying TF specificity

Throughout this work, I also proposed several different algorithms and show their usefulness. One of them is EMSequence. EMSequence is a *de novo* motif discovery algorithm. I proposed to use this new algorithm together with ChIPPartioning to study the chromatin structure at TF binding sites. However, other applications could be explored. For instance, in chapter 5, SMiLE-seq has been demonstrated to be an effective method to assay TF specificity. Even though not discussed in this work, it also proved to be useful to assay AP1 dimer specificity (Isakova et al., 2017). Interestingly, the question of TF dimer specificity remains largely unsolved. Here, an adequat usage of computational methods can allow to alleviate the experimental effort necessary to produce the relevant data. Assaying a pair of TF specificity implies to run at least 3 independent assays : each TF alone to assay the homodimers and both TFs together to assay to heterodimer. Because EMSequence is a partitioning algorithm, it should be able to discover several motifs at the same time. Thus it should be possible to assay a pair of dimerizing TFs at once and to retrieve each TF homodimer motif (if any) and the heterodimer motifs from a single experiment, using EMSequence. However, one should keep in mind that, in competition, certain dimers may be favored over others, depending on the affinity of each TF for its possible partners. Therefore, for instance, it may happen that one dimer never forms

in competition, thus excluding the discovery of its binding specificity.

About the treatment of scATAC-seq data

Nowadays, the technologies deviced at performing single-cell measurements have become commonly used. scRNA-seq data remains the most frequently used. Dedicated computational methods allow to isolate sub-populations of cells by clustering the gene expression matrix (Fan et al., 2016; Kiselev et al., 2017), using gene regulatory network reconstruction (Aibar et al., 2017) or by identifying cellular states based on the accessible region motif content (González-Blas et al., 2019).

scATAC-seq data are encountering an yet ever growing popularity. Currently, the treatment of these data remains quite limited for the time being. It is for instance not unusual to create a matrix, as for scRNA-seq, containing the number of reads mapped at a given location in a cell and subsequently using this matrix for downstream analyses such as the detection of cell populations. However, I think that scRNA-seq and scATAC-seq data are different by nature and thus cannot be treated the same.

In chapter 7, I presented a computational method to unravel footprints from ATAC-seq data. One can imagine using the framework described in this chapter to draw a catalog of chromatin structures from the pool of single-cell data and use it to annotate each cell. More precisely this could be done by going back to each peak in each cell and assigning a qualitative label corresponding to the chromatin model that matches the best (the most similar) this region in this cell. Ultimately, this would lead to the creation of a matrix (cells x regions) that could be used to run clustering methods. How the similarity should be computed and whether each cell will have a high enough coverage for similarity computations to be meaningful remain open questions. Alternatively, one can replace single cells by different bulk experiments. In this case, the clustering would not isolate cell sub-populations but experiments (individuals, culture conditions, etc) that are similar to each other.

9 Published articles

This chapter lists all the publications to which I participated as an author. The publication are listed by chronological order.

Isakova, A., Groux, R., Imbeault, M., Rainer, P., Alpern, D., Dainese, R., Ambrosini, G., Trono, D., Bucher, P., and Deplancke, B. (2017). SMiLE-seq identifies binding motifs of single and dimeric transcription factors. Nature Methods, advance online publication.

Dreos, R., Ambrosini, G., Groux, R., Cavin Périer, R., and Bucher, P. (2017). The eukaryotic promoter database in its 30th year: focus on non-vertebrate organisms. Nucleic Acids Research, 45(D1):D51–D55.

Dreos, R., Ambrosini, G., Groux, R., Périer, R. C., and Bucher, P. (2018). MGA repository: a curated data resource for ChIP-seq and other genome annotated data. Nucleic Acids Research, 46(D1):D175–D180.

Ambrosini, G., Groux, R., and Bucher, P. (2018). PWMScan: a fast tool for scanning entire genomes with a position-specific weight matrix. Bioinformatics, 34(14):2483–2484.

Groux, R. and Bucher, P. (2019). SPar-K: a method to partition NGS signal data. Bioinformatics.

Meylan, P., Dreos, R., Ambrosini, G., Groux, R., and Bucher, P. (2020). EPD in 2020: enhanced data visualization and extension to ncRNA promoters. Nucleic Acids Research, 48(D1):D65–D69.

A Supplementary material

A.1 ENCODE peaks analysis supplementary material

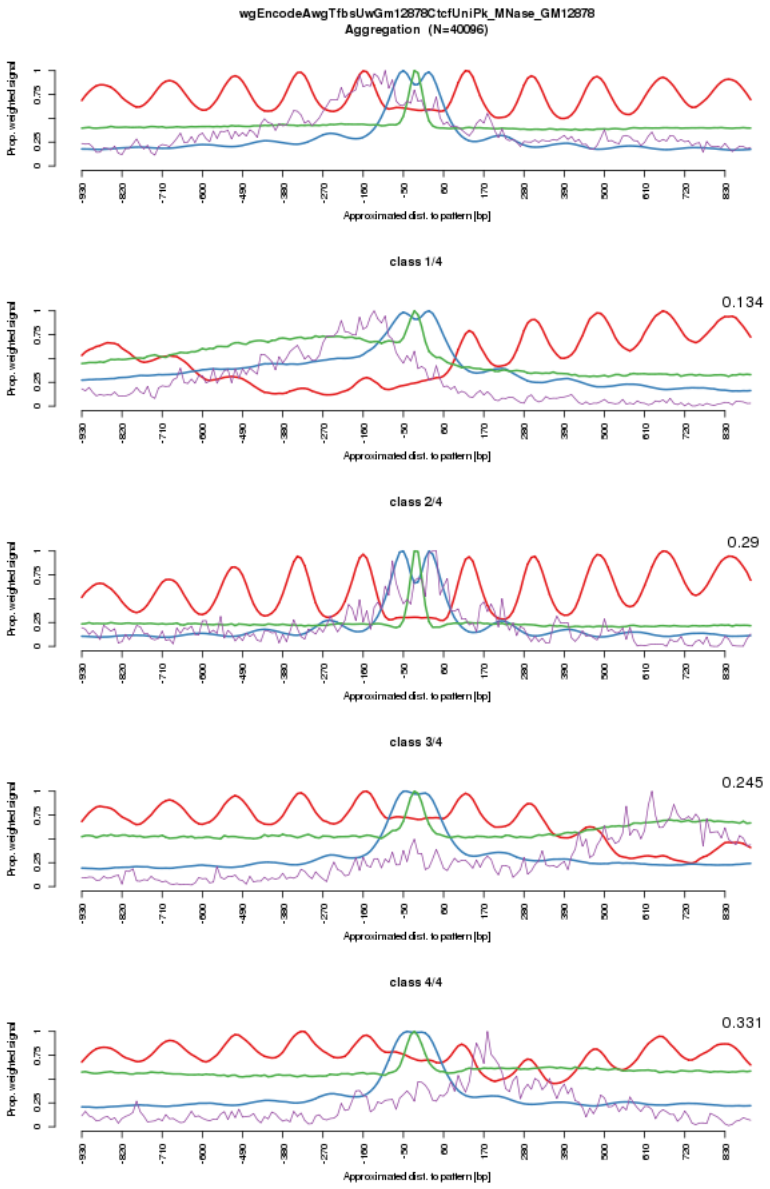


Figure A.1 – Chromatine architectures around CTCF binding sites discovered using ChIP-Partitioning. The partition was done with respect to the MNase reads (red), +/- 1kb around the peaks, in bins of 10bp, that were allowed to be shifted and flipped. DNaseI (blue), TSS density (violet) and sequence conservation (green) were realigned according to MNase classification

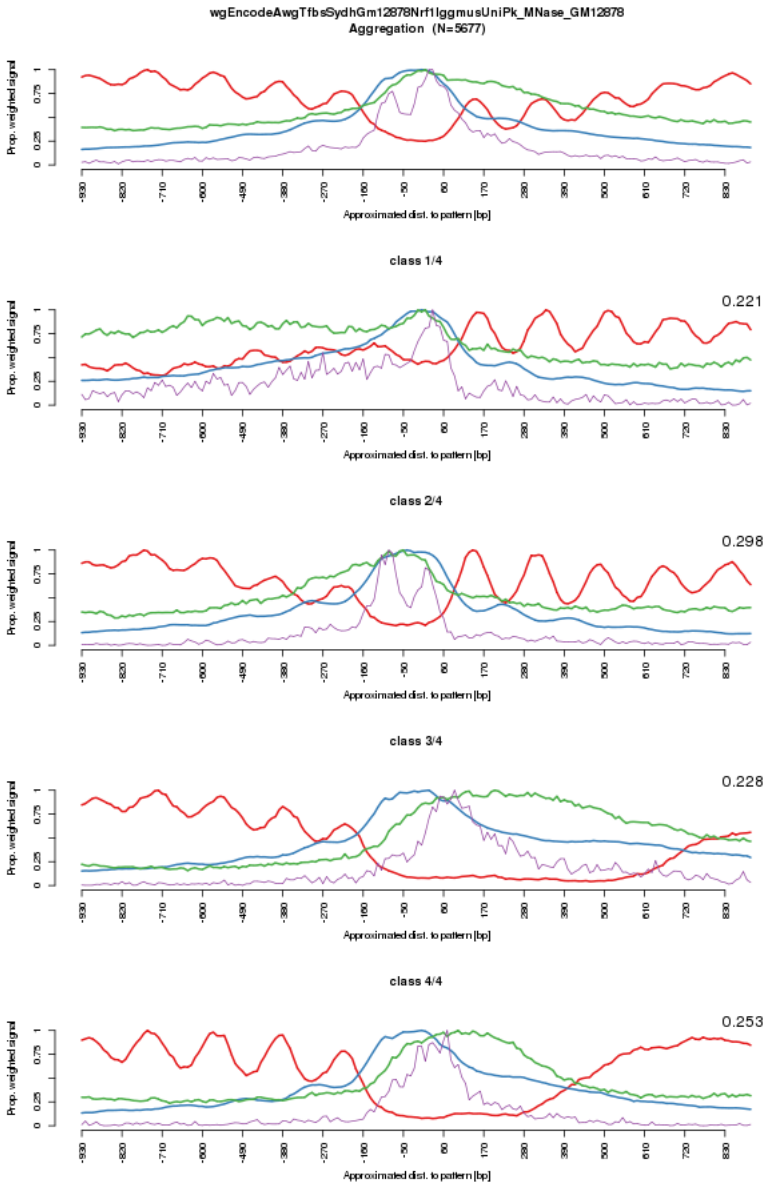


Figure A.2 – Chromatine architectures around NRF1 binding sites discovered using ChIP-Partitioning. The partition was done with respect to the MNase reads (red), +/- 1kb around the peaks, in bins of 10bp, that were allowed to be shifted and flipped. DNaseI (blue), TSS density (violet) and sequence conservation (green) were realigned according to MNase classification and overlaid. The y-axis scale represent the proportion of the highest signal for each chromatin pattern. The first row contains the aggregated signal over all sites. The number of binding sites (peaks) is indicated in parenthesis. The following rows contains the 4 classes discovered. Their overall probability is indicated atop of the class signal, on the right. The y-axis indicates the min/max signal for all densities.

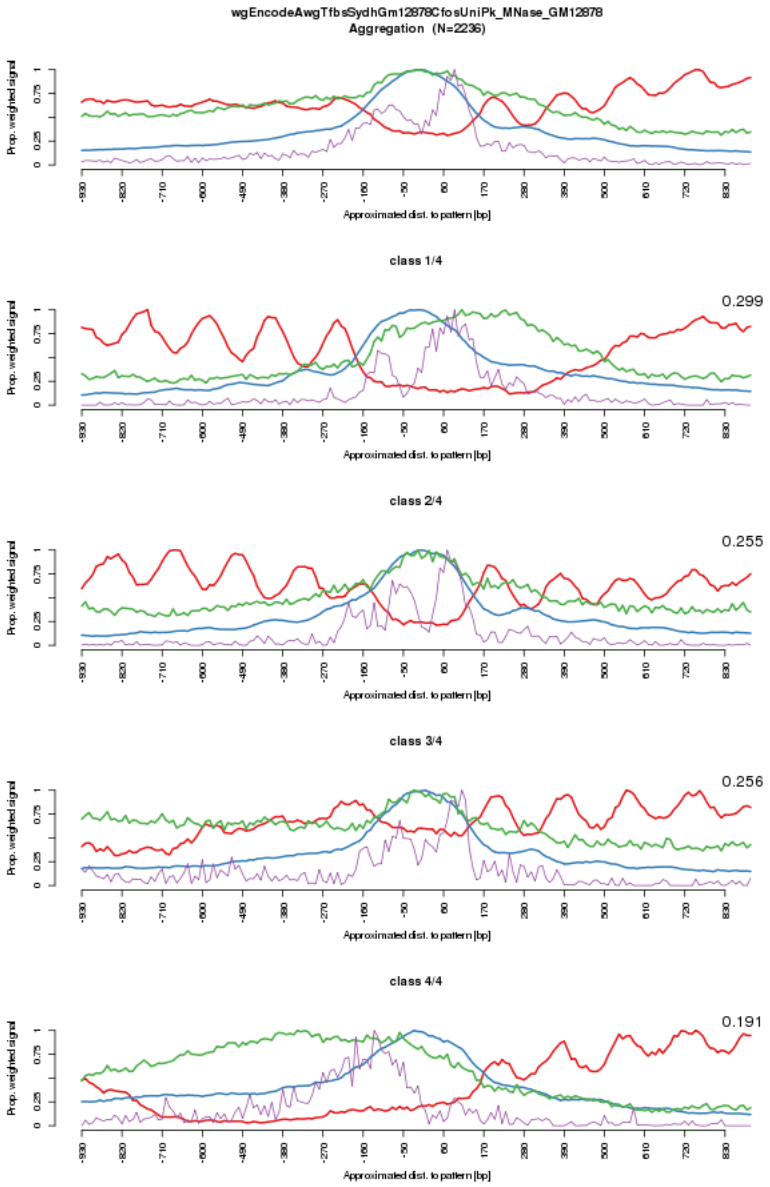


Figure A.3 – Chromatine architectures around cFOS binding sites discovered using ChIPPartitioning. The partition was done with respect to the MNase reads (red), +/- 1kb around the peaks, in bins of 10bp, that were allowed to be shifted and flipped. DNaseI (blue), TSS density (violet) and sequence conservation (green) were realigned according to MNase classification and overlaid. The y-axis scale represent the proportion of the highest signal for each chromatin pattern. The first row contains the aggregated signal over all sites. The number of binding sites (peaks) is indicated in parenthesis. The following rows contains the 4 classes discovered. Their overall probability is indicated atop of the class signal, on the right. The y-axis indicates the min/max signal for all densities.

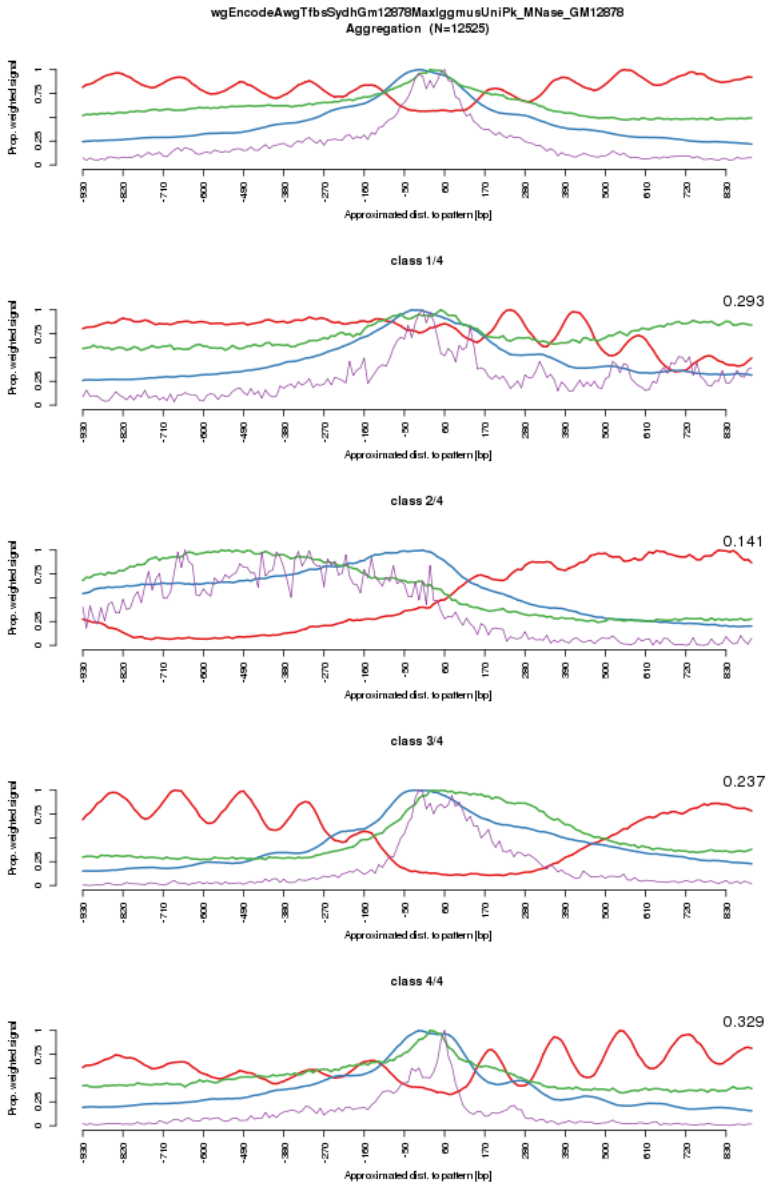


Figure A.4 – Chromatine architectures around max binding sites discovered using ChIPPartitioning. The partition was done with respect to the MNase reads (red), +/- 1kb around the peaks, in bins of 10bp, that were allowed to be shifted and flipped. DNaseI (blue), TSS density (violet) and sequence conservation (green) were realigned according to MNase classification and overlaid. The y-axis scale represent the proportion of the highest signal for each chromatin pattern. The first row contains the aggregated signal over all sites. The number of binding sites (peaks) is indicated in parenthesis. The following rows contains the 4 classes discovered. Their overall probability is indicated atop of the class signal, on the right. The y-axis indicates the min/max signal for all densities.

