



La construction du réseau de régulation transcriptionnelle

Islam Sultan

► To cite this version:

Islam Sultan. La construction du réseau de régulation transcriptionnelle. Statistiques [math.ST]. Université Paris Saclay (COmUE), 2019. Français. NNT : 2019SACLS184 . tel-02373468

HAL Id: tel-02373468

<https://theses.hal.science/tel-02373468>

Submitted on 21 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

La construction du réseau de régulation transcriptionnelle

Thèse de doctorat de l'Université Paris-Saclay
préparée à l'Université Paris-Sud

Ecole doctorale n°577 Structure et dynamique des systèmes vivants (SDSV)
Spécialité de doctorat : Sciences de la vie et de la santé

Thèse présentée et soutenue à Jouy-en-Josas, le 21 juin 2019, par

IBRAHIM SULTAN

Composition du Jury :

Denis Thieffry Professeur des Universités, École Normale Supérieure, IBENS	Rapporteur
Gregory Nuel Directeur de Recherche, CNRS/Sorbonne Université, INSMI	Rapporteur
Juliette Martin Chargée de Recherche, CNRS/Université Lyon, MMSB	Examinatrice
Stéphane ROBIN Directeur de Recherche, INRA/AgroParisTech, EcoStat	Examineur (président)
Sophie Schbath Directrice de Recherche, INRA Jouy-en-Josas, MaIAGE	Directrice de thèse
Pierre Nicolas Directeur de Recherche, INRA Jouy-en-Josas, MaIAGE	Co-directeur de thèse

Contents

List of Figures	5
Acknowledgements	7
Abstract	8
1 Biological background	11
1.1 From transcription to transcriptional regulatory networks	11
1.2 Transcriptomics	16
1.3 The Bacterium <i>Listeria monocytogenes</i>	18
1.3.1 <i>L. monocytogenes</i>	18
1.3.2 The European research network (List_MAPS)	19
1.4 Objective of the PhD project	23
2 Methodological background on DNA regulatory motif discovery	24
2.1 Overview on the existing tools	25
2.2 Probabilistic models to represent DNA motifs	27
2.3 Motif discovery in sequence data	30
2.3.1 The MEME algorithm	30
2.3.2 Gibbs motif sampler	32
2.3.3 Repulsive Parallel MCMC (RPMCMC)	33
2.4 Motif discovery incorporating auxiliary data	35
2.4.1 Making use of ChIP data	35
2.4.2 Making use of expression data	36
2.5 Motivations for the novelties in our approach	42
2.5.1 Modeling overlaps between motif occurrences	42
2.5.2 Incorporating expression data	43

3	A new statistical model for promoter sequences and the associated MCMC algorithm	45
3.1	Improved models for bacterial promoter sequences	46
3.1.1	Ingredients of the sequence model	46
3.1.2	Statistical concepts of parameter and dimension estimation	52
3.1.3	Prior settings and Directed Acyclic Graph of the hierarchical Bayesian model for sequence data	54
3.1.4	Strategy and overview of the MCMC algorithm for the sequence model	58
3.1.5	Details of the steps of MCMC algorithm for the sequence model . .	61
3.2	Incorporating expression data in the sequence model	77
3.2.1	An extended probit model to use expression data as covariates . . .	77
3.2.2	Inference	84
3.2.3	Details of the MCMC steps relative to the extended probit model .	87
4	Data collection and method implementation	94
4.1	Data collection	94
4.1.1	Promoter sequences	95
4.1.2	Expression data set	95
4.2	Building covariates summarizing expression data	96
4.2.1	Hierarchical clustering trees	96
4.2.2	Principle Component Analysis (PCA)	101
4.2.3	Independent Component Analysis (ICA)	103
4.3	Prior distribution and parameter tuning	107
4.4	The developed tool	113
4.4.1	Description of the command line arguments	113
4.4.2	Output files	116
5	Results	120
5.1	Processing of the output files to detect stable motifs	120
5.2	Results on <i>L. monocytogenes</i>	125
5.2.1	Main characteristics of the sequence motifs and the corresponding predicted regulons	125
5.2.2	Links to known transcription factors	125
5.3	Comparisons of different models	130
5.4	Comparisons with other tools	137

5.4.1	Comparison with MEME	137
5.4.2	Comparison with Rpmcmc	138
5.4.3	Comparison with FIRE	140
5.4.4	Comparison with REDUCE-Suite-v2.2	141
5.4.5	Comparison with RED2	141
6	Discussion	144
6.1	Looking backward: comparison with TreeMM and choices made for the validation of the approach	144
6.2	Extensions of the model and improvements of the algorithm	146
6.3	Additional analyses	148
	Résumé en français	149
6.4	Analyses supplémentaires	158
	References	160
	Supplementary materials	172

List of Figures

1.1	The elements of a bacterial transcription unit.	13
1.2	Schematic representation of transcription regulation by sigma factors and transcription factors	14
1.3	Cartoon representation of the 3D structure of dimerized <i>Bacillus subtilis</i> transcription factor YvoA in complex with palindromic operator DNA . . .	15
1.4	Schematic view of tiling array transcriptomics and RNA-Seq workflows . .	17
1.5	Contamination cycle of <i>L. monocytogenes</i>	20
1.6	Schematic representation of the European research project (List_MAPS) .	22
2.1	Timeline of computational tools for detecting transcription factor DNA-binding sites	29
2.2	The different models that are used in the literature to model the TFBSs .	29
2.3	A graphical representation of how an algorithm may work to prevent the repetition of detecting the same TFBS using a repulsion force function . .	34
2.4	A promoter correlation tree (Nicolas et al., 2012)	41
2.5	Motifs overlaps and distance from TSS	43
3.1	DAG of our model	57
3.2	Graphical illustration of our probit model of probability of motif occurrence for one expression covariate	80
3.3	Graphical illustration of our probit model of probability of motif occurrence for two expression covariate	81
3.4	A graphical representation of how the process of selecting a cut on the PCA or the ICA can as well be applied on the clustering tree	83
4.1	Heat map of the missing values inside the expression data matrix	97
4.2	Different clustering trees based on different distance functions	100

4.3	The different ways to summarize expression data before inputting it to the tool	102
4.4	Mean square error between the original expression matrix and the approximated matrix using ICA and PCA	105
4.5	Relationship between component stability and selected number of components in ICA	106
4.6	Set up of the simulation study to explore the impact of the prior on the number of regions	109
4.7	Impact of the prior on the estimation of pdf for motif occurrence posteriors	110
4.8	Posterior probability on K if the prior on motif positions is set according to a piece-wise function	111
4.9	Impact of the choice of the concentration parameter of the Dirichlet prior distribution for PWMs	112
5.1	Covergance plots	122
5.2	Hierarchical clustering of motif components to extract stable motifs	124
5.3	Illustration of three motifs found with our algorithm for <i>de-novo</i> motif discovery.	129
5.4	Hierarchical clustering of motifs resulted from different models	132
5.5	The output results of MEME	139
5.6	The motifs discovered by applying FIRE	140
5.7	The results of MatrixREDUCE	141
5.8	The results of RED2	143

Acknowledgements

This work was made possible thanks to the efforts and patience of my supervisor, Pierre Nicolas. He was always there, from the first day, with his advices and suggestions for me to integrate in the research lab. His present (a book titled "The Conquest Of Happiness") in the beginning of the first year has changed my way of thinking very early in the journey of the Ph.D. project and was of a great help. When times was hard, He pushed me through them while being as patience as possible. I am forever indebted to him.

Many thanks to my co-supervisor, Sophie Schbath, for always offering help, for both scientific and administrative issues.