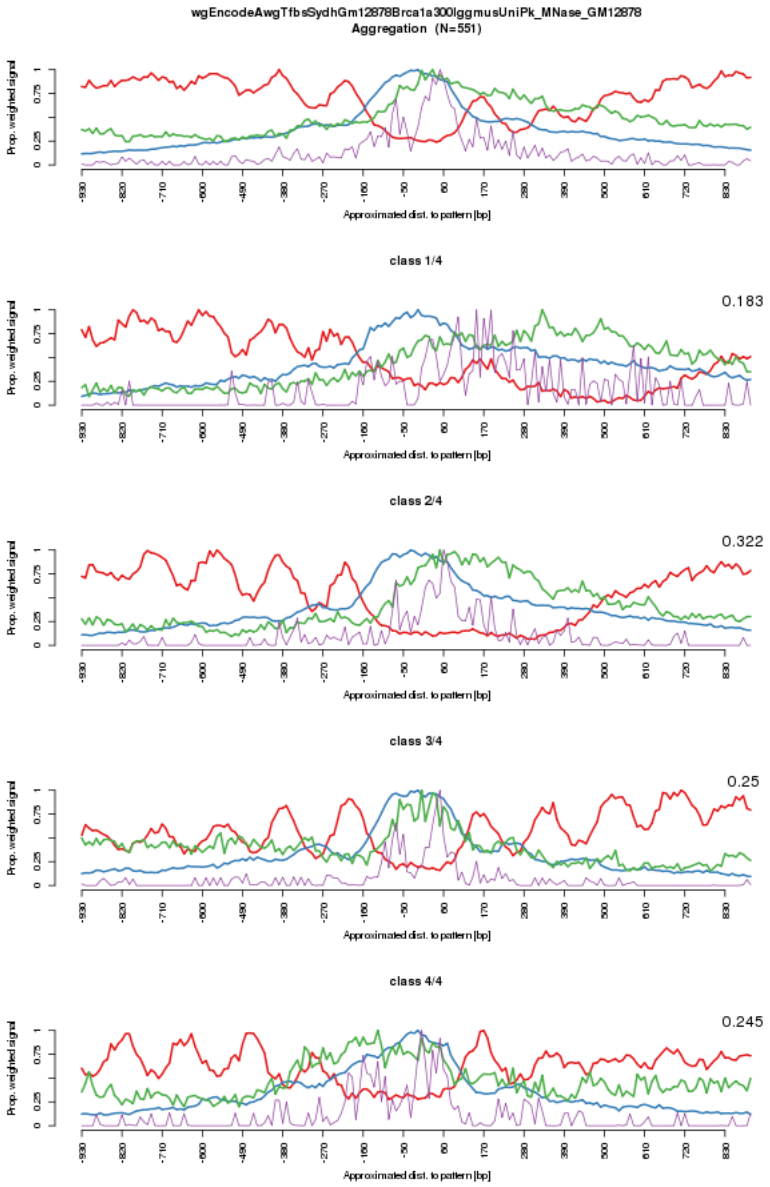


Figure A.5 – Chromatine architectures around BRCA1 binding sites discovered using ChIP-Partitioning. The partition was done with respect to the MNase reads (red), +/- 1kb around the peaks, in bins of 10bp, that were allowed to be shifted and flipped. DNaseI (blue), TSS density (violet) and sequence conservation (green) were realigned according to MNase classification and overlaid. The y-axis scale represent the proportion of the highest signal for each chromatin pattern. The first row contains the aggregated signal over all sites. The number of binding sites (peaks) is indicated in parenthesis. The following rows contains the 4 classes discovered. Their overall probability is indicated atop of the class signal, on the right. The y-axis indicates the min/max signal for all densities.

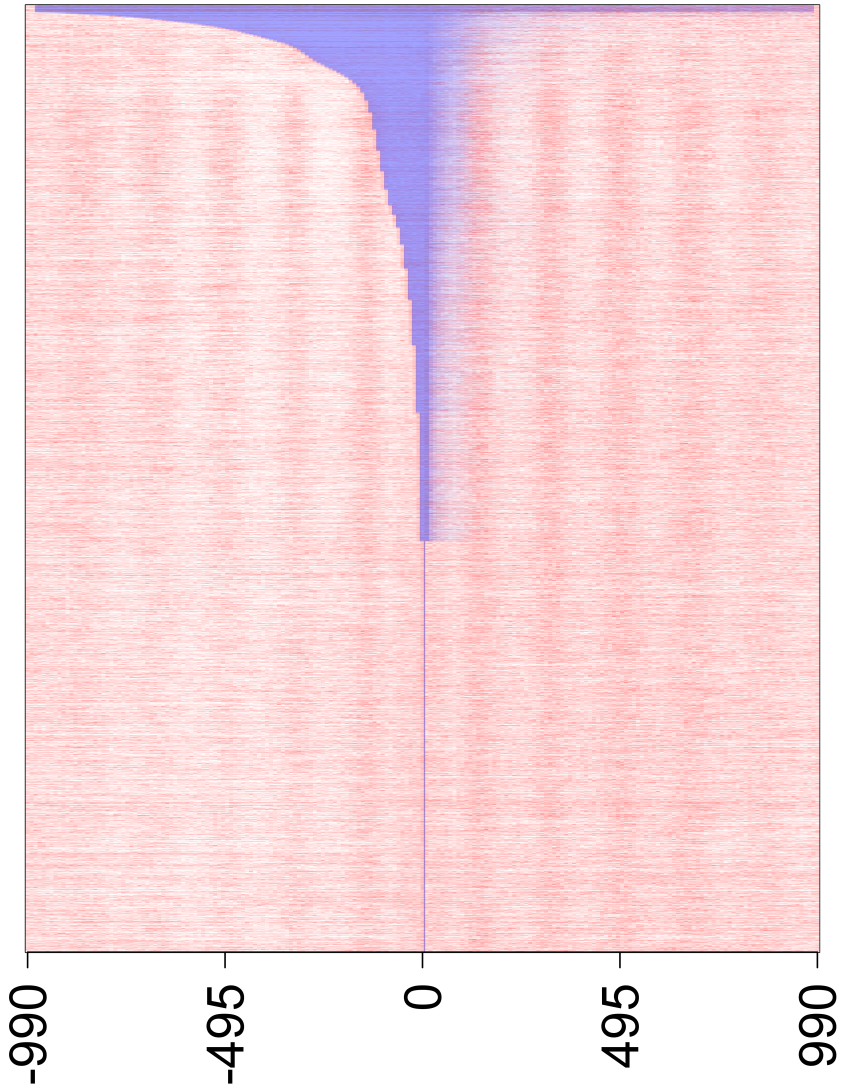


Figure A.6 – Nucleosome occupancy around CTCF peaks measured by MNase-seq, in bins of 10bp. The nucleosome depleted region is displayed in blue.

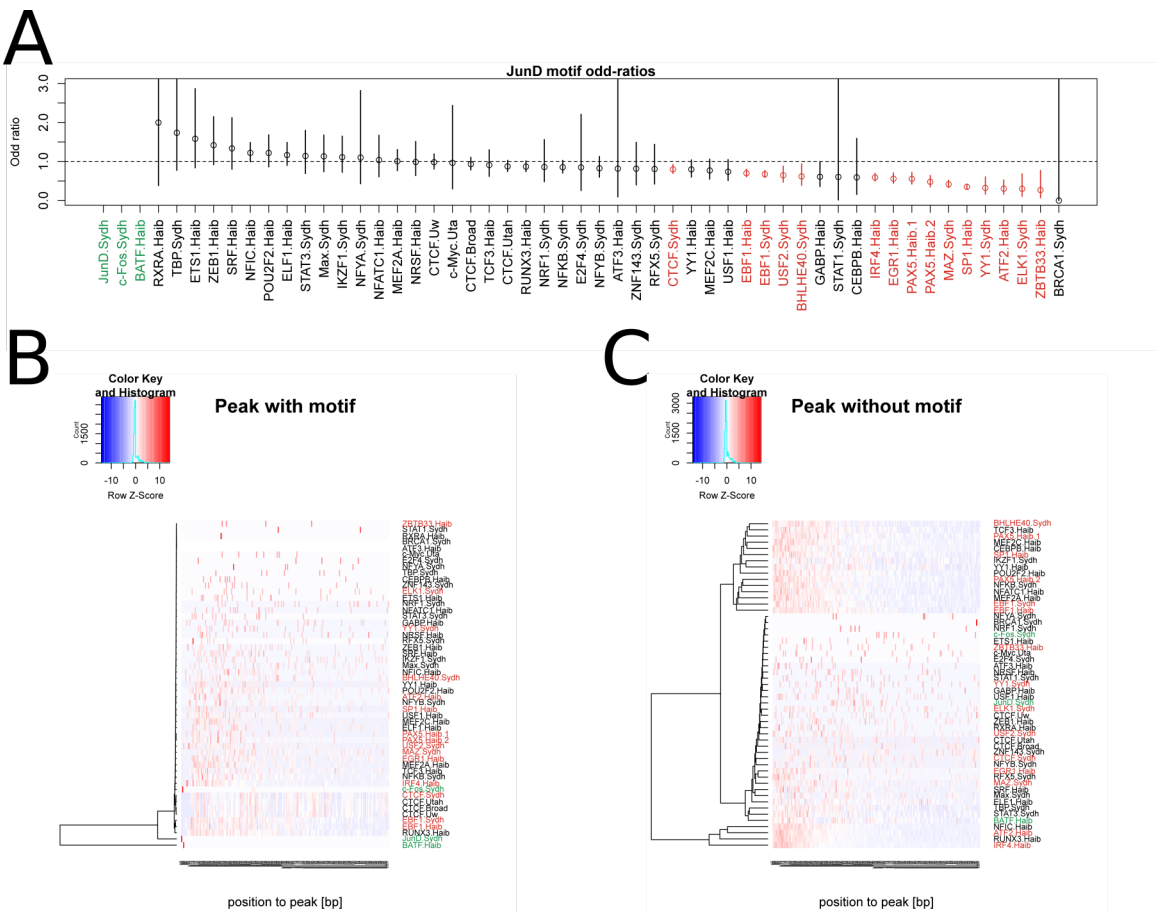


Figure A.7 – JunD motif association measured around the binding sites of different TFs. For a each TF, its binding sites, +/- 500bp, were searched for the presence of i) the TF motif occurrence and ii) CTCF motif occurrence. For each TF, a 2x2 contingency table was created with the number of peaks having i) both motif occurrences, ii) the TF motif occurrences only, iii) CTCF motif occurrences only and iv) no motif occurrence. **A** Odd ratio (OR) of the exact Fisher test performed on each TF contingency table. The ORs are displayed with their 95% confidence interval (CI). ORs > 1 - that is, with 1 not part of the 95%CI - are labeled in green and indicate an association of both motifs more frequent than expected by chance. ORs < 1 are labeled in red and indicate a repulsion of both motifs more frequence than expected by chance. The JunD and cFos dataset ORs are too high to be represented in this plot. **B** Density of JunD motif occurrences at the absolute distance of different TF binding sites (peak centers) which also have their own motif occurring (at distance 0). The rows were standardized and aggregated using the Euclidean distance. **C** Same as in (B) but for TF binding sites that does not have their own motif.

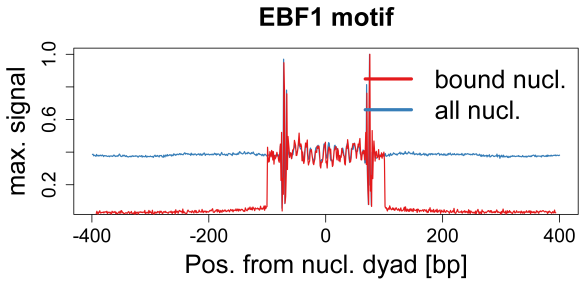


Figure A.8 – EBF1 binding sites around the dyad of nucleosomes having an occupied EBF1 motif occurrence within 100bp (in red) and of all nucleosomes (in blue). The abrupt decrease of EBF1 motif occurrence frequency at +/- 100bp reflects the nucleosome selection process.

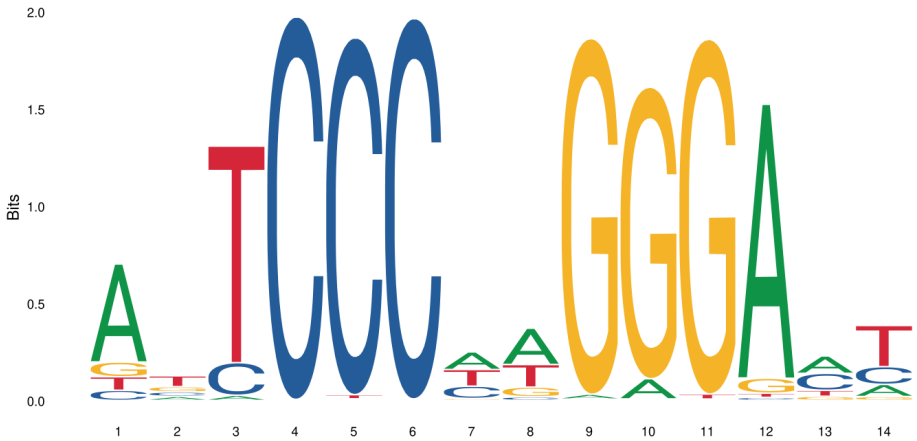


Figure A.9 – EBF1 logo from JASPAR binding model MA0154.3 (Khan et al., 2018).

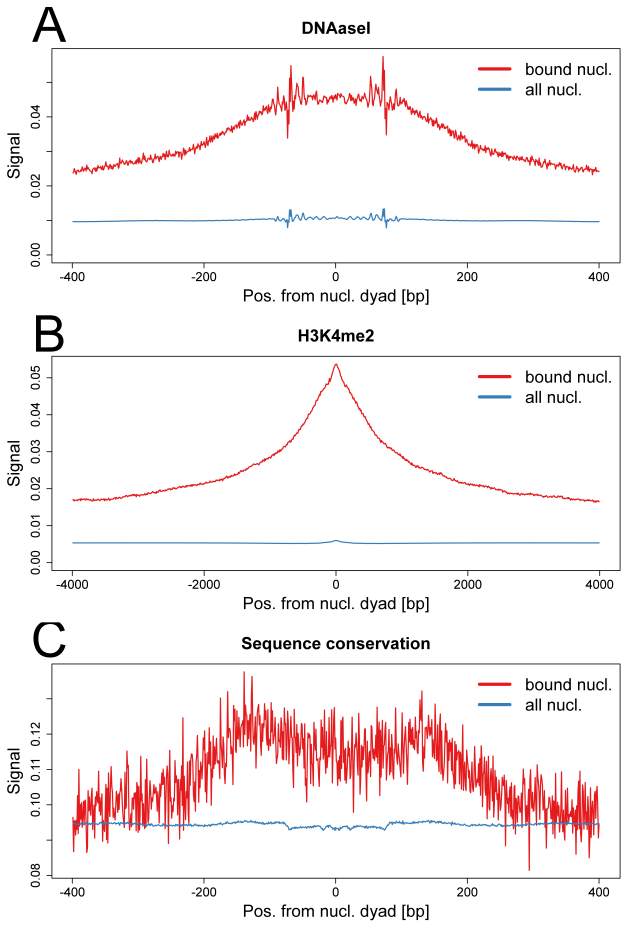


Figure A.10 – EBF1 binding sites chromatin features. **A** Chromatin accessibility around nucleosomes that have an EBF1 binding site within 100bp (red) and all nucleosomes (blue). **B** H3K4me2 deposition around nucleosomes that have an EBF1 binding site within 100bp (red) and all nucleosomes (blue). **C** Sequence conservation around nucleosomes that have an EBF1 binding site within 100bp (red) and all nucleosomes (blue).

A.2 SPar-K supplementary material

Algorithm 3: SPar-K algorithm.

```
1  void SPar-K(R, K, S, I){
       Data: R a 2D matrix of numericals of dimension N × L; K the number clusters; S
             the shifting allowed; I the maximum number of iterations.
       Result: 5 vectors of length N with the cluster labels, the data shift values, the
               reference shift values, the data flip values and the distances to the cluster
               references.
2      CLUSTER = vector of 0 of size N ;
3      SHIFT_DAT = vector of 0 of size N ;
4      SHIFT_REF = vector of 0 of size N ;
5      FLIP_DAT = vector of 0 of size N ;
6      DISTANCE = vector of 0 of size N ;
7      if smoothing then
8          smoothOutliers(R) ;
9      end
10     REF = a matrix of K rows and L columns;
11     select K rows of R and copy them in REF using seedingRandom(K, R) or
        seedingKmeans++(K, R) ;
12     n_iter = 1;
13     converged = false;
14     while n_iter != I or not converged do
15         for i = 0 to N − 1 do
16             D = a vector 0 of size K ;
17             S_REF = a vector 0 of size K ;
18             S_DAT = a vector 0 of size K ;
19             F_DAT = a vector 0 of size K ;
20             for k = 0 to K − 1 do
21                 D_k, S_REF_k, S_DAT_k, F_DAT_k = distanceFast(REF_k,, Ri,, S) ;
22             end
23             j = index of minimum value in REF;
24             CLUSTER_i = D_j ;
25             SHIFT_REF_i = S_REF_j ;
26             SHIFT_DAT_i = S_DAT_j ;
27             FLIP_DAT_i = F_DAT_j ;
28             DISTANCE_i = D_j ;
29         end
30         REF = computeClusterReferences(R, CLUSTER, SHIFT_REF,
            SHIFT_DAT, FLIP_DAT) ;
31         if n_iter > 1 and at least one value of CLUSTER changed since last iteration
            then
32             converged = false ;
33         end
34         else
35             converged = true ;
36         end
37         n_iter = n_iter + 1 ;
38     end
       return CLUSTER, SHIFT_REF SHIFT_DAT, FLIP DISTANCE
```

Algorithm 4: Smooth the data matrix by removing outliers.

```
1  MATRIX smoothOutliers(MATRIX){
       Data: MATRIX a 2D matrix of numericals of dimensions N × L.
       Result: a 2D matrix of numericals of dimensions N × L. with outlier smoothed.
2      foreach ROW in MATRIX do
3          m = mean of ROW ;
4          s = standard deviation of ROW ;
5          lower = m - 3*s ;
6          upper = m + 3*s ;
7          foreach value in ROW do
8              if value < lower or value > upper then
9                  if value has a left neighbor l and a right neighbor r then
10                     value = (l + r) / 2 ;
11                 end
12                 else if value has two left neighbors l1 and l2 then
13                     value = (l1 + l2) / 2 ;
14                 end
15                 else if value has two right neighbors r1 and r2 then
16                     value = (r1 + r2) / 2 ;
17                 end
18             end
19         end
20     end
21     return MATRIX
```

Algorithm 5: Fast algorithm to compute the correlation distance with shift and flip

```
1  d,s_xf,s_y,f_y distanceFast(X,Y,S){
       Data: X and Y two vectors of size L and S the allowed shifting
       Result: the distance between X and Y, the shift of X and Y and whether Y was
               flipped in the alignment.
2      /* initialize all variables and data structures                    */
3      initialize() ;
4      for i = 0 to n − 1 do
5          SUM_X_0 += X_i ;
6          SUM_X2_0 += X_i^2 ;
7          SUM_Y_0 += Y_i ;
8          SUM_Y2_0 += Y_i^2 ;
9      end
10     i = n ;
11     for s = 1 to s = S − 1 do
12         SUM_X_s = SUM_X_{s−1} + X_{i−n} + X_i ;
13         SUM_X2_s = SUM_X2_{s−1} + X_{i−n}^2 + X_i^2 ;
14         SUM_Y_s = SUM_Y_{s−1} + Y_{i−n} + Y_i ;
15         SUM_Y2_s = SUM_Y2_{s−1} + Y_{i−n}^2 + Y_i^2 ;
16         i += 1 ;
17     end
18     /* compute all distances with X having a shift of 0, Y having a
          shift of 0 and all the remains ones.  These functions can
          access and modify all variables and data structures in this
          function.                                                       */
19     fillFirstRow() ;
20     fillFirstColl() ;
21     fillRemaining() ;
22     /* if more than one, find the minimal distance which also
          minimize the alignment score                                    */
23     best_d = best_s_x = best_s_y = best_f_y = 0 ;
24     for tripplet in MIN_DIST do
25         s_x = tripplet_0 ;
26         s_y = tripplet_1 ;
27         f_y = tripplet_2 ;
28         min_score = ∞ ;
29         if SCORES_{s_x,s_y} < min_score then
30             min_score = SCORES_{s_x,s_y} ;
31             best_s_x = s_x ;
32             best_s_y = s_y ;
33             best_f_y = f_y ;
34             best_d = DIST_{s_x,s_y,f_y} ;
35         end
36     end
37     return best_d, best_s_x, best_s_y, best_f_y
```

Algorithm 6: A routine of distanceFast() that initializes all the necessary variables. This function can access and modify variables in distanceFast().

```
initialize(){
    l_half = L/2 ;
    n = L − S + 1 ;
    n_half = n/2 ;
    DIST = 3D matrix of 3 of dimensions S ∗ S ∗ 2 ;
    SUM_XY = 3D matrix of 0 of dimensions S ∗ S ∗ 2 ;
    SUM_X = a vector of 0 of size S ;
    SUM_X2 = a vector of 0 of size S ;
    SUM_Y = a vector of 0 of size S ;
    SUM_Y2 = a vector of 0 of size S ;
    SCORES = a 2D matrix of 0 of dimensions S ∗ S ;
    MIN_DIST = an empty list of triplets ;
    min_dist = 3 ;
```

Algorithm 7: A routine of distanceFast() computing all distances with X having a shift of 0. This function can access and modify all variables declared in distanceFast().

```
1  void fillFirstRow(){
2      from_x = 0;
3      to_x = from_x + n;
4      for i = 0 to S − 1 do
5          /* forward orientation                                    */
6          from_y = i;
7          to_y = from_y + n;
8          SUM_XY from_x, from_y, 0 = 0;
9          for j = 0 to n − 1 do
10             SUM_XY_{from_x,from_y,0} += X_{from_x+j} * Y_{from_y+j};
11         end
12         c = (n*SUM_XY_{from_x,from_y,0} − SUM_X_{from_x} * SUM_Y_{from_y}) /
               sqrt(n*SUM_X2_{from_x} − SUM_X^2_{from_x}) * sqrt(n*SUM_Y2_{from_y} − SUM_Y^2_{from_y});
13         if division by 0 occurred then
14             c = 0;
15         end
16         d = 1 − c;
17         DIST_{from_x,from_y,0} = d;
18         SCORES_{from_x,from_y} =
             |l_half − (n_half + from_x)| + |l_half − (n_half + from_y)|;
19         if d < min_dist then
20             min_distance = d;
21             append a triplet (from_x, from_y, 0) to MIN_DIST;
22         end
23         else if d = min_dist then
24             append a triplet (from_x, from_y, 0) to MIN_DIST;
25         end
26         /* reverse orientation                                    */
27         SUM_X Y from_x, from_y, 1 = 0;
28         for j = 0 to n − 1 do
29             SUM_XY_{from_x,from_y,1} += X_{from_x+j} * Y_{to_y−j−1};
30         end
31         c = (n*SUM_XY_{from_x,from_y,1} − SUM_X_{from_x} * SUM_Y_{from_y}) /
               sqrt(n*SUM_X2_{from_x} − SUM_X^2_{from_x}) * sqrt(n*SUM_Y2_{from_y} − SUM_Y^2_{from_y});
32         if division by 0 occurred then
33             c = 0;
34         end
35         d = 1 − c;
36         DIST_{from_x,from_y,1} = d;
37         if d < min_dist then
38             min_distance = d;
39             append a triplet (from_x, from_y, 1) to MIN_DIST;
40         end
41         else if d = min_dist then
42             append a triplet (from_x, from_y, 1) to MIN_DIST;
43         end
44     end
```

Algorithm 8: A routine of distanceFast() computing all distances with Y having a shift of 0. This function is can access and modify all variables declared in distanceFast().

```
1  void fillFirstCol(){
2      from_y = 0 ; to_y = from_y + n ;
3      from_y_rev = s − from_y − 1 ;
4      to_y_rev = from_y_rev + n ;
5      for i = 0 to S − 1 do
6          from_x = i ;
7          to_x = from_x + n ;
8          /* forward orientation                                   */
9          SUM_XY_{from_x,from_y,0} = 0 ;
10         for j = 0 to n − 1 do
11             SUM_XY from_x, from_y, 0 += X_{from_x+j} * Y_{from_y+j} ;
12         end
13         c = (n*SUM_XY_{from_x,from_y,0} − SUM_X_{from_x} * SUM_Y_{from_y}) /
               sqrt(n*SUM_X2_{from_x} − SUM_X^2_{from_x}) * sqrt(n*SUM_Y2_{from_y} − SUM_Y^2_{from_y}) ;
14         if division by 0 occurred then
15             c = 0 ;
16         end
17         d = 1 − c ;
18         DIST_{from_x,from_y,0} = d ;
19         SCORES_{from_x,from_y} =
             |l_half − (n_half + from_x)| + |l_half − (n_half + from_y)| ;
20         if d < min_dist then
21             min_distance = d ;
22             append a triplet (from_x, from_y, 0) to MIN_DIST ;
23         end
24         else if d = min_dist then
25             append a triplet (from_x, from_y, 0) to MIN_DIST ;
26         end
27         /* reverse orientation                                   */
28         SUM_XY_{from_x,S−1,1} = 0 ;
29         for j = 0 to n − 1 do
30             SUM_XY_{from_x,S−1,1} += X_{from_x+j} * Y_{to_y_rev−1−j} ;
31         end
32         c = (n*SUM_XY_{from_x,S−1,1} − SUM_X_{from_x} * SUM_Y_{S−1}) /
               sqrt(n*SUM_X2_{from_x} − SUM_X^2_{from_x}) * sqrt(n*SUM_Y2_{S−1} − SUM_Y^2_{S−1}) ;
33         if division by 0 occurred then
34             c = 0 ;
35         end
36         d = 1 − c ;
37         DIST_{from_x,from_y,1} = d ;
38         if d < min_dist then
39             min_distance = d ;
40             append a triplet (from_x, S − 1, 1) to MIN_DIST ;
41         end
42         else if d = min_dist then
43             append a triplet (from_x, S − 1, 1) to MIN_DIST ;
44         end
45     end
```

Algorithm 9: A routine of distanceFast() computing all remaining distances between X and Y. This function can access and modify all variables declared in distanceFast().

```
1  void fillRemaining(){
2      for i = 1 to S − 1 do
3          from_x = i ; to_x = from_x + n ;
4          for j = 1 to S − 1 do
5              from_y = j ;
6              to_y = from_y + n ;
7              /* forward orientation                                   */
8              SUM_XY_{from_x,from_y,0} = SUM_XY_{from_x−1,from_y−1,1} − X_{from_x−1} *
                 Y_{from_y−1} + X_{to_x−1} * Y to_y − 1 ;
9              c = (n*SUM_XYfrom_x,from_y,0 − SUM_Xfrom_x*SUM_Yfrom_y) /
                 (sqrt(n*SUM_X2_{from_x} − SUM_X²_{from_x}) * sqrt(n*SUM_Y2_{from_Y} − SUM_Y²_{from_Y})) ;
10             if division by 0 occurred then
11                 c = 0 ;
12             end
13             d = 1 − c ;
14             DIST_{from_x,from_y,0} = d ;
15             SCORES_{from_x,from_y} =
                 |l_half − (n_half + from_x)| + |l_half − (n_half + from_y)| ;
16             if d < min_dist then
17                 min_distance = d ;
18                 append a triplet (from_x, from_y, 0) to MIN_DIST ;
19             end
20             else if d = min_dist then
21                 append a triplet (from_x, from_y, 0) to MIN_DIST ;
22             end
23             /* reverse orientation                                   */
24             j_rev = S − j − 1 ;
25             from_y_rev = j_rev ;
26             to_y_rev = from_y_rev + n ;
27             SUM_XY_{from_x,from_y_rev,1} = SUM_XY_{from_x−1,from_y_rev+1,1} −
                 X_{from_x−1} * Y_{to_y_rev} + X_{to_x−1} * Y from_y_rev ;
28             c = (n*SUM_XY_{from_x,from_y_rev,1} − SUM_X_{from_x}*SUM_Y_{from_y_rev}) /
                 (sqrt(n*SUM_X2_{from_x} − SUM_X²_{from_x}) * sqrt(n*SUM_Y2_{from_y_rev} − SUM_Y²_{from_yrev})) ;
29             if division by 0 occurred then
30                 c = 0 ;
31             end
32             d = 1 − c ;
33             DIST_{from_x,from_y,1} = d ;
34             if d < min_dist then
35                 min_distance = d ;
36                 append a triplet (from_x, from_y_rev, 1) to MIN_DIST ;
37             end
38             else if d = min_dist then
39                 append a triplet (from_x, from_y_rev, 1) to MIN_DIST ;
40             end
41         end
42     end
```

Algorithm 10: Random seeding algorithm

```
1  REF seedingRandom(K, R){
       Data: K the number references to initialize from R, a 2D matrix of numericals of
             dimensions N × L.
       Result: REF a 2D matrix of numericals of dimensions K × L.
2      REF = a 2D matrix of dimensions K ∗ L ;
3      for i = 0 to K − 1 do
4          j = sample a number from 0 to N − 1 with uniform probabilities ;
5          while j has already been sampled before do
6              j = sample a number from 0 to N − 1 with uniform probabilities ;
7          end
8          REF_{i,} = R_{j,} ;
9      end
10     return REF
```

Algorithm 11: Kmeans++ seeding algorithm.

```
1  REF seedingKmeans++(K, R, S){
       Data: K the number references to initialize from R, a 2D matrix of numericals of
             dimensions N × L, allowing a shifting of S.
       Result: REF a 2D matrix of numericals of dimensions K × L.
2      REF = a 2D matrix of dimensions K ∗ L ;
3      DIST_TO_CLOSEST = a vector of size N;
4      /* choose first reference, a standard deviation equals to 0
          should be avoided otherwise computing a distance against this
          reference will always be equal to 1 (see how distance are
          computed).                                                    */
5      j = sample a number from 0 to N − 1 with uniform probabilities but for which
        DATA_{j,} has a standard deviation different from 0 ;
6      REF_{0,} = DATA_{j,} ;
7      /* choose next references                                        */
8      for k = 1 to K − 1 do
9          for i = 0 to N − 1 do
10             /* distance to all current references for this region    */
11             DIST_TO_CENTERS = a vector of size k ; for j = 0 to k − 1 do
12                 DIST_TO_CENTERS_{j,} = distanceFast(REF_{j,}, R_{i,}, S) ;
13             end
14             /* distance to closest current reference, for this region
                  */
15             DIST_TO_CLOSESTi, = minimum value of DIST_TO_CENTERS;
16         end
17         j = sample a number from 0 to N − 1 using DIST_TO_CLOSEST as the
            probability/weight distribution for each point to be sample. ;
18         while j has already been sampled before do
19             j = sample a number from 0 to N − 1 using DIST_TO_CLOSEST as the
                probability/weight distribution for each point to be sample. ;
20         end
21         REF_{k,} = Rj, ;
22     end
23     return REF
```

A.3 SMiLE-seq supplementary material

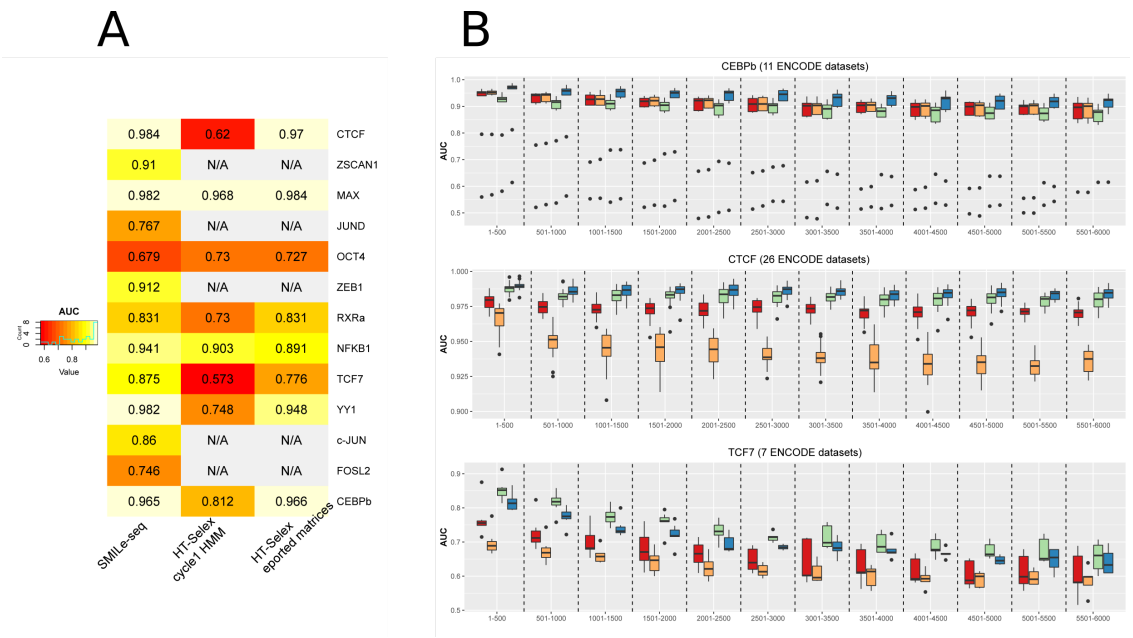


Figure A.11 – Predictive power of SMiLE-seq : A binding models were derived *de novo* from HT-SELEX 1st cycle data using the HMM discovery method (labelled HT-SELEX cycle 1 HMM) and their performances were assessed using the AUC-ROC. AUC-ROC values for the corresponding TF models derived from SMiLe-seq data (labelled SMiLE-seq) and reported by Jolma and colleagues (labelled HT-SELEX reported matrices, Jolma et al. (2013)) are also displayed. **B** the predictive performances of CEBPb, CTCF and TCF7 binding models were assessed using subsets of binding sites of decreasing affinities. Inside each peak list, the peaks were ranked by score and subsets of 500 peaks were selected. Peaks 1-500 have the highest affinity, then peaks 501-1000, and so on. The boxplots indicate the distribution of AUC-ROC obtained over all available peak-lists.

A.4 Chromatin accessibility of monocytes supplementary material

A.4.1 Fragment size analysis

Nucleosomes fragments are expected to be large, as a nucleosome is wrapped by $\tilde{1}$50bp of DNA whereas nucleosome free region fragments can be expected to be shorter. Long nucleosome free region fragments are unlikely. The longer an accessible region is, the most likely an insertion will happen resulting in the creation of two shorter fragments. A fragment size analysis allowed to identify different categories of fragments (Figure A.12). In this figure, open regions, mono- and di-nucleosome fragments are clearly visible. Morever, a 10bp periodicity oscillations reflecting the DNA pitch is also visible. This pattern is expected and indicates a good data quality (Buenrostro et al., 2013).