To my dear friend, Wahid Awad, for his guidance and support all over the way, I can not imagine myself completing the road until the end if he was not there on every step of the journey to advice me when I felt that I can not go further.

To my colleagues that I have met in INRA and who became close friends afterwards: Alma, Moaz, and Nezar.

To my first office mates at INRA, Elhussain and Marie. They were of a great help for me in my first year in INRA and in France. Elhussain was like a big brother to me, helping me with so many administrative issues. Marie delivered to me free French classes on daily basis and was pushing me on the journey of learning French.

To my friends that I have met through the journey at INRA: Bernardo, Francesca, Myriam, and many more.

To my colleagues at MaIAGE, Sam, Julie, Cyprian, Slim, Anne-Laure, Gwen, Juliette, Ba, and the rest, for so many interesting discussions.

To my colleagues in the European network (List_MAPS) for sharing the journey with me from the beginning until the end, and for their biology lessons to me.

To my French teacher, Stephanie, for being an awesome teacher and for introducing me and Wahid to each other.

To my family, They have been always the base to any success in my life.

Abstract

Transcription factors play a key role in mediating the adaptation of bacteria to environmental conditions. Powerful algorithms and approaches have been developed for the discovery of their binding sites but automatic *de novo* identification of the main regulons of a bacterium from genome and transcriptome data remains a challenge. The approach that we propose here to address this task is based on a probabilistic model of the DNA sequence that can make use of precise information on the position of the transcription start sites and of condition-dependent transcription profiles. Two main novelties of our model are to allow overlaps between motif occurrences and to incorporate covariates summarizing transcription profiles into the probability of occurrence in a given promoter region. Each covariate may correspond to the coordinate of the gene on an axis (e.g. obtained by PCA or ICA) or to its position in a tree (e.g. obtained by hierarchical clustering). All the parameters are estimated in a Bayesian framework using a dedicated trans-dimensional MCMC algorithm. This allows simultaneously adjusting, for many motifs and with many transcription covariates, the width of the corresponding position weight matrices, the number of parameters to describe positions with respect to the transcription start site, and the covariates that are relevant.

The thesis manuscript is divided into six chapters. The first and second chapters are dedicated to the biological and methodological backgrounds, respectively. In the third chapter, we present the methodological core of the new approach developed during this thesis (probabilistic model, Bayesian inference). The fourth chapter is dedicated to data collection and preparation (sequence and expression data), which encompasses the dimensionality reduction techniques that served to summarize the position of the promoters in the expression space. The fifth chapter is dedicated to the presentation of the results obtained on the bacterium *Listeria monocytogenes* which was the focus of the European project List_MAPS in which this work took place. In this chapter, the results are also compared to those obtained with other motif discovery methods. The final chapter dis-

cusses briefly the future directions that could be envisioned to continue the work realized in this PhD project.

Chapter 1

Biological background

Bacteria, like all living cells, have their inherited genetic information encoded in very long deoxyribonucleic acid (DNA) double stranded molecules with a double helical structure whose sequences of four deoxyribonucleotides A, C, G, T constitute their genomes (typical size of a few millions of nucleotides). To multiply and survive through the different environmental conditions, bacterial cells need to express this genetic information by producing ribonucleic acid (RNA) molecules in the right time and right amount. Transcription is the process by which a segment (typical size of hundreds to thousands of nucleotides) of the DNA sequence called a gene serves as template for the synthesis of an RNA molecule. The most prominent role of RNA molecules is to serve as template for the synthesis of proteins consisting of sequences of amino-acids by a process named translation. These RNA molecules that are translated are called messenger RNAs. The objective of this thesis is to contribute to the identification of the elements (typically words of 5-25 nucleotides) in the DNA sequence that are responsible for the modulation of the local transcriptional activity.

1.1 From transcription to transcriptional regulatory networks

Transcription is the first and essential step of gene expression carried out by the RNA polymerase. In bacteria, a transcription unit (TU) is composed of a regulatory region (containing the promoter), a transcription start site, one or more protein coding sequences (CDSs) that are translated to proteins, and a transcription termination site in that order (5' to 3') (Fig 1.1). Since it is common for genes to be transcribed by several promoters,

TUs tend to overlap.

The regulatory region contains *cis* elements such as the promoter, where the RNA polymerase initially binds, and transcription factor binding sites (TFBSs), where transcription factors (TFs) bind to modulate the recruitment of the RNA polymerase (Browning and Busby, 2004). To recognize promoter regions, the RNA polymerase form a complex with a protein subunit known as the sigma factor which binds to sequences with specific properties. Once transcription is initiated, the sigma factor can dissociate from the complex and the RNAP continues elongation on its own (Maeda et al., 2000; Heimann, 2002; Paget and Helmann, 2003; Kazmierczak et al., 2005). Most bacteria can express several sigma factors. The so-called housekeeping sigma factor is responsible for the expression of the majority of the genes in normal growth conditions. Alternative sigma factors are activated in specific conditions upon environmental or physiological triggers and can redirect the transcription to other sets of promoters which provides a first level of regulation of the transcriptional activity. In bacteria, sigma factors are divided into two main phylogenetic families. Most sigma factors, including the housekeeping sigma factor, belong to the family Sigma-70 that recognize bipartite sequence motif composed of two elements directly upstream the transcription start site : the -10 and -35 boxes named after their respective positions with respect to the TSS. The other family, Sigma-54, is usually represented in a bacterium by a single member and recognizes a sequence motif located between positions -12 and -24.

TFs are proteins that can be classified into two groups, activators and repressors, where an activator TF is the protein that aims to increase the transcription rate while on the other hand a repressor TF is a protein that aims to reduce the transcription rate. Commonly, repressors bind to the promoter, interfering directly with the binding of RNA polymerase; while on the other hand, an activator typically binds to the promoter's upstream region, helping to recruit the polymerase and start transcription (Collado-Vides et al., 1991; Babu and Teichmann, 2003). It is worth mentioning that there are TFs with a dual regulatory role, these TFs act at the same time as an activator for some genes and as a repressor for some other genes. A TF which is bound at a given site in the intergenic region between two divergently transcribed units can regulate each one of them in a different manner (Balleza et al., 2008). TFs modulate transcriptional activity in a condition-dependent manner. They are activated in response to specific signals by protein modification such as phosphorylation/dephosphorylation and are themselves often also regulated at the transcriptional level. TFs work together in harmony and it is not uncommon that a regulatory region could be occupied by several TFs. Affinity of a TF for a particular TFBS defines how