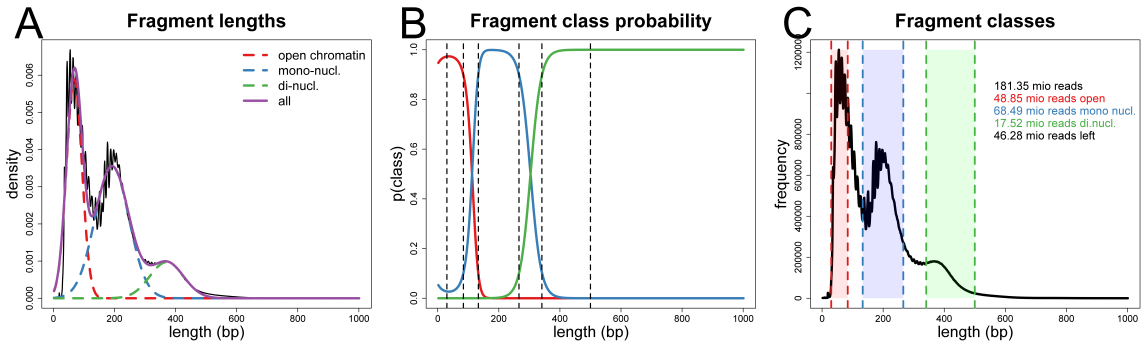


Figure A.12 – Fragment size analysis A sequenced fragment size density. The three peaks, from left to right, indicate i) the open chromatin fragments, ii) the mono-nucleosome fragments and iii) the di-nucleosome fragments. A mixture model composed of three Gaussian distributions was fitted to the data in order to model the fragment sizes. The class fit is shown as dashed lines : open chromatin (red), mono-nucleosomes (blue) and di-nucleosomes (green). The violet dashed line show the sum of the three classes. **B :** probability that a fragment belongs to any of the three fragment classes, given its size i) open chromatin (red), ii) mono-nucleosomes (blue) and iii) di-nucleosomes (green). The vertical dashed lines indicates, for each class, the size limit at which the class probability drops below 0.9. With these limites, the class spans are i) 30-84bp for open chromatin (red), ii) 133-266bp for mono-nucleosomes (blue) and iii) 341-500bp for di-nucleosomes (green). The upper limit of the di-nucleosome class was arbitrarily set to 500bp. **C :** final fragment classes. Each fragments which size overlapped the size range spanned by a class, was assigned to that class. This ensured a high confidence assignment for more than 134 million fragments, leaving 46 millions of ambiguous and long fragments (>500bp) unassigned.

Rather than assigning arbitrary fragment size threshold to separate the categories, I preferred to use the approach developed by (Buenrostro et al., 2013). The fragment sizes were fitted by a mixture of three Gaussian distributions. Then, the limits for each fragment class was defined as the size at which the probability of assignment to that fragment class dropped under 0.9 (Figure A.12B).

This method ensured the classification of 134 millions of fragments, leaving 46 millions reads unassigned (Figure A.12C). However, this reduces drastically the risks of fragment misclassification and protects the downstream analyses from a strong bias.

A.4.2 Measuring open chromatin and nucleosome occupancy

Once the different fragment populations have been identified, the next question to solve is how should each category of fragment be represented?

First, for open chromatin fragment, it is clear that we want to know where the DNA is accessible. This information is provided by the fragment edges – the transpositions sites. However, to

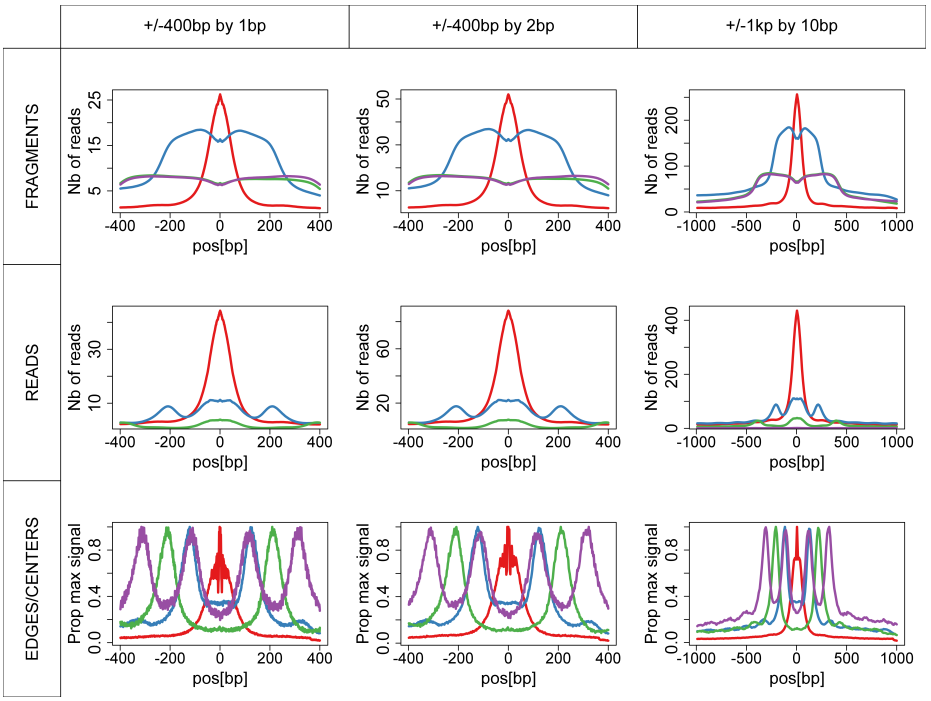


Figure A.13 – Signal around CTCF motif occurrences : the human genome was scanned with a CTCF PWM and different aggregated signal densities were measured for open chromatin (red lines), mono nucleosome (blue lines), di-nucleosomes (green lines) and for a pool of mono-nucleosome fragments with di-nucleosomes fragments cut in two at their center position (violet line). **Top row :** each position of the fragments, from the start of the first read to the end of the second, were used. **Middle row :** each position of the reads were used. **Bottom row :** only one position at the read edges for open chromatin fragment and the central position of nucleosome fragment were used. The open chromatin read edges were modified by +4bp and -5bp for +strand and -strand reads respectively.
The aggregated densities were measured using bin sizes of 1 (left column), 2 (middle column) and 10bp (right column).

account for the fact that the Tn5 transposase acts as a homo-dimer and inserts two barcodes side by side (Adey et al., 2010), the fragment edges positions were modified by +4bp for reads mapping the + strand and -5bp for reads mapping the - strand, as done in other studies (Buenrostro et al., 2013; Li et al., 2019).

Second, for mono and di-nucleosome fragments, we are interested in knowing where the nucleosomes are sitting. For this, the fragment edges may not be the most informative. A better way to represent those fragments would be to use the center positions, which should correspond to the dyad for mono-nucleosomes or even to consider the entire reads or fragments.

To test these hypotheses I investigated the different signal aggregations around predicted CTCF

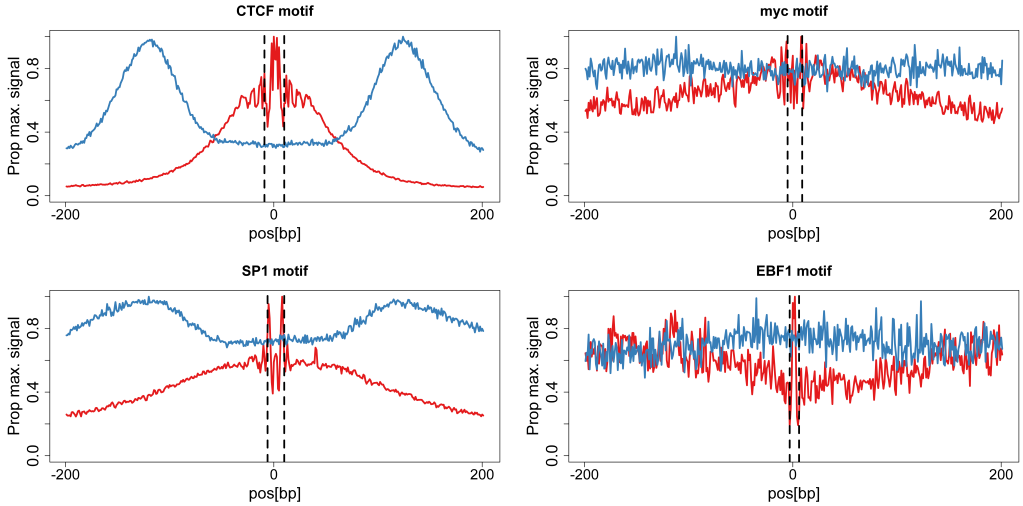


Figure A.14 – Signal around CTCF, SP1, myc and EBF1 motif occurrences : the human genome was scanned with one PWM per TF to predict their binding sites (see section 7.9.9). For each TF, the open chromatin accessibility was measured (red) as well as and the nucleosome occupancy (blue) around their predicted binding sites. For the chromatin accessibility, the corrected read edges were considered and for nucleosomes, the center of the fragments. The motif location is indicated by the dashed lines.

binding sites using. The signal, +/- 1kb around the motif occurrences, was aggregated inside bins of 1, 2 or 10bp size. CTCF predicted binding sites were good candidates because CTCF is know to bind mostly through its motif (Neph et al. (2012) and Figure 3.2). Additionally CTCF binding produces a really typical chromatin architecture with strongly positioned nucleosomes arrays (Fu et al., 2008) and leaves a footprint (Neph et al., 2012).

As seen in Figure A.13, entire open chromatin reads and fragments do not allow to visualize a footprint signature (upper and middle rows, red lines). Both of them, nonetheless highlight open chromatin regions. The footprint becomes visible when considering the edges of the open chromatin fragments (bottom row, red line). Increasing the bin size blurs it and eventually makes it disappear (10bp, lower right).

Regarding nucleosomes, considering the entire fragments blurs the signal (upper row, blue and green lines) and the entire reads reveal the region upstream and downstream of the nucleosomes (middle row, blue and green lines). The only way to obtain a precise nucleosomes occupancy information was to use the middle position of the mono-nucleosome fragments (bottom row, blue line). Interesting enough, the middle position of di-nucleosome fragments indicates the DNA linker between two adjacent nucleosomes but does not accumulate in open chromatin regions (bottom row, green line). This suggested that di-nucleosome fragments could be separated in two mono-nucleosome fragments. I tested this hypothesis by simply dividing a di-nucleosome fragment in two smaller ones, at its center position. I then pooled

these new fragments with the mono-nucleosome fragments to create a nucleosome fragment dataset. When looking at the middle of these fragments, they could perfectly reveal the nucleosomes directly adjacent to the CTCF motif. Additionally this nucleosome dataset was also able to reveal a second nucleosome in the arrays (bottom row, violet lines).

To further support these results, I also measured the chromatin organization (+5/-4 corrected read edges for open chromatin and center of the nucleosome fragments from the nucleosome fragment dataset) around SP1, myc and EBF1 binding motif occurrences as well. As shown in Figure A.14, the aggregation of the signal around the CTCF and SP1 motif occurrences show an enhanced accessibility on the motif occurrences as well as a clear footprint. Moreover, the footprint is in a nucleosome free region. The situation was different for myc and EBF1. Neither of the two aggregations showed a nucleosome free region, nor an increased accessibility around the motif occurrences. Regarding myc, even though its aggregation presented a signal compatible with a local protection of its motif, this was shallow in comparison of CTCF and SP1. Finally, EBF1 presented a somewhat decreased accessibility around its motif and a striking increase accessibility directly at the level of the motif occurrences.

CTCF and SP1 motif occurrences are supporting the fact that footprints and nucleosome occupancy can be revealed using this method. Together with myc and EBF1, they clearly show an heterogeneity of chromatin organizations, at least at the aggregation level.

There are many possible explanations for these results. One of them is that the aggregation hides the variability of the individual regions and that SP1 and CTCF present a more conserved organization around their motif than myc and EBF1. Another would that the most visible and obvious footprint reflect an stronger TF activity. However, one should remain cautious on the interpretation of aggregation patterns as the individual sites signal may interfere with each other, creating an artificial aggregation that does not exist at any individual site (Kundaje et al., 2012).

In the light of these results, I decided to use the +5/-4 corrected edges of the open chromatin reads to investigate footprints and the fragment centers of the newly created nucleosome dataset to investigate nucleosome occupancy. If not explicitly stated otherwise, the reader should consider that any signal was measured using this procedure.

A.4.3 Evaluation of EMSequence and ChIPPartitioning

It was important to assess the performances of the partitioning methods to discover sequence motifs and footprint classes.

EMSequence

In order to measure the ability of EMSequence to retrieve over-represented motifs from a set of sequences, I simulated 2'000 synthetic DNA sequences of 100bp long. The sequences

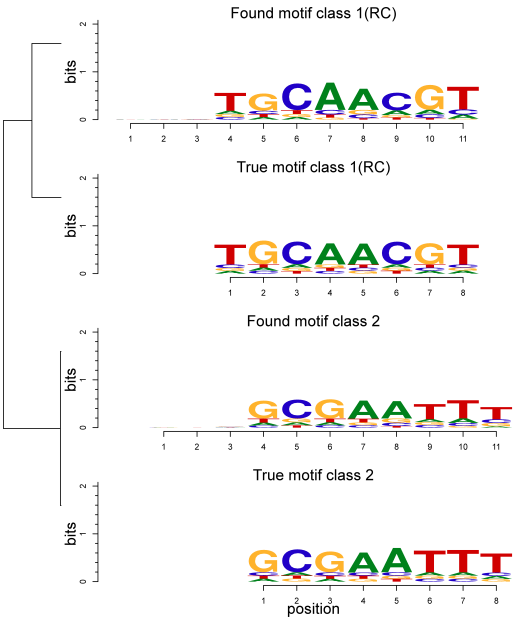


Figure A.15 – Simulated data motifs : motifs used for the data generation (labeled "True motif") and the best scoring - based on the AUC - partition motifs (labeled "Found motif"). The partition with EMSequence was run such that it was searching for motifs of 11bp, slightly longer than those used for the data generation. "RC" stands for reverse complement. The motifs tree and alignment was build using the motifStack R package (Ou et al., 2018).

were separated in two classes. Each class was defined by a 8bp sequence motif (Figure A.15). Each sequence had exactly one motif occurrence, anywhere in the sequence (with a uniform probability), on either strand (equiprobable).

These sequences were partitioned with flipping into 2 classes by EMSequence in order to find 2 motifs of 11bp ($100bp - 11bp + 1 = 90$bp of shifting). The optimization was run for 200 iterations. To assert the quality of the motifs discovered, I set up a classification framework inspired by PWMEval-ChIP-peak (see section 5.3). Using equation 5.1, each sequence was scored with both model of each partition and the area under the curve (AUC) of the receiver operator characteristic (ROC) value was computed for each partition. The same was done using the true motif models. Because EMSequence is sensitive to its initial state, 50 partitions were performed. As shown in Figure A.16, the *de novo* discovered models are as good as the actual sequence motifs to segregate both sequence classes. Additionally, a visual inspection of the discovered motif logo confirmed that most of the discovered motifs actually match the true sequence motifs (Figure A.15).

In order to further demonstrate the ability of EMSequence on a more significant biological case, I investigated SP1 sequence specificity. As for ChIPPartitioning, a list of 15'883 predicted SP1 binding sites were compiled using a PWM genome scan. The sequences +/- 400bp around the motif occurrences centers were extracted. Thus, all regions contained at least

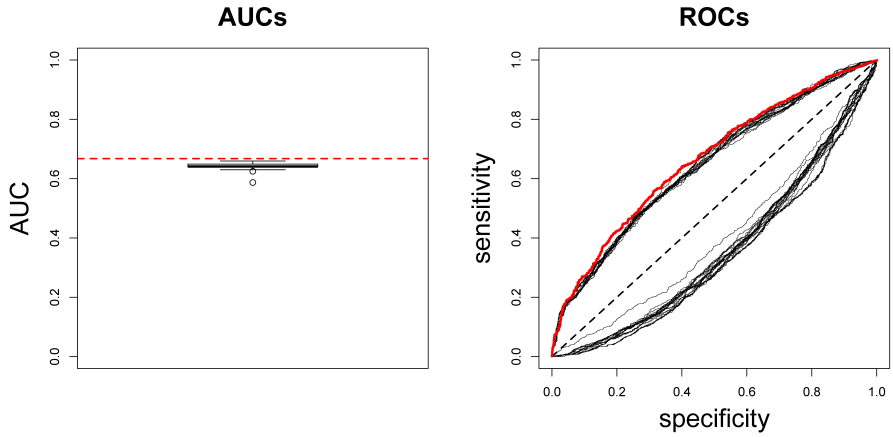


Figure A.16 – Classification performances on simulated data : Left 50 different data partitions were run using EMSequence. The discovered models were then used to assign a class label to each sequence. These assigned labels were then compared to the true labels using the AUC under the ROC curve. The red line indicates the AUC value achieved by the true motifs. **Right** the 50 ROC curves corresponding to each partition. The red lines indicates the true motifs ROC curve. The curves under the diagonal are the cases where the 1st discovered class corresponded to the 2nd true class and vice-versa. For these cases, the AUC is actually the area over the curve.

one SP1 site. Thus, retrieving the SP1 binding site is expected. Additionally, as SP1 tends to bind to promoters, we cannot exclude to see other motif being being discovered. These sequences were then given to EMSequence to search for several different 31bp long motifs ($801-31+1=771$ of shifting freedom). The optimization was run for 20 iterations.