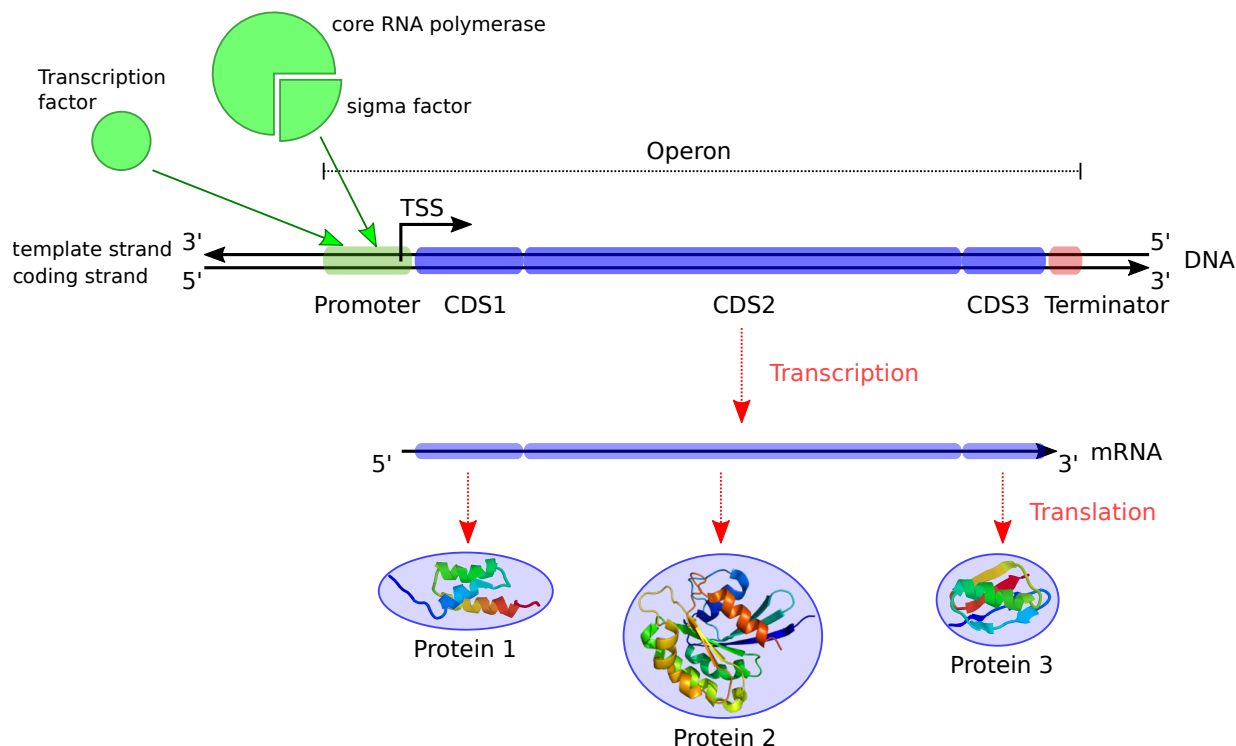


Figure 1.1: The elements of a bacterial transcription unit.

a promoter function, weak sites require high concentrations of TFs whereas strong sites respond to lower amounts (Alon, 2006, 2007).

Many TFs form dimers and therefore bind to DNA sequence elements that harbor a palindromic structure where the nucleotide composition properties that define the recognized motif are mirrored with respect to a center of symmetry (Fig 1.3). At the level of molecular three-dimensional structures, palindromic DNA recognition elements allow the formation of symmetric protein/DNA complexes (Higgins et al., 1988).

Transcriptional regulatory networks (TRN) are networks that capture the direct influence of TFs over the transcription activity of different target genes (TG) (McAdams and Arkin, 1998; Thieffry and Thomas, 1997; Lee et al., 2002). The network representation helps us to see the global organization of transcriptional regulation which has been described as modular (distinct sets of target genes regulated by distinct TFs) and hierarchical (the network typically consists of several layers with genes encoding TFs being target of other TFs) (Balleza et al., 2008). One main element in TRNs are the regulons defined as the sets of genes that are coregulated by each TF (Gutierrez-Rios et al., 2003). It is important to remember that fixation of TFs in the promoter region is not the only

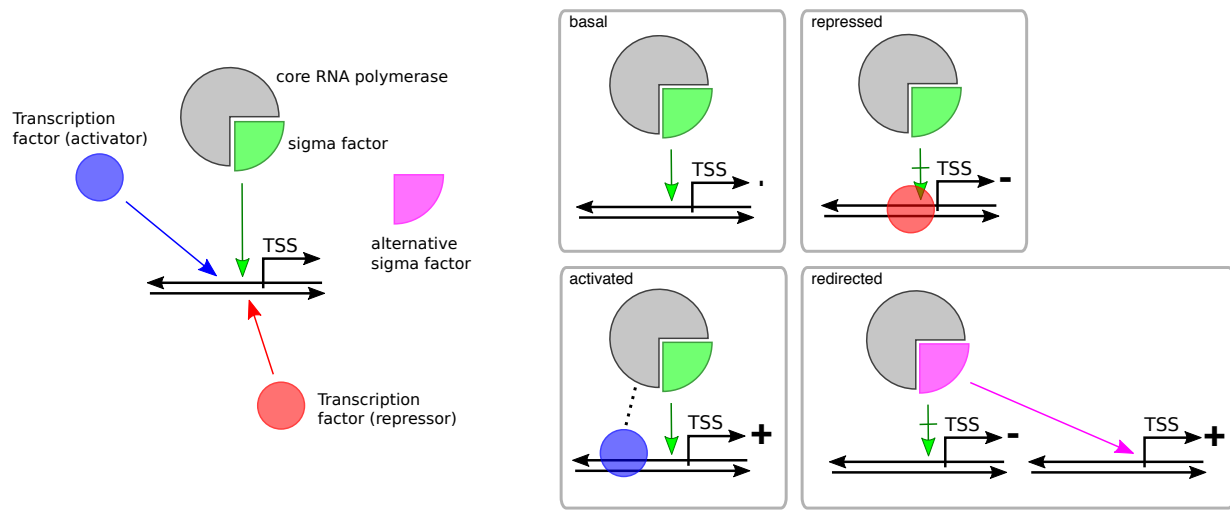


Figure 1.2: Schematic representation of transcription regulation by sigma factors and transcription factors

level of regulation, and not all the regulation is at the level of transcriptional activity. In particular, RNA degradation and regulation of translation also play important roles.