The motifs that were retrieved matched the expectations (Figure A.17). All classes retrieved an SP1 motif. Four classes (1,5,6,7) retrieved a single SP1 motif. Even though they are highly similar, they vary in term of flanking regions (class 6 versus 7 for instance). Class 3, which contained a surprisingly long motif, representing 24% of the data, actually captured a LINE element. Indeed, the "GCAGCGAGGCTGGGGGAGGGGC" is characteristic of it (determined using BLAT (Kent, 2002) on the UCSC Genome Browser). Finally, and more interesting, classes 2 and 4 could capture two rare (about 1% of the cases each) tandem repeats of SP1 motifs with two different spacers (1 and 9bp). Additionally, head-to-head SP1 motif repeats could be detected (A.18, classes 1 and 4). This suggested that SP1 binds as i) an homo-dimer or ii) as an hetero-dimer with another member of its family, binding a resembling motif. Moreover, the tandem and heat-to-head motif repeats suggested that different structural arrangement exist. According to BioGrid (Chatr-aryamontri et al., 2017), SP1 has been reported to physically interact with SP1 (homo-dimer), SP3 and SP4 (hetero-dimer). According to JASPAR 2018 matrix clustering (Castro-Mondragon et al., 2017), the KLF and EGR families recognizes similar motifs. Members of either families are also listed as SP1 interactors in Biogrid (KLF4, KLF6, KLF9, KLF10 and EGR1).

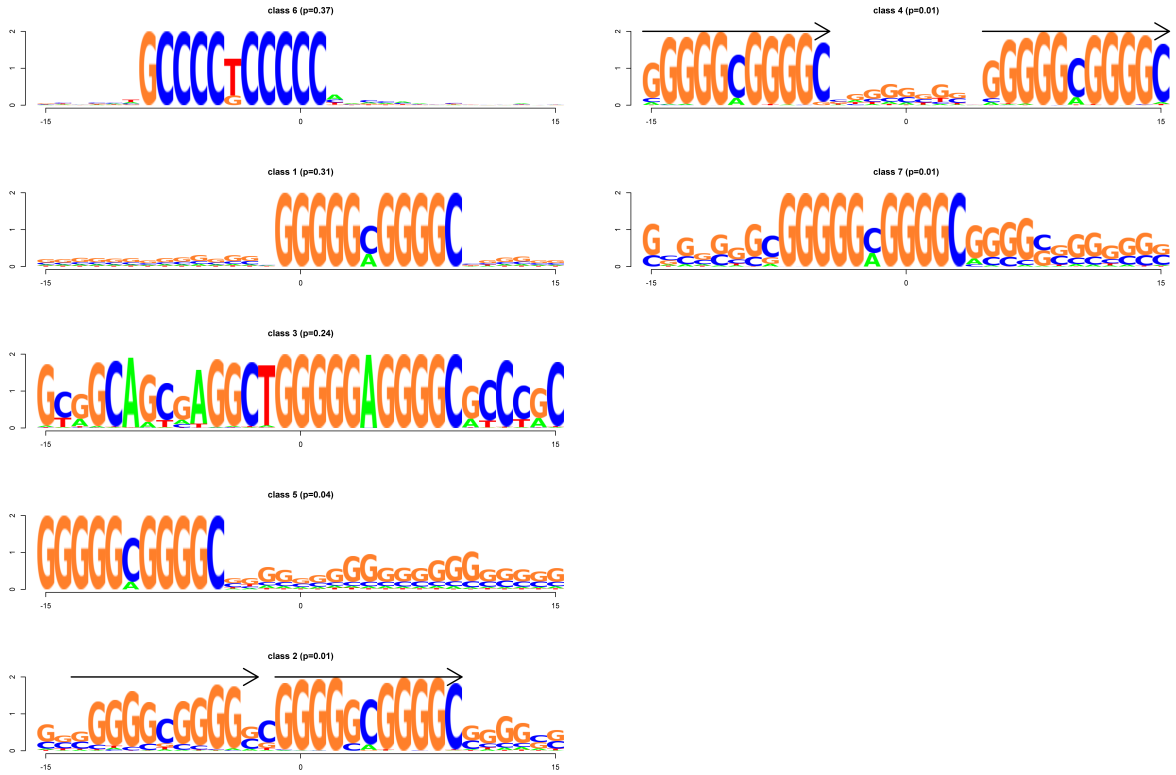


Figure A.17 – SP1 motifs : partition of 15'883 801bp sequences centered on a SP1 binding site using EMSequence. The different classes are ordered by decreasing overall probability. Arrows atop of the motifs indicates tandem arrangements of SP1 motifs.

The lack of non-SP1 motif discovered could be explained by at least one reason. The list of SP1 binding sites compiled was performed using a quite stringent threshold. The consequence is that the motif occurrences are highly similar to each other. This makes SP1 motifs strongly dominant within the dataset. Since EMSequence optimizes a set of models, it is highly sensitive to its starting state. In this experiment, EMSequence was initialised randomly. Because of the dominance of SP1 motifs within the data, it is likely that the different classes were attracted by them rather than allowed to diverge to detect other motifs.

Together, these evidences support the fact that EMSequence is suited to perform a meaningful partition of DNA sequences and to retrieve biologically important DNA motifs.

ChIPPartitioning

A complete benchmark of the ChIPPartitioning has been performed in (Nair et al., 2014). In this paper, the authors have generated simulated ChIP-seq data with patterns to retrieve, at different coverages and compared the performances with other similar software. It turned

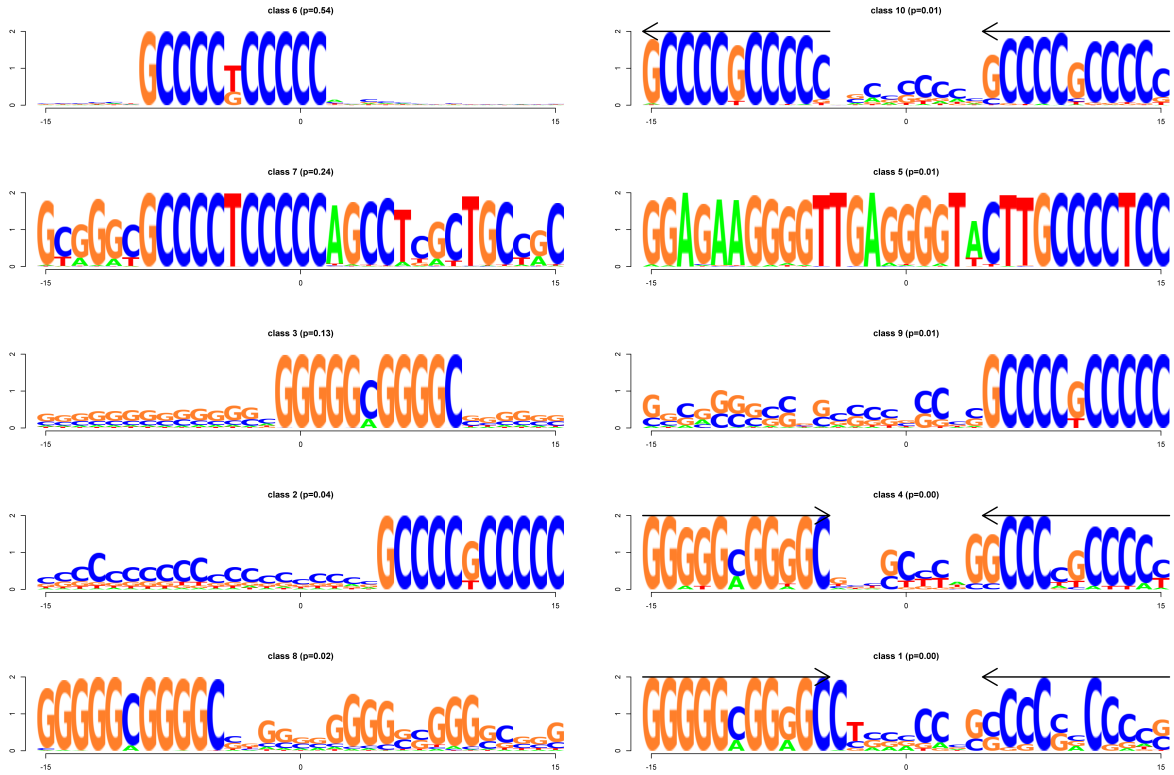


Figure A.18 – SP1 motifs : partition of 15'883 801bp sequences centered on a SP1 binding site using EMSequence. These sequences were classified by EMSequence to search for 10 different 30bp long motifs ($801-30=771$ of shifting freedom). The optimization was run for 20 iterations. The different classes are ordered by decreasing overall probability. Arrows atop of the motifs indicates head-to-tail arrangements of SP1 motifs.

out that ChIPPartitioning was the best performing method. For this reason, I did not repeat this benchmark. However, ChIPPartitioning ability to retrieve footprint classes from from ATAC-seq data has not been performed yet.

To evaluate this, a simple situation was considered. As in the previous section, a list of predicted CTCF and SP1 binding sites were compiled using a genome scan with suited binding models. For each TF, the open chromatin read density around these sites was measured +/- 400bp aroud the motif instances, at the single base pair resolution, and classified. As the motif instances were already aligned in the center of the regions, no shifting was used. However, the region orientations were not corrected based on the strand on which the motif instance appeared.

To evaluate the capability of ChIPPartitioning to retrieve classes of footprints, these data were classified i) without shifting and with flip (Figure A.19 and Figure A.20) and ii) with shifting and flipping (Figure A.21 and A.22).

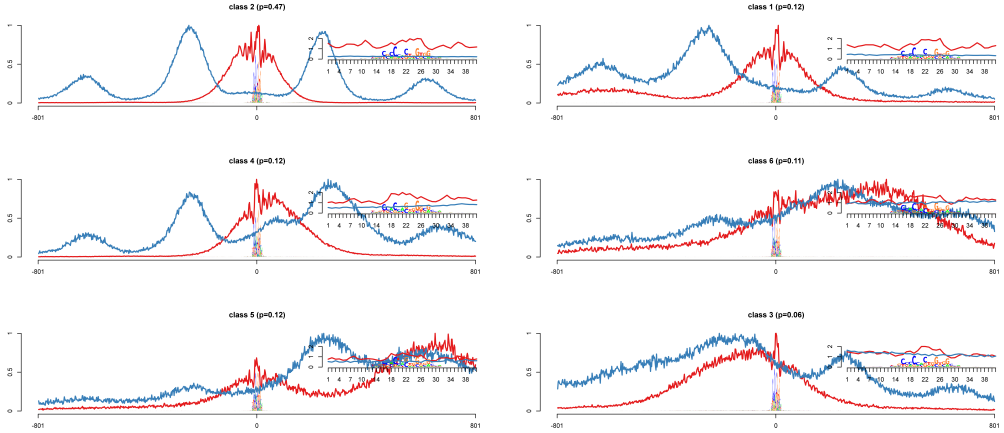


Figure A.19 – Open chromatin classes around CTCF motif occurrences found by ChIPPartitioning without shifing but with flipping to identify different classes of footprints around 26'650 CTCF motif occurrences. The aggregation signal around the 6 different classes found are shown by decreasing class probability. The open chromatin patterns are displayed in red, the nucleosomes are displayed in blue. The aggregated DNA sequence is displayed as a logo. The y-axis ranges from the minimum to the maximum signal observed. For the DNA logo, this corresponds to 0 and 2 bits respectively.

First, in both conditions - with and without shifting - different open chromatin signal classes have been discovered. Second, in most cases, the chromatin accessibility is anti-correlated with the nucleosome occupancy, which is something expected. However this is not always the case, such as in Figure A.19 classes 3 and 6. Such pattern may reflect a complex chromatin architecture, with variably positioned nucleosomes, that the partition cannot realign. But it is also likely to be an artifactual signal caused by the partition itself. Third, allowing the regions to be flipped based on the chromatin accessibility signature (Figure A.19 and Figure A.20) does not allow to resolve properly the orientation of the underlying CTCF and SP1 motif occurrences. Indeed, the sequence logos, in the center, are symetric indicating a superposition of +strand and -strand motif occurrences. Finally, allowing a moderated shifting freedom (+/- 10bp, Figure A.21 and A.22) results in blurred out sequence logo. This indicates that the chromatin accessibility signal realignment unphased the underlying CTCF and SP1 motif occurrences. Thus the signal that is observed does not represent classes of footprints.

In this case, each the region contained a motif occurrence at its center. Nonetheless, even a limited shifting according to the open chromatin signal resulted in the dephasing the underlying motif occurrence. Trying to resolve the motif occurrence orientation by allowing flipping according to the open chromatin was not more successful. Thus, discovering footprint classes from a highly unaligned set of regions does not seem to be possible. Workaround strategies have to be found.

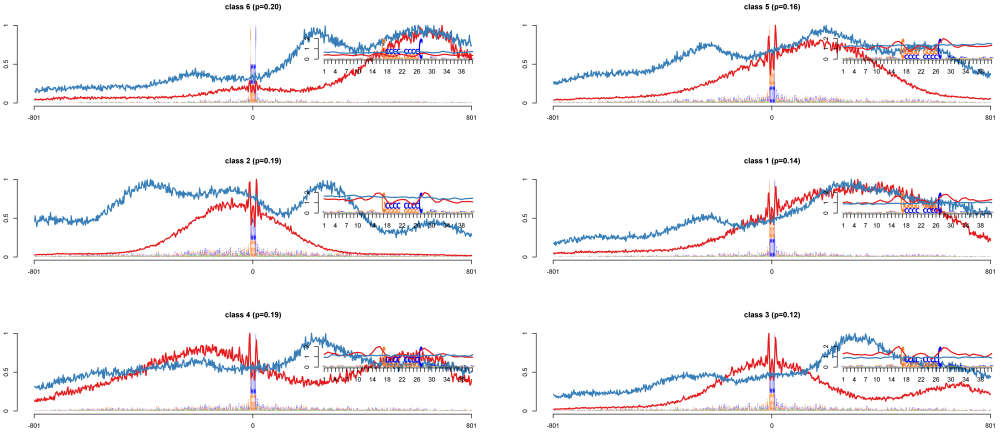
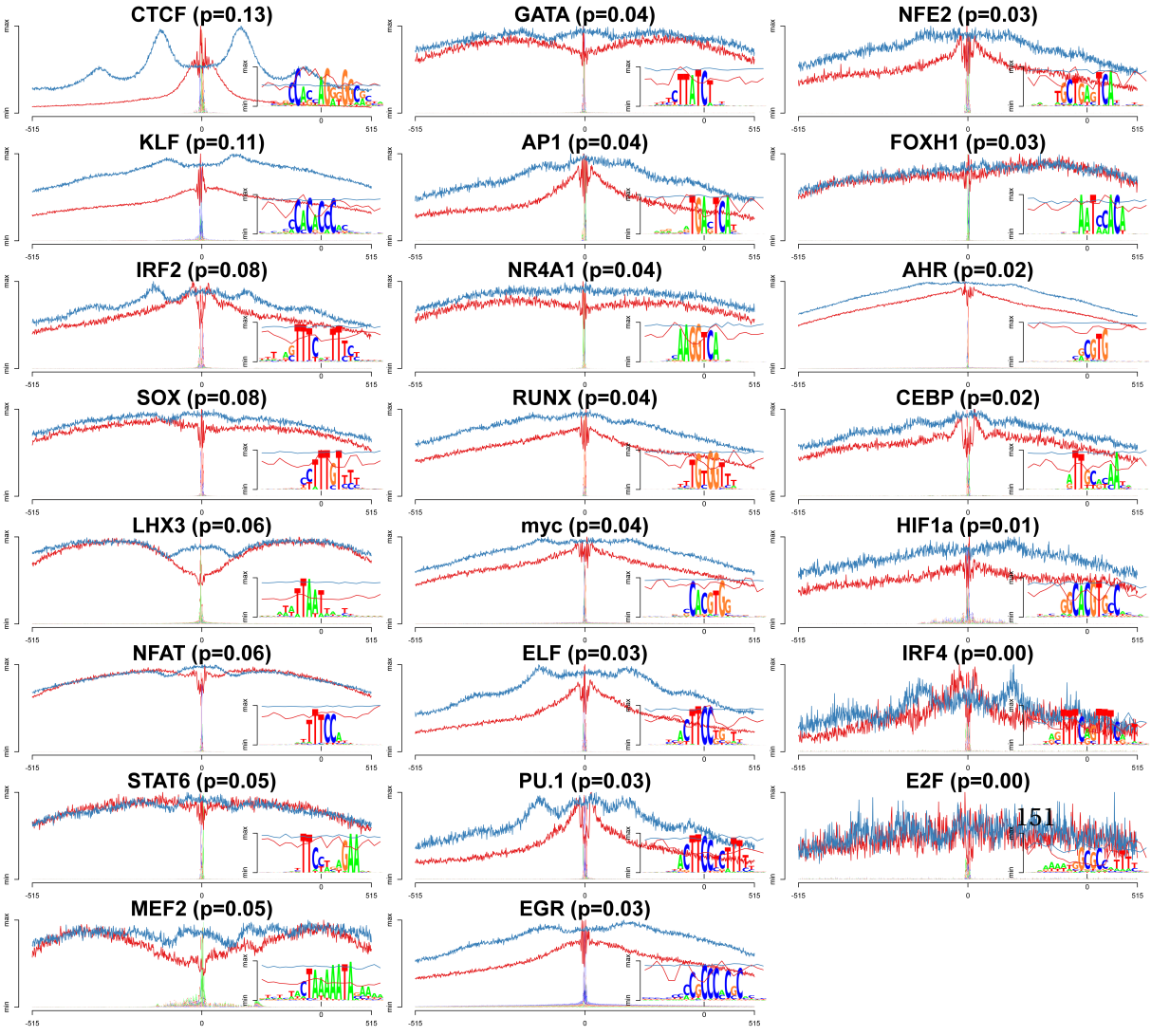


Figure A.20 – Open chromatin classes around SP1 motif occurrences : EMRead was run without shifing (+/- 10bp) but with flipping to identify different classes of footprints around 15'883 SP1 motif occurrences. The aggregation signal around the 6 different classes found are shown by decreasing class probability. The open chromatin patterns are displayed in red, the nucleosomes are displayed in blue. The aggregated DNA sequence is displayed as a logo. The y-axis ranges from the minimum to the maximum signal observed. For the DNA logo, this corresponds to 0 and 2 bits respectively.