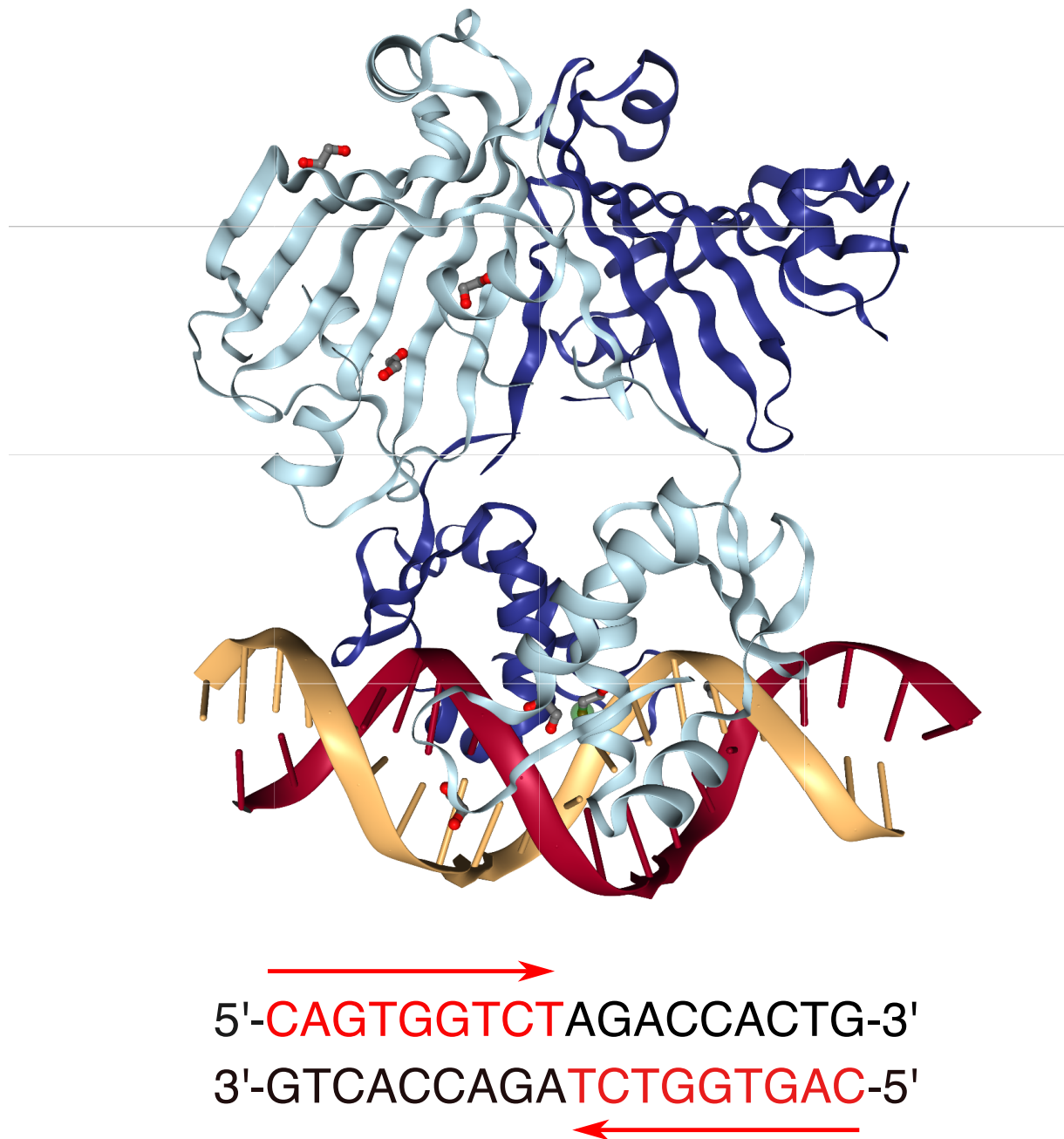


Figure 1.3: Cartoon representation of the 3D structure of dimerized *Bacillus subtilis* transcription factor YvoA in complex with palindromic operator DNA. Crystal structure obtained by X-ray diffraction, taken from the Protein data bank (Pdb accession number 4WWC). Each of the four chains is represented by a different color (light and dark blue for proteins, red and yellow for DNA). The sequence of the double stranded DNA molecule is also represented and its palindromic structure highlighted by red arrows.

1.2 Transcriptomics

Transcriptome is the complete set of RNA transcripts that are produced by the genome, under a specific condition. Transcriptomics refers to experimental approaches that aim to obtain a global picture of the transcriptome. Two main categories of approaches have been used: microarrays and RNA-Seq (Fig 1.4). They have been widely applied to compare growth conditions ("*in vitro*" stress, stage of growth, infection condition) and/or specific mutants. Both types of data can be informative for transcriptional network reconstruction.

In the microarrays technology, different single-stranded DNA probes are designed and arrayed to monitor the mRNA expression of different genes. Then transcriptional products, isolated from a culture sample, are converted to cDNA tagged with fluorescent proteins and hybridized on the microarray against their complementary sequences. The intensity of the fluorescence, in the different locations of the array, gives an estimate of the abundance of the different probed transcripts. Microarray technology has been refined since its first appearance when they detected exclusively annotated ORFs (Schena et al., 1995). The later versions of the microarray technology is represented by one-color high-density whole-genome tiling arrays. In this implementation, the arrayed set of probes is richer, containing, for example, DNA probes for both intragenic and intergenic regions and the technology distinguishes transcription from the two DNA strands (Reppas et al., 2006). The raw data generated from microarrays must be treated in two levels: correction for background noise and normalization. The correction attempts to eliminate the contribution from unspecific hybridization; while the normalization intends to make gene intensities from different experiments comparable (Quackenbush, 2002). The widespread use of this technology has led to the appearance of useful databases with collections of hundreds of arrays of different bacterial organisms under diverse experimental conditions (Demeter et al., 2006; Faith et al., 2007; Kanehisa et al., 2007).

The second experimental approach to transcriptomics is based on sequencing of the RNA molecules and has progressively replaced the microarrays during the last ten years with the development of high-throughput sequencing. This technique, coined RNA-Seq, consists of sequencing the transcripts that the cell expresses under a specific condition and then to map the sequence reads back to their corresponding regions on the genome to detect and quantify abundances of the transcripts (Nagalakshmi et al., 2008). Here, the detection of transcripts is not conditioned on a possibly biased set of probes nor on the resolution of the array, giving the possibility to discover new transcriptionally active

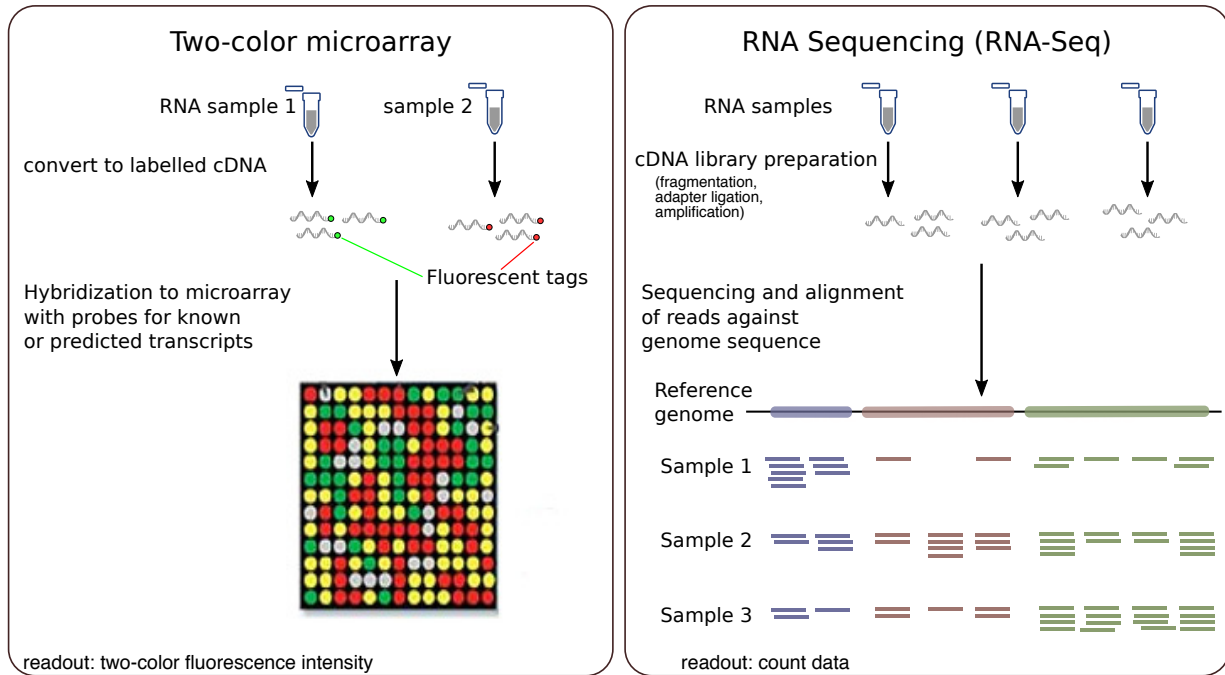


Figure 1.4: Schematic view of tiling array transcriptomics and RNA-Seq workflows. The left side of the figure shows the microarray technology, the conversion from RNA to cDNA is done via: reverse transcription, optional PCR amplification, and labeling to obtain labelled cDNA. The right side shows the RNA-Seq technology, the cDNA library preparation is done via: reverse transcription, fragmentation, adapter ligation, and PCR amplification to obtain sequencing library.

regions. The RNA-Seq technology achieves higher dynamic ranges than microarrays in transcript abundance measurements since unspecific hybridization is not present in the sequencing and read counts does not saturate like hybridization signal for highly expressed genes. The lower limit of quantification depends on the number of collected sequence reads which is decided by the user (Fig 1.4).