A.4.4 Other supplementary figures



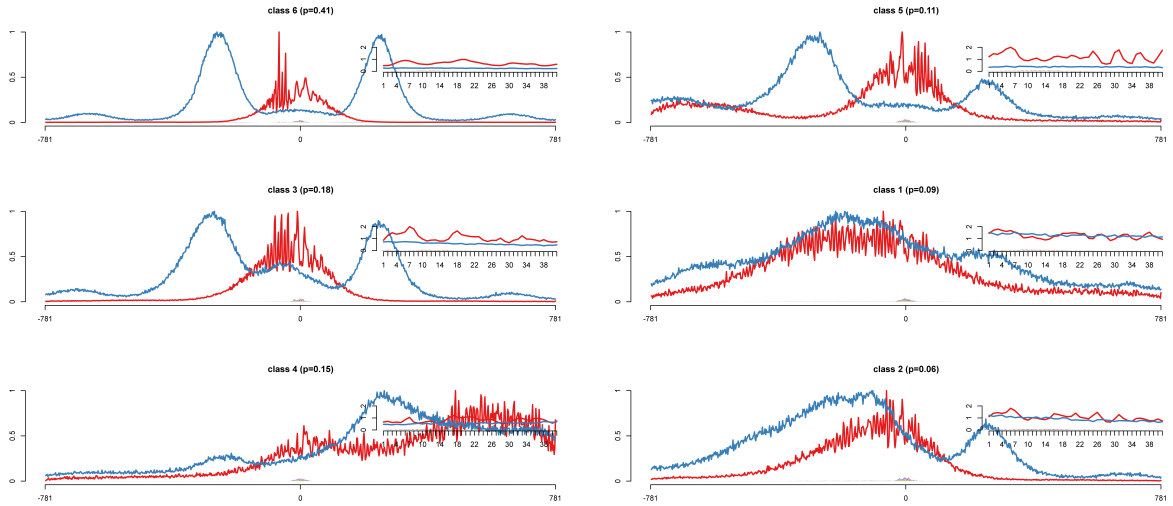


Figure A.21 – Open chromatin classes around CTCF motif occurrences found by ChIPPartitioning with shifing but with flipping to identify different classes of footprints around 26'650 CTCF motif occurrences. The aggregation signal around the 6 different classes found are shown by decreasing class probability. The open chromatin patterns are displayed in red, the nucleosomes are displayed in blue. The aggregated DNA sequence is displayed as a logo. The y-axis ranges from the minimum to the maximum signal observed. For the DNA logo, this corresponds to 0 and 2 bits respectively.

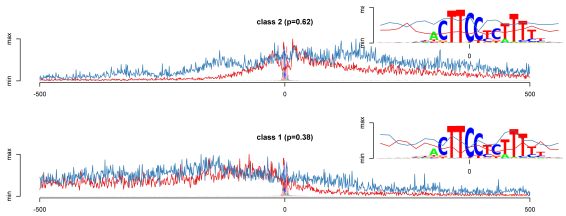


Figure A.24 – PU.1 sub-classes obtained by extracting PU.1 class data and subjecting them to a ChIPPartitioning classification into 2 classes. The displayed logos correspond to each class sequence aggregation. The corresponding chromatin accessibility (red) and nucleosome occupancy (blue) are displayed atop of the logos. The classes are displayed by overall decreasing probability. A zoom over the central part of each class aggregation is shown in the top right inlet.

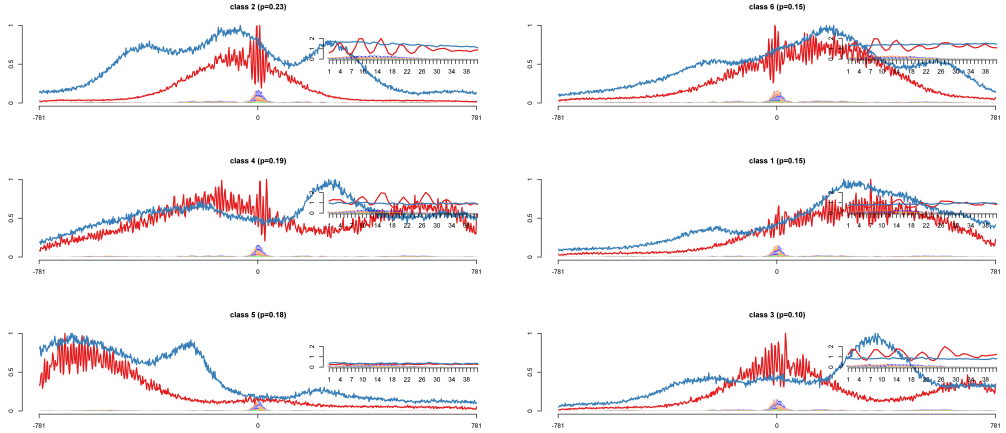


Figure A.22 – Open chromatin classes around SP1 motif occurrences : EMRead was run with shifing (+/- 10bp) flipping to identify different classes of footprints around 15'883 SP1 motif occurrences. The aggregation signal around the 6 different classes found are shown by decreasing class probability. The open chromatin patterns are displayed in red, the nucleosomes are displayed in blue. The aggregated DNA sequence is displayed as a logo. The y-axis ranges from the minimum to the maximum signal observed. For the DNA logo, this corresponds to 0 and 2 bits respectively.

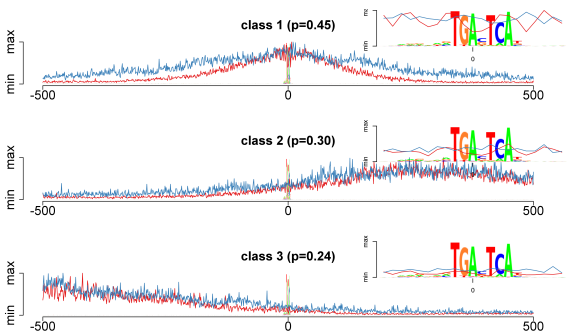


Figure A.25 – AP1 sub-classes obtained by extracting AP1 class data and subjecting them to a ChIPPartitioning classification into 3 classes. The displayed logos correspond to each class sequence aggregation. The corresponding chromatin accessibility (red) and nucleosome occupancy (blue) are displayed atop of the logos. The classes are displayed by overall decreasing probability. A zoom over the central part of each class aggregation is shown in the top right inlet.

Bibliography

Adey, A., Morrison, H. G., Asan, Xun, X., Kitzman, J. O., Turner, E. H., Stackhouse, B., MacKenzie, A. P., Caruccio, N. C., Zhang, X., and Shendure, J. (2010). Rapid, low-input, low-bias construction of shotgun fragment libraries by high-density in vitro transposition. *Genome Biology*, 11(12):R119.

Aerts, S., Thijs, G., Coessens, B., Staes, M., Moreau, Y., and Moor, B. D. (2003). Toucan: deciphering the cis -regulatory logic of coregulated genes. *Nucleic Acids Research*, 31(6):1753–1764.

Aibar, S., González-Blas, C. B., Moerman, T., Huynh-Thu, V. A., Imrichova, H., Hulselmans, G., Rambow, F., Marine, J.-C., Geurts, P., Aerts, J., van den Oord, J., Atak, Z. K., Wouters, J., and Aerts, S. (2017). SCENIC: single-cell regulatory network inference and clustering. *Nature Methods*, 14(11):1083–1086.

Ambrosini, G., Dreos, R., Kumar, S., and Bucher, P. (2016a). The ChIP-Seq tools and web server: a resource for analyzing ChIP-seq and other types of genomic data. *BMC Genomics*, 17:938.

Ambrosini, G., Dreos, R., Kumar, S., and Bucher, P. (2016b). The ChIP-Seq tools and web server: a resource for analyzing ChIP-seq and other types of genomic data. *BMC Genomics*, 17(1):938.

Ambrosini, G., Groux, R., and Bucher, P. (2018). PWMScan: a fast tool for scanning entire genomes with a position-specific weight matrix. *Bioinformatics*, 34(14):2483–2484.

Ambrosini, G., Praz, V., Jagannathan, V., and Bucher, P. (2003). Signal search analysis server. *Nucleic Acids Research*, 31(13):3618–3620.

Angerer, P., Simon, L., Tritschler, S., Wolf, F. A., Fischer, D., and Theis, F. J. (2017). Single cells make big data: New challenges and opportunities in transcriptomics. *Current Opinion in Systems Biology*, 4:85–91.

Arthur, D. and Vassilvitskii, S. (2007). K-means++: The Advantages of Careful Seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pages 1027–1035, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.

Athar, A., Füllgrabe, A., George, N., Iqbal, H., Huerta, L., Ali, A., Snow, C., Fonseca, N. A., Petryszak, R., Papatheodorou, I., Sarkans, U., and Brazma, A. (2019). ArrayExpress update – from bulk to single-cell expression data. *Nucleic Acids Research*, 47(D1):D711–D715.

Bailey, S. D., Zhang, X., Desai, K., Aid, M., Corradin, O., Cowper-Sal·lari, R., Akhtar-Zaidi, B., Scacheri, P. C., Haibe-Kains, B., and Lupien, M. (2015). ZNF143 provides sequence specificity to secure chromatin interactions at gene promoters. *Nature Communications*, 2:6186.

Bailey, T. L., Boden, M., Buske, F. A., Frith, M., Grant, C. E., Clementi, L., Ren, J., Li, W. W., and Noble, W. S. (2009). MEME Suite: tools for motif discovery and searching. *Nucleic Acids Research*, 37(suppl_2):W202–W208.

Bailey, T. L. and Elkan, C. (1994). Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proceedings. International Conference on Intelligent Systems for Molecular Biology*, 2:28–36.

Barrett, T., Wilhite, S. E., Ledoux, P., Evangelista, C., Kim, I. F., Tomashevsky, M., Marshall, K. A., Phillippy, K. H., Sherman, P. M., Holko, M., Yefanov, A., Lee, H., Zhang, N., Robertson, C. L., Serova, N., Davis, S., and Soboleva, A. (2013). NCBI GEO: archive for functional genomics data sets—update. *Nucleic Acids Research*, 41(D1):D991–D995.

Barski, A., Cuddapah, S., Cui, K., Roh, T.-Y., Schones, D. E., Wang, Z., Wei, G., Chepelev, I., and Zhao, K. (2007). High-Resolution Profiling of Histone Methylations in the Human Genome. *Cell*, 129(4):823–837.

Beckstette, M., Homann, R., Giegerich, R., and Kurtz, S. (2006). Fast index based algorithms and software for matching position specific scoring matrices. *BMC Bioinformatics*, 7:389.

Benos, P. V., Bulyk, M. L., and Stormo, G. D. (2002). Additivity in protein–DNA interactions: how good an approximation is it? *Nucleic Acids Research*, 30(20):4442–4451.

Berest, I., Arnold, C., Reyes-Palomares, A., Palla, G., Rasmussen, K. D., Helin, K., and Zaugg, J. (2018). Quantification of differential transcription factor activity and multiomics-based classification into activators and repressors: diffTF. *bioRxiv*.

Berg, O. G. and von Hippel, P. H. (1988). Selection of DNA binding sites by regulatory proteins. *Journal of Molecular Biology*, 200(4):709–723.

Berger, M. F. and Bulyk, M. L. (2009). Universal protein-binding microarrays for the comprehensive characterization of the DNA-binding specificities of transcription factors. *Nature Protocols*, 4(3):393–411.

Berger, M. F., Philippakis, A. A., Qureshi, A. M., He, F. S., Estep, P. W., and Bulyk, M. L. (2006). Compact, universal DNA microarrays to comprehensively determine transcription-factor binding site specificities. *Nature Biotechnology*, 24(11):1429–1435.

Boller, S., Li, R., and Grosschedl, R. (2018). Defining B Cell Chromatin: Lessons from EBF1. *Trends in Genetics*, 34(4):257–269.

Boller, S., Ramamoorthy, S., Akbas, D., Nechanitzky, R., Burger, L., Murr, R., Schübeler, D., and Grosschedl, R. (2016). Pioneering Activity of the C-Terminal Domain of EBF1 Shapes the Chromatin Landscape for B Cell Programming. *Immunity*, 44(3):527–541.

Bonev, B. and Cavalli, G. (2016). Organization and function of the 3D genome. *Nature Reviews Genetics*, 17(11):661–678.

Boyle, A. P., Davis, S., Shulha, H. P., Meltzer, P., Margulies, E. H., Weng, Z., Furey, T. S., and Crawford, G. E. (2008). High-Resolution Mapping and Characterization of Open Chromatin across the Genome. *Cell*, 132(2):311–322.

Bucher, P. and Trifonov, E. N. (1986). Compilation and analysis of eukaryotic POL II promoter sequences. *Nucleic Acids Research*, 14(24):10009–10026.

Buenrostro, J. D., Giresi, P. G., Zaba, L. C., Chang, H. Y., and Greenleaf, W. J. (2013). Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, DNA-binding proteins and nucleosome position. *Nature Methods*, 10(12):1213–1218.

Bulyk, M. L., Huang, X., Choo, Y., and Church, G. M. (2001). Exploring the DNA-binding specificities of zinc fingers with DNA microarrays. *Proceedings of the National Academy of Sciences*, 98(13):7158–7163.

Bulyk, M. L., Johnson, P. L. F., and Church, G. M. (2002). Nucleotides of transcription factor binding sites exert interdependent effects on the binding affinities of transcription factors. *Nucleic Acids Research*, 30(5):1255–1261.

Cairns, B. R. (2009). The logic of chromatin architecture and remodelling at promoters. *Nature*, 461(7261):193–198.

Castro-Mondragon, J. A., Jaeger, S., Thieffry, D., Thomas-Chollier, M., and van Helden, J. (2017). RSAT matrix-clustering: dynamic exploration and redundancy reduction of transcription factor binding motif collections. *Nucleic Acids Research*, 45(13):e119–e119.

Chatr-aryamontri, A., Oughtred, R., Boucher, L., Rust, J., Chang, C., Kolas, N. K., O'Donnell, L., Oster, S., Theesfeld, C., Sellam, A., Stark, C., Breitkreutz, B.-J., Dolinski, K., and Tyers, M. (2017). The BioGRID interaction database: 2017 update. *Nucleic Acids Research*, 45(D1):D369–D379.

Cheng, C., Alexander, R., Min, R., Leng, J., Yip, K. Y., Rozowsky, J., Yan, K.-K., Dong, X., Djebali, S., Ruan, Y., Davis, C. A., Carninci, P., Lassman, T., Gingeras, T. R., Guigó, R., Birney, E., Weng, Z., Snyder, M., and Gerstein, M. (2012). Understanding transcriptional regulation by integrative analysis of transcription factor binding data. *Genome Research*, 22(9):1658–1667.

Cirillo, L. A., Lin, F. R., Cuesta, I., Friedman, D., Jarnik, M., and Zaret, K. S. (2002). Opening of Compacted Chromatin by Early Developmental Transcription Factors HNF3 (FoxA) and GATA-4. *Molecular Cell*, 9(2):279–289.

Consortium, T. E. P. (2012). An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74.

Dalton, L., Ballarin, V., and Brun, M. (2009). Clustering Algorithms: On Learning, Validation, Performance, and Applications to Genomics. *Current Genomics*, 10(6):430–445.

Donohoe, M. E., Zhang, L.-F., Xu, N., Shi, Y., and Lee, J. T. (2007). Identification of a Ctcf Cofactor, Yy1, for the X Chromosome Binary Switch. *Molecular Cell*, 25(1):43–56.

Dreos, R., Ambrosini, G., and Bucher, P. (2016). Influence of Rotational Nucleosome Positioning on Transcription Start Site Selection in Animal Promoters. *PLOS Computational Biology*, 12(10):e1005144.

Dreos, R., Ambrosini, G., Cavin Périer, R., and Bucher, P. (2013). EPD and EPDnew, high-quality promoter resources in the next-generation sequencing era. *Nucleic Acids Research*, 41(D1):D157–D164.

Dreos, R., Ambrosini, G., Groux, R., Cavin Périer, R., and Bucher, P. (2017). The eukaryotic promoter database in its 30th year: focus on non-vertebrate organisms. *Nucleic Acids Research*, 45(D1):D51–D55.

Dreos, R., Ambrosini, G., Groux, R., Périer, R. C., and Bucher, P. (2018). MGA repository: a curated data resource for ChIP-seq and other genome annotated data. *Nucleic Acids Research*, 46(D1):D175–D180.

Dreos, R., Ambrosini, G., Périer, R. C., and Bucher, P. (2015). The Eukaryotic Promoter Database: expansion of EPDnew and new promoter analysis tools. *Nucleic Acids Research*, 43(D1):D92–D96.

Fan, J., Salathia, N., Liu, R., Kaeser, G. E., Yung, Y. C., Herman, J. L., Kaper, F., Fan, J.-B., Zhang, K., Chun, J., and Kharchenko, P. V. (2016). Characterizing transcriptional heterogeneity through pathway and gene set overdispersion analysis. *Nature Methods*, 13(3):241–244.

Field, Y., Sharon, E., and Segal, E. (2011). Methods for Analysis of Transcription Factor DNA-Binding Specificity In Vitro, Chapter 9, How Transcription Factors Identify Regulatory Sites in Genomic Sequence. In Hughes, T. R., editor, *A Handbook of Transcription Factors*, number 52 in Subcellular Biochemistry, pages 193–204. Springer Netherlands.

Fu, Y., Frith, M. C., Haverty, P. M., and Weng, Z. (2004). MotifViz: an analysis and visualization tool for motif discovery. *Nucleic Acids Research*, 32(suppl_2):W420–W423.

Fu, Y., Sinha, M., Peterson, C. L., and Weng, Z. (2008). The Insulator Binding Protein CTCF Positions 20 Nucleosomes around Its Binding Sites across the Human Genome. *PLOS Genetics*, 4(7):e1000138.

Gaffney, D. J., McVicker, G., Pai, A. A., Fondufe-Mittendorf, Y. N., Lewellen, N., Michelini, K., Widom, J., Gilad, Y., and Pritchard, J. K. (2012). Controls of Nucleosome Positioning in the Human Genome. *PLoS Genet*, 8(11):e1003036.

Geertz, M., Shore, D., and Maerkl, S. J. (2012). Massively parallel measurements of molecular interaction kinetics on a microfluidic platform. *Proceedings of the National Academy of Sciences*, 109(41):16540–16545.

Gerstein, M. B., Kundaje, A., Hariharan, M., Landt, S. G., Yan, K.-K., Cheng, C., Mu, X. J., Khurana, E., Rozowsky, J., Alexander, R., Min, R., Alves, P., Abyzov, A., Addleman, N., Bhardwaj, N., Boyle, A. P., Cayting, P., Charos, A., Chen, D. Z., Cheng, Y., Clarke, D., Eastman, C., Euskirchen, G., Frietze, S., Fu, Y., Gertz, J., Grubert, F., Harmanci, A., Jain, P., Kasowski, M., Lacroute, P., Leng, J., Lian, J., Monahan, H., O'Geen, H., Ouyang, Z., Partridge, E. C., Patacsil, D., Pauli, F., Raha, D., Ramirez, L., Reddy, T. E., Reed, B., Shi, M., Slifer, T., Wang, J., Wu, L., Yang, X., Yip, K. Y., Zilberman-Schapira, G., Batzoglou, S., Sidow, A., Farnham, P. J., Myers, R. M., Weissman, S. M., and Snyder, M. (2012). Architecture of the human regulatory network derived from ENCODE data. *Nature*, 489(7414):91–100.

Ghirlando, R. and Felsenfeld, G. (2016). CTCF: making the right connections. *Genes & Development*, 30(8):881–891.

González-Blas, C. B., Minnoye, L., Papasokrati, D., Aibar, S., Hulselmans, G., Christiaens, V., Davie, K., Wouters, J., and Aerts, S. (2019). cisTopic: cis-regulatory topic modeling on single-cell ATAC-seq data. *Nature Methods*, 16(5):397.

Grant, C. E., Bailey, T. L., and Noble, W. S. (2011). FIMO: scanning for occurrences of a given motif. *Bioinformatics*, 27(7):1017–1018.

Grossman, S. R., Engreitz, J., Ray, J. P., Nguyen, T. H., Hacohen, N., and Lander, E. S. (2018). Positional specificity of different transcription factor classes within enhancers. *Proceedings of the National Academy of Sciences*, 115(30):E7222–E7230.

Groux, R. and Bucher, P. (2019). SPar-K: a method to partition NGS signal data. *Bioinformatics*.

Guo, Y., Mahony, S., and Gifford, D. K. (2012). High Resolution Genome Wide Binding Event Finding and Motif Discovery Reveals Transcription Factor Spatial Binding Constraints. *PLOS Computational Biology*, 8(8):e1002638.

Hagman, J. and Lukin, K. (2005). Early B-cell factor 'pioneers' the way for B-cell development. *Trends in Immunology*, 26(9):455–461.

Heinz, S., Benner, C., Spann, N., Bertolino, E., Lin, Y. C., Laslo, P., Cheng, J. X., Murre, C., Singh, H., and Glass, C. K. (2010). Simple Combinations of Lineage-Determining Transcription Factors Prime cis-Regulatory Elements Required for Macrophage and B Cell Identities. *Molecular Cell*, 38(4):576–589.

Heinz, S., Romanoski, C. E., Benner, C., and Glass, C. K. (2015). The selection and function of cell type-specific enhancers. *Nature Reviews Molecular Cell Biology*, 16(3):144–154.

Henikoff, S. and Smith, M. M. (2015). Histone Variants and Epigenetics. *Cold Spring Harbor Perspectives in Biology*, 7(1):a019364.

Herrera, J. E. and Chaires, J. B. (1994). Characterization of Preferred Deoxyribonuclease I Cleavage Sites. *Journal of Molecular Biology*, 236(2):405–411.

Hertz, G. Z., Hartzell, G. W., and Stormo, G. D. (1990). Identification of consensus patterns in unaligned DNA sequences known to be functionally related. *Computer applications in the biosciences: CABIOS*, 6(2):81–92.

Hon, G., Ren, B., and Wang, W. (2008). ChromaSig: A Probabilistic Approach to Finding Common Chromatin Signatures in the Human Genome. *PLOS Computational Biology*, 4(10):e1000201.

Hyun, K., Jeon, J., Park, K., and Kim, J. (2017). Writing, erasing and reading histone lysine methylations. *Experimental & Molecular Medicine*, 49(4):e324–e324.

Imbeault, M., Helleboid, P.-Y., and Trono, D. (2017). KRAB zinc-finger proteins contribute to the evolution of gene regulatory networks. *Nature*, 543(7646):550–554.

Ioshikhes, I., Hosid, S., and Pugh, B. F. (2011). Variety of genomic DNA patterns for nucleosome positioning. *Genome Research*, 21(11):1863–1871.

Isakova, A., Groux, R., Imbeault, M., Rainer, P., Alpern, D., Dainese, R., Ambrosini, G., Trono, D., Bucher, P., and Deplancke, B. (2017). SMiLE-seq identifies binding motifs of single and dimeric transcription factors. *Nature Methods*, advance online publication.

Iwafuchi-Doi, M. and Zaret, K. S. (2014). Pioneer transcription factors in cell reprogramming. *Genes & Development*, 28(24):2679–2692.

Jiang, C. and Pugh, B. F. (2009). Nucleosome positioning and gene regulation: advances through genomics. *Nature Reviews Genetics*, 10(3):161–172.

Jin, C., Zang, C., Wei, G., Cui, K., Peng, W., Zhao, K., and Felsenfeld, G. (2009). H3.3/H2A.Z double variant–containing nucleosomes mark 'nucleosome-free regions' of active promoters and other regulatory regions. *Nature Genetics*, 41(8):941–945.

Jolma, A., Kivioja, T., Toivonen, J., Cheng, L., Wei, G., Enge, M., Taipale, M., Vaquerizas, J. M., Yan, J., Sillanpää, M. J., Bonke, M., Palin, K., Talukder, S., Hughes, T. R., Luscombe, N. M., Ukkonen, E., and Taipale, J. (2010). Multiplexed massively parallel SELEX for characterization of human transcription factor binding specificities. *Genome Research*, 20(6):861–873.

Jolma, A., Yan, J., Whitington, T., Toivonen, J., Nitta, K., Rastas, P., Morgunova, E., Enge, M., Taipale, M., Wei, G., Palin, K., Vaquerizas, J., Vincentelli, R., Luscombe, N., Hughes, T., Lemaire, P., Ukkonen, E., Kivioja, T., and Taipale, J. (2013). DNA-Binding Specificities of Human Transcription Factors. *Cell*, 152(1–2):327–339.

Jothi, R., Cuddapah, S., Barski, A., Cui, K., and Zhao, K. (2008). Genome-wide identification of in vivo protein–DNA binding sites from ChIP-Seq data. *Nucleic Acids Research*, 36(16):5221–5231.

Kent, W. J. (2002). BLAT—The BLAST-Like Alignment Tool. *Genome Research*, 12(4):656–664.

Khan, A., Fornes, O., Stigliani, A., Gheorghe, M., Castro-Mondragon, J. A., van der Lee, R., Bessy, A., Chèneby, J., Kulkarni, S. R., Tan, G., Baranasic, D., Arenillas, D. J., Sandelin, A., Vandepoele, K., Lenhard, B., Ballester, B., Wasserman, W. W., Parcy, F., and Mathelier, A. (2018). JASPAR 2018: update of the open-access database of transcription factor binding profiles and its web framework. *Nucleic Acids Research*, 46(D1):D260–D266.

Kiselev, V. Y., Kirschner, K., Schaub, M. T., Andrews, T., Yiu, A., Chandra, T., Natarajan, K. N., Reik, W., Barahona, M., Green, A. R., and Hemberg, M. (2017). SC3: consensus clustering of single-cell RNA-seq data. *Nature Methods*, 14(5):483–486.

Klemm, S. L., Shipony, Z., and Greenleaf, W. J. (2019). Chromatin accessibility and the regulatory epigenome. *Nature Reviews Genetics*, 20(4):207–220.

Kouzarides, T. (2007). Chromatin Modifications and Their Function. *Cell*, 128(4):693–705.

Kubik, S., Bruzzone, M., Jacquet, P., Falcone, J.-L., Rougemont, J., and Shore, D. (2015). Nucleosome Stability Distinguishes Two Different Promoter Types at All Protein-Coding Genes in Yeast. *Molecular Cell*, 60(3):422–434.

Kulakovskiy, I. V., Vorontsov, I. E., Yevshin, I. S., Sharipov, R. N., Fedorova, A. D., Rumynskiy, E. I., Medvedeva, Y. A., Magana-Mora, A., Bajic, V. B., Papatsenko, D. A., Kolpakov, F. A., and Makeev, V. J. (2018). HOCOMOCO: towards a complete collection of transcription factor binding models for human and mouse via large-scale ChIP-Seq analysis. *Nucleic Acids Research*, 46(D1):D252–D259.

Kulakovskiy, I. V., Vorontsov, I. E., Yevshin, I. S., Soboleva, A. V., Kasianov, A. S., Ashoor, H., Baalawi, W., Bajic, V. B., Medvedeva, Y. A., Kolpakov, F. A., and Makeev, V. J. (2016). HOCOMOCO: expansion and enhancement of the collection of transcription factor binding sites models. *Nucleic Acids Research*, 44(D1):D116–D125.

Kundaje, A., Kyriazopoulou-Panagiotopoulou, S., Libbrecht, M., Smith, C. L., Raha, D., Winters, E. E., Johnson, S. M., Snyder, M., Batzoglou, S., and Sidow, A. (2012). Ubiquitous heterogeneity and asymmetry of the chromatin environment at regulatory elements. *Genome Research*, 22(9):1735–1747.

Kurotaki, D., Sasaki, H., and Tamura, T. (2017). Transcriptional control of monocyte and macrophage development. *International Immunology*, 29(3):97–107.

Lai, W. K. and Buck, M. J. (2010). ArchAlign: coordinate-free chromatin alignment reveals novel architectures. *Genome Biology*, 11:R126.

Langmead, B. and Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. *Nature Methods*, 9(4):357–359.

Langmead, B., Trapnell, C., Pop, M., and Salzberg, S. L. (2009). Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology*, 10(3):R25.

Latchman, D. S. (1997). Transcription factors: An overview. *The International Journal of Biochemistry & Cell Biology*, 29(12):1305–1312.

Lawrence, C. E. and Reilly, A. A. (1990). An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins*, 7(1):41–51.

Leisch, F. (2006). A Toolbox for K-Centroids Cluster Analysis.

Lenhard, B., Sandelin, A., and Carninci, P. (2012). Metazoan promoters: emerging characteristics and insights into transcriptional regulation. *Nature Reviews Genetics*, 13(4):233–245.

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., and Durbin, R. (2009). The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, 25(16):2078–2079.

Li, Z., Schulz, M. H., Look, T., Begemann, M., Zenke, M., and Costa, I. G. (2019). Identification of transcription factor binding sites using ATAC-seq. *Genome Biology*, 20(1):45.

Lizio, M., Harshbarger, J., Shimoji, H., Severin, J., Kasukawa, T., Sahin, S., Abugessaisa, I., Fukuda, S., Hori, F., Ishikawa-Kato, S., Mungall, C. J., Arner, E., Baillie, J. K., Bertin, N., Bono, H., de Hoon, M., Diehl, A. D., Dimont, E., Freeman, T. C., Fujieda, K., Hide, W., Kaliyaperumal, R., Katayama, T., Lassmann, T., Meehan, T. F., Nishikata, K., Ono, H., Rehli, M., Sandelin, A., Schultes, E. A., 't Hoen, P. A., Tatum, Z., Thompson, M., Toyoda, T., Wright, D. W., Daub, C. O., Itoh, M., Carninci, P., Hayashizaki, Y., Forrest, A. R., Kawaji, H., and the FANTOM consortium (2015). Gateways to the FANTOM5 promoter level mammalian expression atlas. *Genome Biology*, 16(1):22.

Losada, A. (2014). Cohesin in cancer: chromosome segregation and beyond. *Nature Reviews Cancer*, 14(6):389–393.

Längst, G. and Manelyte, L. (2015). Chromatin Remodelers: From Function to Dysfunction. *Genes*, 6(2):299–324.

Maerkl, S. J. and Quake, S. R. (2007). A Systems Approach to Measuring the Binding Energy Landscapes of Transcription Factors. *Science*, 315(5809):233–237.

Maier, H., Ostraat, R., Gao, H., Fields, S., Shinton, S. A., Medina, K. L., Ikawa, T., Murre, C., Singh, H., Hardy, R. R., and Hagman, J. (2004). Early B cell factor cooperates with Runx1 and mediates epigenetic changes associated with mb-1 transcription. *Nature Immunology*, 5(10):1069–1077.

Man, T.-K. and Stormo, G. D. (2001). Non-independence of Mnt repressor–operator interaction determined by a new quantitative multiple fluorescence relative affinity (QuMFRA) assay. *Nucleic Acids Research*, 29(12):2471–2478.

Marsland, S. (2015). *Machine Learning, An algorithmic Perspective, Chapter 7 Probabilistic Learning*. CRC Press, Boca Raton, second edition edition.

Mathelier, A., Zhao, X., Zhang, A. W., Parcy, F., Worsley-Hunt, R., Arenillas, D. J., Buchman, S., Chen, C.-y., Chou, A., Ienasescu, H., Lim, J., Shyr, C., Tan, G., Zhou, M., Lenhard, B., Sandelin, A., and Wasserman, W. W. (2014). JASPAR 2014: an extensively expanded and updated open-access database of transcription factor binding profiles. *Nucleic Acids Research*, 42(D1):D142–D147.

McGinty Robert K. and Tan Song (2014). *Fundamentals of Chromatin, chapter 1 Histone, Nucleosomes and Chromatin Structure*. Jerry L. Workman and Susan M. Abmayr, New York, 2014 edition.

Meylan, P., Dreos, R., Ambrosini, G., Groux, R., and Bucher, P. (2020). EPD in 2020: enhanced data visualization and extension to ncRNA promoters. *Nucleic Acids Research*, 48(D1):D65–D69.

Mukherjee, S., Berger, M. F., Jona, G., Wang, X. S., Muzzey, D., Snyder, M., Young, R. A., and Bulyk, M. L. (2004). Rapid analysis of the DNA-binding specificities of transcription factors with DNA microarrays. *Nature Genetics*, 36(12):1331–1339.

Nair, N. U., Kumar, S., Moret, B. M. E., and Bucher, P. (2014). Probabilistic partitioning methods to find significant patterns in ChIP-Seq data. *Bioinformatics*, 30(17):2406–2413.

Neph, S., Vierstra, J., Stergachis, A. B., Reynolds, A. P., Haugen, E., Vernot, B., Thurman, R. E., John, S., Sandstrom, R., Johnson, A. K., Maurano, M. T., Humbert, R., Rynes, E., Wang, H., Vong, S., Lee, K., Bates, D., Diegel, M., Roach, V., Dunn, D., Neri, J., Schafer, A., Hansen, R. S., Kutyavin, T., Giste, E., Weaver, M., Canfield, T., Sabo, P., Zhang, M., Balasundaram, G., Byron, R., MacCoss, M. J., Akey, J. M., Bender, M. A., Groudine, M., Kaul, R., and Stamatoyannopoulos, J. A. (2012). An expansive human regulatory lexicon encoded in transcription factor footprints. *Nature*, 489(7414):83–90.

Nielsen, F. G. G., Markus, K. G., Friborg, R. M., Favrholdt, L. M., Stunnenberg, H. G., and Huynen, M. (2012). CATCHprofiles: Clustering and Alignment Tool for ChIP Profiles. *PLOS ONE*, 7(1):e28272.

Odom, D. T. (2011). Methods for Analysis of Transcription Factor DNA-Binding Specificity In Vitro, Chapter 8, Identification of Transcription Factor–DNA Interactions In Vivo. In Hughes, T. R., editor, *A Handbook of Transcription Factors*, number 52 in Subcellular Biochemistry, pages 175–191. Springer Netherlands.

Ong, C.-T. and Corces, V. G. (2014). CTCF: an architectural protein bridging genome topology and function. *Nature Reviews Genetics*, 15(4):234–246.

Orenstein, Y. and Shamir, R. (2014). A comparative analysis of transcription factor binding models learned from PBM, HT-SELEX and ChIP data. *Nucleic Acids Research*, 42(8):e63–e63.

Ou, J., Wolfe, S. A., Brodsky, M. H., and Zhu, L. J. (2018). motifStack for the analysis of transcription factor binding site evolution. *Nature Methods*, 15(1):8–9.

Pizzi, C. and Ukkonen, E. (2008). Fast profile matching algorithms — A survey. *Theoretical Computer Science*, 395(2):137–157.

Pollard, K. S., Hubisz, M. J., Rosenbloom, K. R., and Siepel, A. (2010). Detection of nonneutral substitution rates on mammalian phylogenies. *Genome Research*, 20(1):110–121.

Quinlan, A. R. and Hall, I. M. (2010). BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, 26(6):841–842.

Raney, B. J., Dreszer, T. R., Barber, G. P., Clawson, H., Fujita, P. A., Wang, T., Nguyen, N., Paten, B., Zweig, A. S., Karolchik, D., and Kent, W. J. (2014). Track data hubs enable visualization of user-defined genome-wide annotations on the UCSC Genome Browser. *Bioinformatics*, 30(7):1003–1005.