Experimental approaches related to the global RNA-Seq described in the previous paragraph have also been developed for genome-wide mapping of TSSs at single base resolution. These data are particularly relevant in the context of this thesis since they provide a repertoire of promoter regions that can be aligned with respect to the TSS and thus will allow to focus the search for sequence motifs recognized by TFs. The general idea consists of ligating an adapter on the 5'-end of the RNA molecule during the library preparation and sequencing from this adapter. Its implementation comes in different refinements (Wurtzel et al., 2010; Irnov et al., 2010; Ettwiller et al., 2016).

1.3 The Bacterium *Listeria monocytogenes*

L. monocytogenes bacterium was the focus of the European project whose this PhD is a part of and applying our methodology on *L. monocytogenes* was one main objective of the PhD. This section intends to give some elements of information on *L. monocytogenes* as well as on the European network. In the first subsection, we speak about *L. monocytogenes* and its different life styles. In the second subsection we introduce the European network, its objective and how the work realized in this PhD fits in the objective of the overall project.

1.3.1 *L. monocytogenes*

L. monocytogenes is a lethal food-borne pathogen. It is the agent of listeriosis, a severe infection of humans that can result in meningoencephalitis, septicaemia and spontaneous abortion. In the EU, listeriosis is a notifiable disease. A report published in 2013 by the European Food Safety Authority (EFSA)¹ on zoonoses, zoonotic agents and food-borne outbreaks 2011. A total of 1476 cases of listeriosis were reported in 2011. Of all the zoonotic diseases under EU surveillance, listeriosis caused the most severe human disease with 93.6 % of the cases hospitalised and 134 fatal cases (case fatality rate 12.7 %). The 2017–18 South African listeriosis outbreak which resulted from contaminated processed meats was the world's worst-ever listeriosis outbreak. It caused around 200 deaths out of 973 confirmed infections².

Following human consumption of *L. monocytogenes*-infected food, the pathogen must adapt and survive to the acidic environment of the stomach and the hostile environment of the upper gastrointestinal tract. It is emerging that the alternative sigma factor SigB orchestrates molecular adaptation in this environment by activating transcription of genes involved in tolerance of acid, salt and bile. SigB also triggers expression of the internalin proteins that promote internalisation into host enterocytes (Ferreira et al., 2003; Abram et al., 2008). It is most likely that the SigB regulon is then down-regulated and the pathogen induces primary virulence genes that are regulated by the "master and commander" transcriptional activator PrfA (de las Heras et al., 2011).

Beside being a pathogen, *L. monocytogenes* is a ubiquitous bacterium present in agri-

¹<https://efsa.onlinelibrary.wiley.com/doi/epdf/10.2903/j.efsa.2013.3129>

²<https://www.timeslive.co.za/news/south-africa/2018-05-18-death-toll-from-listeria-outbreak-rises-to-more-than-200/>

cultural and food environments (soil, animals, food industry). The variety of environments where *L. monocytogenes* can be detected is striking (Fig 1.5). It has been isolated from soil, plants and vegetables, decaying vegetation, groundwater, biowastes, composts (Welshimer, 1960) and in some cases from environmental hosts such as protozoans and lower vertebrates.

As most bacteria, *L. monocytogenes* has the ability to attach and form biofilms on abiotic surfaces (Borucki et al., 2003). This is a major concern for the food industry where *L. monocytogenes* can survive and persist on surfaces under harsh conditions. *L. monocytogenes* is able to survive harsh conditions and to withstand environmental stresses, including food technology-related stresses. its ability to respond to a variety of suboptimal growth conditions makes *L. monocytogenes* a very versatile pathogen found in many habitats. It is well adapted to persistence in soil environments, in food-processing environments and in chilled, processed foods. In addition, it is likely that stress-adapted cells of *L. monocytogenes* are better equipped to survive during infection of the host gastrointestinal tract (Gahan and Hill, 2014). Very little molecular information is currently available concerning the ability of *L. monocytogenes* to adapt to environmental stress within the food matrix.

Owing to its status as important human pathogen and model bacterium for the study of host-pathogen interaction and bacterial transcriptomics (Sorek and Cossart, 2010), a wealth of transcriptomics data has been collected on *L. monocytogenes* and will be used in the context of this study. In particular, one could cite the use of RNA-Seq to detect new transcribed regions and high-density tiling arrays to compare various *in vitro* and *in vivo* conditions by Toledo-Arana et al. (2009), the use of genome-wide TSS mapping approach by Wurtzel et al. (2012) providing a repertoire of 2299 TSSs, and the work done by Bécavin et al. (2017) to aggregate the diversity of transcriptomics data in the Listeriomics database (3159 genes and their expression profiles over 254 comparisons of experimental condition pairs).

1.3.2 The European research network (List_MAPS)

List_MAPS is a European project funded by the Research and Innovation programme of the European Union Horizon2020 under the Marie-Skłodowska Curie actions (ITN-ETN). Coordinated by Université de Bourgogne, the project started on March 2015 and associates nine full partners and two associated partners from five European countries (France, Ireland, Germany, Netherlands, and Denmark) which belong to private and public sectors.