Rico, D., Martens, J. H., Downes, K., Carrillo-de Santa-Pau, E., Pancaldi, V., Breschi, A., Richardson, D., Heath, S., Saeed, S., Frontini, M., Chen, L., Watt, S., Müller, F., Clarke, L., Kerstens, H. H., Wilder, S. P., Palumbo, E., Djebali, S., Raineri, E., Merkel, A., Esteve-Codina, A., Sultan, M., Bommel, A. v., Gut, M., Yaspo, M.-L., Rubio, M., Fernandez, J. M., Attwood, A., Torre, V. d. l., Royo, R., Fragkogianni, S., Gelpí, J. L., Torrents, D., Iotchkova, V., Logie, C., Aghajanirefah, A., Singh, A. A., Janssen-Megens, E. M., Berentsen, K., Erber, W., Rendon, A., Kostadima, M., Loos, R., Ent, M. A. v. d., Kaan, A., Sharifi, N., Paul, D. S., Ifrim, D. C., Quintin, J., Love, M. I., Pisano, D. G., Burden, F., Foad, N., Farrow, S., Zerbino, D. R., Dunham, I., Kuijpers, T., Lehrach, H., Lengauer, T., Bertone, P., Netea, M. G., Vingron, M., Beck, S., Flicek, P., Gut, I., Ouwehand, W. H., Bock, C., Soranzo, N., Guigo, R., Valencia, A., and Stunnenberg, H. G. (2017). Comparative analysis of neutrophil and monocyte epigenomes. *bioRxiv*, page 237784.

Roadmap Epigenomics Consortium, Kundaje, A., Meuleman, W., Ernst, J., Bilenky, M., Yen, A., Heravi-Moussavi, A., Kheradpour, P., Zhang, Z., Wang, J., Ziller, M. J., Amin, V., Whitaker, J. W., Schultz, M. D., Ward, L. D., Sarkar, A., Quon, G., Sandstrom, R. S., Eaton, M. L., Wu, Y.-C., Pfenning, A. R., Wang, X., Claussnitzer, M., Yaping Liu, Coarfa, C., Alan Harris, R., Shoresh, N., Epstein, C. B., Gjoneska, E., Leung, D., Xie, W., David Hawkins, R., Lister, R., Hong, C., Gascard, P., Mungall, A. J., Moore, R., Chuah, E., Tam, A., Canfield, T. K., Scott Hansen, R., Kaul, R., Sabo, P. J., Bansal, M. S., Carles, A., Dixon, J. R., Farh, K.-H., Feizi, S., Karlic, R., Kim, A.-R., Kulkarni, A., Li, D., Lowdon, R., Elliott, G., Mercer, T. R., Neph, S. J., Onuchic, V., Polak, P., Rajagopal, N., Ray, P., Sallari, R. C., Siebenthall, K. T., Sinnott-Armstrong, N. A., Stevens, M., Thurman, R. E., Wu, J., Zhang, B., Zhou, X., Beaudet, A. E., Boyer, L. A., Jager, P. L. D., Farnham, P. J., Fisher, S. J., Haussler, D., Jones, S. J. M., Li, W., Marra, M. A., McManus, M. T., Sunyaev, S., Thomson, J. A., Tlsty, T. D., Tsai, L.-H., Wang, W., Waterland, R. A., Zhang, M. Q., Chadwick, L. H., Bernstein, B. E., Costello, J. F., Ecker, J. R., Hirst, M., Meissner, A., Milosavljevic, A., Ren, B., Stamatoyannopoulos, J. A., Wang, T., and Kellis, M. (2015). Integrative analysis of 111 reference human epigenomes. *Nature*, 518(7539):317–330.

Robertson, G., Hirst, M., Bainbridge, M., Bilenky, M., Zhao, Y., Zeng, T., Euskirchen, G., Bernier, B., Varhol, R., Delaney, A., Thiessen, N., Griffith, O. L., He, A., Marra, M., Snyder, M., and

Jones, S. (2007). Genome-wide profiles of STAT1 DNA association using chromatin immunoprecipitation and massively parallel sequencing. *Nature Methods*, 4(8):651–657.

Roulet, E., Busso, S., Camargo, A. A., Simpson, A. J. G., Mermod, N., and Bucher, P. (2002). High-throughput SELEX–SAGE method for quantitative modeling of transcription-factor binding sites. *Nature Biotechnology*, 20(8):831–835.

Schneider, T. D. and Stephens, R. M. (1990). Sequence logos: a new way to display consensus sequences. *Nucleic Acids Research*, 18(20):6097–6100.

Schneider, T. D., Stormo, G. D., Gold, L., and Ehrenfeucht, A. (1986). Information content of binding sites on nucleotide sequences. *Journal of Molecular Biology*, 188(3):415–431.

Schones, D. E., Cui, K., Cuddapah, S., Roh, T.-Y., Barski, A., Wang, Z., Wei, G., and Zhao, K. (2008). Dynamic Regulation of Nucleosome Positioning in the Human Genome. *Cell*, 132(5):887–898.

Schones, D. E., Smith, A. D., and Zhang, M. Q. (2007). Statistical significance of cis-regulatory modules. *BMC Bioinformatics*, 8(1):19.

Schütz, F. and Delorenzi, M. (2008). MAMOT: hidden Markov modeling tool. *Bioinformatics*, 24(11):1399–1400.

Siepel, A., Bejerano, G., Pedersen, J. S., Hinrichs, A. S., Hou, M., Rosenbloom, K., Clawson, H., Spieth, J., Hillier, L. W., Richards, S., Weinstock, G. M., Wilson, R. K., Gibbs, R. A., Kent, W. J., Miller, W., and Haussler, D. (2005). Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. *Genome Research*, 15(8):1034–1050.

Soufi, A., Garcia, M. F., Jaroszewicz, A., Osman, N., Pellegrini, M., and Zaret, K. S. (2015). Pioneer Transcription Factors Target Partial DNA Motifs on Nucleosomes to Initiate Reprogramming. *Cell*, 161(3):555–568.

Stedman, W., Kang, H., Lin, S., Kissil, J. L., Bartolomei, M. S., and Lieberman, P. M. (2008). Cohesins localize with CTCF at the KSHV latency control region and at cellular c-myc and H19 Igf2 insulators. *The EMBO Journal*, 27(4):654–666.

Stormo, G. D. and Fields, D. S. (1998). Specificity, free energy and information content in protein–DNA interactions. *Trends in Biochemical Sciences*, 23(3):109–113.

Stormo, G. D. and Hartzell, G. W. (1989). Identifying protein-binding sites from unaligned DNA fragments. *Proceedings of the National Academy of Sciences of the United States of America*, 86(4):1183–1187.

Stormo, G. D. and Zhao, Y. (2010). Determining the specificity of protein–DNA interactions. *Nature Reviews Genetics*, 11(11):751–760.

Takahashi, K. and Yamanaka, S. (2006). Induction of Pluripotent Stem Cells from Mouse Embryonic and Adult Fibroblast Cultures by Defined Factors. *Cell*, 126(4):663–676.

Thurman, R. E., Rynes, E., Humbert, R., Vierstra, J., Maurano, M. T., Haugen, E., Sheffield, N. C., Stergachis, A. B., Wang, H., Vernot, B., Garg, K., John, S., Sandstrom, R., Bates, D., Boatman, L., Canfield, T. K., Diegel, M., Dunn, D., Ebersol, A. K., Frum, T., Giste, E., Johnson, A. K., Johnson, E. M., Kutyavin, T., Lajoie, B., Lee, B.-K., Lee, K., London, D., Lotakis, D., Neph, S., Neri, F., Nguyen, E. D., Qu, H., Reynolds, A. P., Roach, V., Safi, A., Sanchez, M. E., Sanyal, A., Shafer, A., Simon, J. M., Song, L., Vong, S., Weaver, M., Yan, Y., Zhang, Z., Zhang, Z., Lenhard, B., Tewari, M., Dorschner, M. O., Hansen, R. S., Navas, P. A., Stamatoyannopoulos, G., Iyer, V. R., Lieb, J. D., Sunyaev, S. R., Akey, J. M., Sabo, P. J., Kaul, R., Furey, T. S., Dekker, J., Crawford, G. E., and Stamatoyannopoulos, J. A. (2012). The accessible chromatin landscape of the human genome. *Nature*, 489(7414):75–82.

Trifonov, E. N. (2011). Cracking the chromatin code: Precise rule of nucleosome positioning. *Physics of Life Reviews*, 8(1):39–50.

Tsompana, M. and Buck, M. J. (2014). Chromatin accessibility: a window into the genome. *Epigenetics & Chromatin*, 7.

Tuerk, C. and Gold, L. (1990). Systematic evolution of ligands by exponential enrichment: RNA ligands to bacteriophage T4 DNA polymerase. *Science*, 249(4968):505–510.

Turatsinze, J.-V., Thomas-Chollier, M., Defrance, M., and Helden, J. v. (2008). Using RSAT to scan genome sequences for transcription factor binding sites and cis -regulatory modules. *Nature Protocols*, 3(10):1578–1588.

Vierstra, J. and Stamatoyannopoulos, J. A. (2016). Genomic footprinting. *Nature Methods*, 13(3):213–221.

von Bakel, H. (2011). Methods for Analysis of Transcription Factor DNA-Binding Specificity In Vitro, Chapter 11, Interactions of Transcription Factors with Chromatin. In Hughes, T. R., editor, *A Handbook of Transcription Factors*, number 52 in Subcellular Biochemistry, pages 223–259. Springer Netherlands.

Voong, L. N., Xi, L., Wang, J.-P., and Wang, X. (2017). Genome-wide Mapping of the Nucleosome Landscape by Micrococcal Nuclease and Chemical Mapping. *Trends in Genetics*, 33(8):495–507.

Voss, T. C. and Hager, G. L. (2014). Dynamic regulation of transcriptional states by chromatin and transcription factors. *Nature Reviews Genetics*, 15(2):69–81.

Wang, J., Zhuang, J., Iyer, S., Lin, X., Whitfield, T. W., Greven, M. C., Pierce, B. G., Dong, X., Kundaje, A., Cheng, Y., Rando, O. J., Birney, E., Myers, R. M., Noble, W. S., Snyder, M., and Weng, Z. (2012). Sequence features and chromatin structure around the genomic regions bound by 119 human transcription factors. *Genome Research*, 22(9):1798–1812.

Weirauch, M., Yang, A., Albu, M., Cote, A. G., Montenegro-Montero, A., Drewe, P., Najafabadi, H., Lambert, S., Mann, I., Cook, K., Zheng, H., Goity, A., van Bakel, H., Lozano, J.-C., Galli,

M., Lewsey, M. G., Huang, E., Mukherjee, T., Chen, X., Reece-Hoyes, J., Govindarajan, S., Shaulsky, G., Walhout, A. M., Bouget, F.-Y., Ratsch, G., Larrondo, L., Ecker, J., and Hughes, T. (2014). Determination and Inference of Eukaryotic Transcription Factor Sequence Specificity. *Cell*, 158(6):1431–1443.

Weirauch, M. T., Cote, A., Norel, R., Annala, M., Zhao, Y., Riley, T. R., Saez-Rodriguez, J., Cokelaer, T., Vedenko, A., Talukder, S., Dream5 Consortium, Bussemaker, H. J., Morris, Q. D., Bulyk, M. L., Stolovitzky, G., and Hughes, T. R. (2013). Evaluation of methods for modeling transcription factor sequence specificity. *Nature Biotechnology*, 31(2):126–134.

Weirauch, M. T. and Hughes, T. R. (2011). Methods for Analysis of Transcription Factor DNA-Binding Specificity In Vitro, Chapter 3, A Catalogue of Eukaryotic Transcription Factor Types, Their Evolutionary Origin, and Species Distribution. In Hughes, T. R., editor, *A Handbook of Transcription Factors*, number 52 in Subcellular Biochemistry, pages 25–73. Springer Netherlands.

West, J. A., Cook, A., Alver, B. H., Stadtfeld, M., Deaton, A. M., Hochedlinger, K., Park, P. J., Tolstorukov, M. Y., and Kingston, R. E. (2014). Nucleosomal occupancy changes locally over key regulatory regions during cell differentiation and reprogramming. *Nature Communications*, 5(1):1–12.

Wiechens, N., Singh, V., Gkikopoulos, T., Schofield, P., Rocha, S., and Owen-Hughes, T. (2016). The Chromatin Remodelling Enzymes SNF2H and SNF2L Position Nucleosomes adjacent to CTCF and Other Transcription Factors. *PLOS Genetics*, 12(3):e1005940.

Wilson, B. G. and Roberts, C. W. M. (2011). SWI/SNF nucleosome remodellers and cancer. *Nature Reviews Cancer*, 11(7):481–492.

Wingender, E., Schoeps, T., and Dönitz, J. (2013). TFClass: an expandable hierarchical classification of human transcription factors. *Nucleic Acids Research*, 41(D1):D165–D170.

Wu, C., Jin, X., Tsueng, G., Afrasiabi, C., and Su, A. I. (2016). BioGPS: building your own mash-up of gene annotations and expression profiles. *Nucleic Acids Research*, 44(D1):D313–D316.

Zaret, K. S. and Carroll, J. S. (2011). Pioneer transcription factors: establishing competence for gene expression. *Genes & Development*, 25(21):2227–2241.

Zhang, Y., Vastenhouw, N. L., Feng, J., Fu, K., Wang, C., Ge, Y., Pauli, A., Hummelen, P. v., Schier, A. F., and Liu, X. S. (2014). Canonical nucleosome organization at promoters forms during genome activation. *Genome Research*, 24(2):260–266.

Zhao, F., Xuan, Z., Liu, L., and Zhang, M. Q. (2005). TRED: a Transcriptional Regulatory Element Database and a platform for in silico gene regulation studies. *Nucleic Acids Research*, 33(suppl_1):D103–D107.

Zhao, Y., Granas, D., and Stormo, G. D. (2009). Inferring Binding Energies from Selected Binding Sites. *PLOS Comput Biol*, 5(12):e1000590.

Zhao, Y. and Stormo, G. D. (2011). Quantitative analysis demonstrates most transcription factors require only simple models of specificity. *Nature Biotechnology*, 29(6):480–483.

Zhou, V. W., Goren, A., and Bernstein, B. E. (2011). Charting histone modifications and the functional organization of mammalian genomes. *Nature Reviews Genetics*, 12(1):7–18.

Curriculum vitae

Romain Groux

Bioinformatics software developer /
bioinformatician
32 years old, Swiss
Rue de Lausanne 49G, 1020 Renens
Phone +41(0)76 304 00 20
Email : romain.groux@hotmail.com
Linkedin : https://www.linkedin.com/in/romain-groux-b15aa4b4/
GitHub : https://github.com/romaingroux/

Strenghts

PhD in bioinformatics
Bioinformatics software developer
NGS data specialist
Naturally curious and always eager to learn

Professional experience:

04.2015 – 09.2019 EPFL, scientific assistant in the laboratory of “Computational Cancer Genomics”.

- Development of a software predicting transcription factor binding sites (C). This software predicts which regions of a genome are bound given a specificity model. This software is currently the most efficient available for this type of problem.
- Development of unsupervised classification methods (C++) for genomic region of interest, based on i) the sequence of the regions and ii) diverse NGS sequencing profiles (ChIP-seq, DNase-seq, ATAC-seq, MNase-seq). These methods allow to identify i) functionally different families of regions and ii) their signatures.
- Development, maintenance and optimization of diverse software components and analysis pipelines hosted on the laboratory web server backend.
- Curator for a public database containing NGS data (ChIP-seq, DNase-seq, ATAC-seq, MNase-seq, etc).

Technical skills:

Developing and maintaining of software and analysis pipelines in C++11 (STL, Boost, SFML, SeqAn) Python 2.7/3.6 (STL), R 2.X/3.X (diverse libraries) and bash.
Testing code using dedicated libraries in C++ (Unittest++) and Python (unittest)
Measuring and optimizing the performances of code and analysis pipelines.
Handling and processing of NGS data using dedicated softwares/libraries (samtools, bedtools, pysam), mapping (bowtie, bowtie2) and data quality control, etc.
Genomic data analysis using machine learning methods (unsupervised classification algorithm) and function prediction (transcription factor binding prediction).
Algorithm understanding and ability to implement and optimize them.

Academic projects:

PhD project : setting up and pursuing a research project studying i) transcription factor binding specificity in the human genome (development of methods allowing to model their sequence preferences, training of sequence preference models and performance assessments of the models) and ii) the chromatin structure in the vicinity of transcription factor binding sites in the human genome by the developing dedicated analysis methods. These methods were then applied on NGS datasets (ChIP-seq, DNase-seq, ATAC-seq, etc) in order to characterize some regions of the human genome.
Master project : studying the regulation of gene expression as part of a swiss consortium (CycliX). I worked on i) the development of a pipeline that performs data processing/formatting and quality control (bash, Python et R), ii) the development of different software modules required for the data analysis and iii) the application of the above for the data analysis (Python et R).

Education:

2016 – 2020	**PhD** in bioinformatics, EPFL
2013 – 2014	**Master** in bioinformatics, UNIL
2010 – 2013	**Bachelor** in biology, UNIL
2006 – 2009	**CFC** of biology laboratory technician, EPFL

Languages:

French	mother tongue
English	fluent written and spoken (equivalent to C1)
German	basic scholar knowledge

References:

Dr. Philipp Bucher	Head of the “Computational Cancer Genomics” laboratory of EPFL +41 21 693 09 56 philipp.bucher@epfl.ch
Dr. René Dreos	Senior biostatistician at the Integrative Center for Genomics of UNIL +41 21 692 40 44 rene.dreos@unil.ch
Dr. Giovanna Ambrosini	Senior scientist in the “Computational Cancer Genomics” laboratory of EPFL +41 21 693 09 57 giovanna.ambrosini.epfl.ch
Dr. Vincent Gardeux	Senior scientist in the “Systems Biology and Genetics” laboratory of EPFL vincent.gardeux@epfl.ch