List_MAPS focuses on *L. monocytogenes* which is costing the EU millions of euro

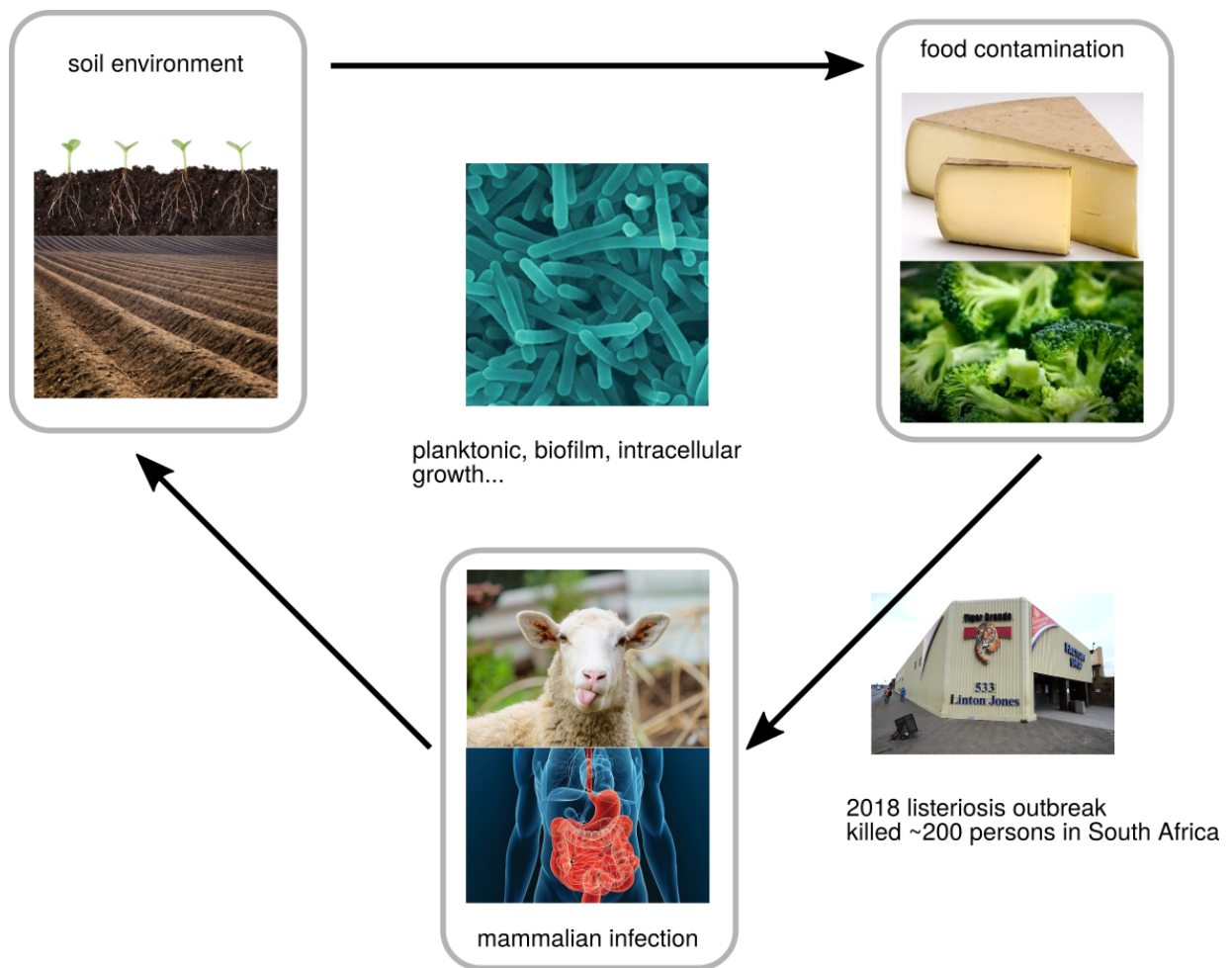


Figure 1.5: Contamination cycle of *L. monocytogenes*.

per annum in medical care and associated costs in the food sector. The research programme is based on a combination of high throughput Epigenetics, RNA-Seq, Proteomics, Bioinformatics, Mathematics and Microbiology (Fig 1.6). With the combination of these expertises List_MAPS focuses on several ambitious objectives:

- To understand how environmental conditions in soil, plants, biofilms and food matrices influence the capacity of *L. monocytogenes* to cause infection, and to study the possible impact of "stress hardening" upon subsequent virulence potential.
- To develop an integrated model of the regulatory circuitry of the pathogenic bacterium *L. monocytogenes* in order to refine our knowledge of the environmentally-dependent gene modules that underpin its ubiquitous nature and its capacity to generate infection.
- To assess intraspecific diversity of virulence potential and biofilm in relation to environmental cues.
- To develop a cost efficient, rapid semiconductor sequencing application designed to assess the virulence potential of large numbers of isolates, sparing the cost, burden and ethical issues related to animal models.

To achieve these objectives eleven Early-Stage Researchers (ESR) have been recruited for 24 to 36 months to develop scientific expertise through an innovative Ph.D training, including mobility of researchers, participation to national and international events like summer schools, workshops and conferences and transfer-of-knowledge in the areas of Transcriptomics, Proteomics, Sequencing and Systems Biology.

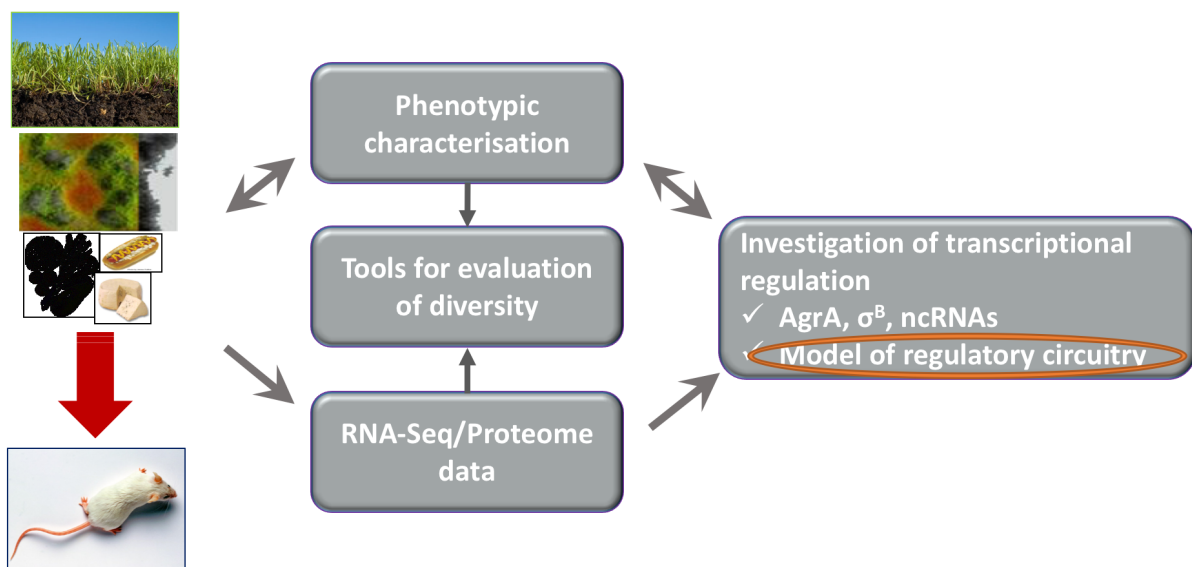


Figure 1.6: This schema represents the overall research project (List_MAPS). The contribution done this PhD project is circled with red line (reconstruction of the regulatory circuit of *L. monocytogenes*) .

1.4 Objective of the PhD project

The objective of this PhD is developing a new method that can integrate several types of information (sequence data, TSS maps, and expression profiles over many conditions) for *de novo* identification of the main regulons of a bacterium and to apply this methodology to *L. monocytogenes*.

The available data was of a great help to us in applying our tool, specifically the genome-wide TSS mapping approach by Wurtzel et al. (2012), and the work done by Bécavin et al. (2017) to aggregate the diversity of transcriptomics results in the Listeriomics database.

Chapter 2

Methodological background on DNA regulatory motif discovery

Over the last four decades, since Korn et al. (1977) has published their computer algorithm and until now, the motif discovery problem has been a major concern in the field of bioinformatics. Given that it is a problem whose solution requires a multidisciplinary approach, advances in the fields of applied statistics, computer science and computational biology keep improving the proposed solutions.

I start this chapter by a global overview on the existing tools, making "a survey on the available surveys" in the first section. I have dedicated the second and third sections to discuss some modeling options and the different algorithms for parameter estimation. In the second section, I focus on the probabilistic models to represent DNA motifs. The methodology on how to estimate the model parameters is the focus of the third section, in which I give some details on two classical approaches that differ by the type of algorithm, deterministic algorithm (Expectation-Maximization) and randomized algorithm (Gibbs sampling).

In the fourth section, I will focus on how different tools use different types of auxiliary data to improve the search for sequence motifs. I will review the use the genome-wide ChIP data and how their introduction has impacted this field of research. After that I will detail several approaches that were proposed to make use of expression data for motif discovery. I will finish the chapter by introducing the novelty about our approach and how they overcome some of the limitations of the existing tools.

2.1 Overview on the existing tools

There have been more than hundred(s) publications detailing motif discovery methods and, by turn, there was a need for surveys of the existing methods. Since, there is no simple global framework to classify all the existing approaches and to tell us how much they are related to each other from methodological point of view, different publications have chosen different approaches to classify the available algorithms. Examples for the approaches used to classify the tool include the algorithm used (ex. deterministic or probabilistic), the type of auxiliary data used (ex. expression data or ChIP data), the type of motif model (ex. single or composite), and the motif model representation (probabilistic or word-based) (Das and Dai, 2007; Sandve and Drabløs, 2006; Zambelli et al., 2012). Some of these angles are detailed below.

The type of motif under search can be a direction to classify the tools. As done by Sandve and Drabløs (2006) when they divided most of the existing tools by then (119 tools) into two sets, the first set contains tools that search for single motif models and the second set contains tools that search for composite motif models. Where, the single motif model refers to short contiguous sequence elements and the composite motif model refers to clusters of elements in the DNA in proximity to each other, but with a certain flexibility regarding distance between them.

Relating the development of new predictive tools to the available experimental data was the focus of Slattery et al. (2014) who have presented the existing computational methods along a timeline and related the prediction quality of these computational methods with the availability of experimental data (Fig 2.1 on page 29).

The type of auxiliary data which is used by the search algorithm can be the criteria of classification. For example, Das and Dai (2007) have divided many of the existing algorithms by then (51 tools) into two sets, the first set contained algorithms that use promoter sequences of genes that have similar expression activity from single genome and search for statistically over-represented motifs, as co-regulated genes are more likely to contain some common motifs, as discussed in subsection 1.2. The second set of algorithms contains the ones designed to use phylogenetic footprinting or orthologous sequences. The simple premise underlying phylogenetic footprinting is that selective pressure causes functional elements to evolve at a slower rate than non-functional sequences. This means that usually well conserved sites among a set of orthologous promoter regions are good candidates for functional regulatory elements such as TFBSs. Several motif finding algorithms have been

developed based on phylogenetic footprinting (Cliften et al., 2001; Blanchette and Tompa, 2002; Cliften et al., 2003; Wang and Stormo, 2005; Carmack et al., 2007).

Chromatin immunoprecipitation (ChIP) made it possible to assess experimentally which sequences are bound by a given transcription factor. This technology had a deep impact on the field of motif discovery and some have chosen to divide the tools into two groups, one for the tools that make use of the ChIP data and the other is for those which do not (Zambelli et al., 2012) (see section 2.4.1).

With the existence of many methods that share the same goal, benchmarks were expected to assess the relative performance of these methods. Weirauch et al. (2013) is a good example for these benchmarks where, as a part of the DREAM5 challenge¹, they applied 26 different approaches to *in vitro* protein binding microarray data for 66 mouse TFs belonging to various families. Their results indicated that simple models based on mononucleotide position weight matrices trained by the best methods perform similarly to more complex models for $\sim 90\%$ of the TFs examined.

¹<http://dreamchallenges.org/project/dream-5-tf-dna-motif-recognition-challenge/>

2.2 Probabilistic models to represent DNA motifs

Most of the word-based methods for motif discovery start by computing the score of every possible k -mer (ex. $k = 7$) and then proceed with a local optimization of the highest scoring k -mers (seeds). Scoring usually implies a hypothesis testing framework aiming for finding the unexpected words with respect to a null hypothesis H_0 . An example of H_0 is to assume that words are generated from a Markov model of order $k' < k - 1$ which allows to account for the frequency of words of length $k' + 1$ (Schbath, 2000). Another example of H_0 is to assume that the frequency of a word is the same between two set of sequences which can correspond to promoter regions of genes in different expression clusters (see subsection 2.4.2). After local optimization of the seeds, these methods usually represent motifs using consensus strings with degenerate symbols (IUPAC system). The IUPAC notation uses the following ambiguity codes: W = {A,T}, S = {C,G}, M = {A,C}, K = {G,T}, R = {A,G}, Y = {C,T}, B = not A, D = not C, H = not G, V = not T, N = any Nucleotide (not a gap).

Probabilistic models aim to offer more precise description of the motifs than aforementioned word-based methods. The most common way to model or represent motifs in a probabilistic way is the position weight matrix (PWM). A motif of length \mathcal{W} is represented by a $4 * \mathcal{W}$ matrix, where the 4 rows correspond to the 4 possibilities of the DNA nucleotides (A,C,G,T) and the \mathcal{W} columns represent the different positions within the motif. We will refer to a nucleotide as a residue and use the letter r to represent it. The value θ_{rw} at the r^{th} row and the w^{th} column in the PWM reflects the probability of seeing residue r in position w . Often the elements in PWMs are provided as log-likelihood ratio terms (comparing motif and background) which permits to use them directly as additive scores. In this manuscript, the elements will simply be probabilities of occurrence and, for every column w in the PWM, we have thus ($\sum_r \theta_{wr} = 1$). In this context, the motif discovery task consists of estimating the parameters that represent the motif. One example for a PWM is the following one that corresponds to the -35 box of *L. monocytogenes* alternative sigma factor SigB as detected by our tool:

$$\theta^T = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} \\ \begin{matrix} A \\ C \\ G \\ T \end{matrix} & \begin{pmatrix} 0.111 & 0.003 & 0.090 & 0.001 & 0.001 & 0.324 & 0.791 \\ 0.223 & 0.116 & 0.001 & 0.027 & 0.006 & 0.074 & 0.063 \\ 0.431 & 0.878 & 0.001 & 0.001 & 0.001 & 0.203 & 0.068 \\ 0.235 & 0.003 & 0.908 & 0.971 & 0.992 & 0.399 & 0.078 \end{pmatrix} \end{matrix}$$

Since the basic PWM represents a motif where every position is independent from the other positions, we can look at the PWM as an inhomogeneous fixed order Markov model of order 0, where inhomogeneous here means position-dependent. The approach of PWM to model the motif can be generalized to account for the dependencies between adjacent residues, represented by Markov model (MM), or even the dependencies between any positions inside the motif, represented by Bayesian Networks (BN), see Fig 2.2 on page 29.

To generalize the concept of the PWM, some have proposed higher order Markov models to represent the motif. When selecting the order of the Markov model, a specific trade-off appears. Intuitively, the higher the order the better, since more parameters are used to describe the model which can provide additional predictive power. On the other hand, the higher the order, the less reliable the parameters estimates are, since less training data are available to estimate the value of each parameter. The motif can as well be represented by a variable order MM, in that case the order may depend not only on the position inside the motif but also on the exact nucleotides found at the preceding positions (Bühlmann et al., 1999; Ron et al., 1996).

The concept of modeling the dependencies between adjacent nucleotides can be generalized even further by considering the dependencies between any pair of nucleotides within a motif. This generalization can be modeled using Bayesian Networks (BN) (Ben-Gal et al., 2005). The Dinucleotide Weight Matrices (DWMs) (Siddharthan, 2010) was presented to model the motif in this manner, it is actually a first order BN. BNs, as well, can be divided into fixed order or variable order (Boutilier et al., 1996; Friedman and Goldszmidt, 1998; Friedman et al., 2000; Heckerman et al., 1995).

Beside a probabilistic representation of the motif, a full probabilistic method also involves a background model representing the nucleotides outside the motif blocks referred to hereafter as θ_0 and a model for the position of the motif in the sequence. The first background models assumed independent and identically distributed (iid) nucleotides as in MEME (see subsection 2.3.1), and later homogeneous Markov models were shown to provide better results (Thijs et al., 2001). Different modeling of the motif position will be briefly mentioned in subsection 2.3.1 (OOPS, ZOOPS, TCM).

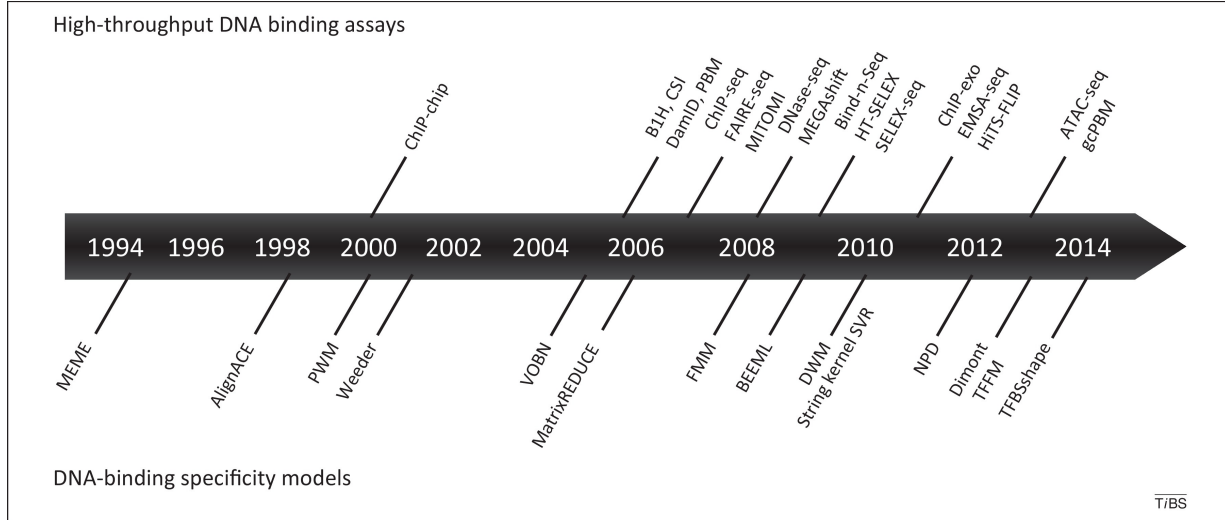


Figure 2.1: Timeline relating the development of computational tools for detecting TFBSs below the timeline axis and the availability of the experimental data above the timeline axis (Slattery et al., 2014).

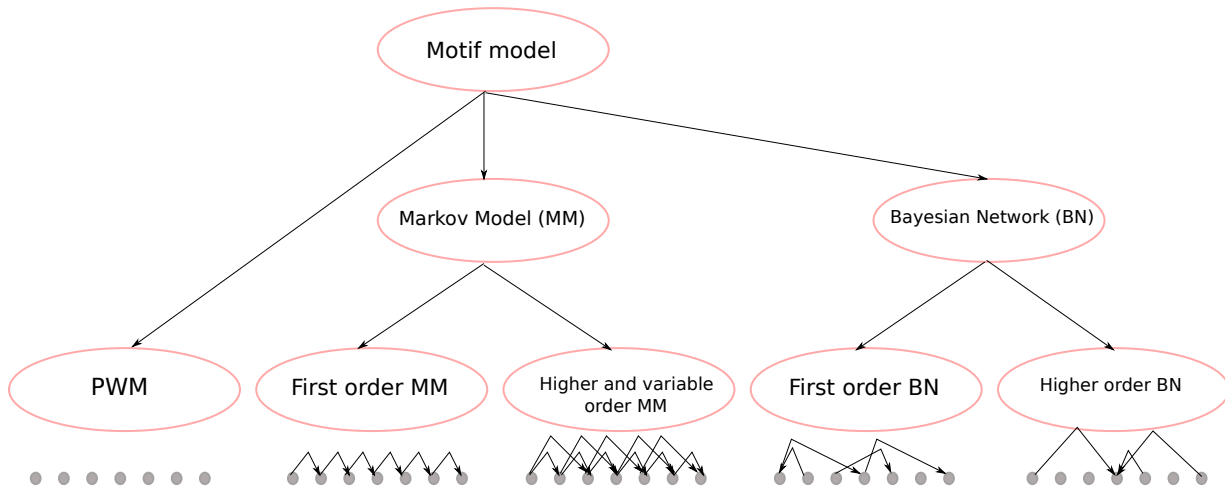


Figure 2.2: The different models that are used in the literature to model the TFBSs. MM can be seen as a generalization to PWM and also the BN can be seen as a generalization to MM.

2.3 Motif discovery in sequence data

Here I am presenting two classical tools that can be regarded as representatives to the set of tools that are based on EM and gibbs algorithms. These two tools are MEME which is based on EM and Gibbs Motif Sampler (GMS) which is based on Gibbs sampling. I am ending the Section by presenting a tool that is based on GMS with extensions, this tool is called Repulsive Parellel MCMC and it can be regarded as representative for the set of tool that are variations or extensions of GMS and MEME.

Both tools (MEME, GEMS) have made the choice of modeling the starting position of the motif in the context of hidden variables, but this was done in two slightly different ways by the two tools: MEME introduces a binary variable for every position in the sequence equal to 1 if that position is the starting position of the motif and zero otherwise. GEMS has one variable per sequence where its value (not binary in this case) represents the starting position of the motif.

2.3.1 The MEME algorithm

MEME (standing for "Multiple EM for Motif Elicitation") is very popular and one of the first unsupervised learning algorithms for discovering motifs in sets of protein or DNA sequences. The two first versions of the algorithm were described in Bailey et al. (1994); Bailey and Elkan (1995a). Here we base our description of MEME on the third publication (Bailey and Elkan, 1995b) that presents more modeling options and provides more details on the implementation.

Overview of MEME

In a nutshell, MEME algorithm is a combination of: Expectation Maximization (EM) (Dempster et al., 1977) for parameter estimation via local maximization of the incomplete log-likelihood; an EM-based heuristic for choosing the starting point for EM; multi-start for searching over possible motif widths; greedy search for finding multiple motifs; and a maximum likelihood ratio-based (LRT-based) heuristic for choosing the number of model's free parameters (depending on the width of the motif and on whether or not the DNA palindrome constraints are in force).

OOPS, ZOOPS, and TCM models

MEME offers several options for the modeling of the number of occurrences per sequence in the dataset. OOPS is the simplest model type, since it assumes that there is exactly one occurrence of the motif per sequence. This type of model was introduced by Lawrence and

Reilly (1990). The generalization of OOPS is called ZOOPS, which assumes zero or one motif occurrence per dataset sequence. Finally, TCM (two-component mixture) models assume that there can be any number (including zero) of non-overlapping occurrences of the motif in each sequence in the dataset, as described by Bailey et al. (1994). All these models assume a uniform distribution for the position of the motif inside each sequence.

Expectation Maximization

Consider searching for a single motif of width \mathcal{W} in a set of sequences. The dataset of \mathcal{N} sequences, each of length \mathcal{L} , will be referred to as $x = (x_{n,l})$, $n = 1 : \mathcal{N}$, $l = 1 : \mathcal{L}$. MEME does not require that all the sequences have the same length but this assumption simplifies the presentation of the method. There are $\mathcal{T} = \mathcal{L} - \mathcal{W} + 1$ possible starting positions for a motif occurrence in each sequence. The starting point(s) of the occurrence(s) of the motif, if any, in each of the sequences are unknown and are represented by hidden variables $Z = Z_{n,l} | 1 \leq n \leq \mathcal{N}, 1 \leq l \leq \mathcal{T}$ where $Z_{n,l} = 1$ if a motif occurrence starts in position l in sequence x_n , and $Z_{n,l} = 0$ otherwise. MEME uses the EM algorithm to obtain a maximum likelihood estimate of the model parameters including θ . This requires maximizing the so-called incomplete likelihood since the value of Z are unknown and need to be integrated out. This is done by iterating the following two steps until a convergence criterion is met.

The E-step. The E-step of EM computes for all n and l , $\pi(Z_{n,l} = 1 | x_n, \theta^{(i)})$, the probability that a motif occurrence starts in position l of sequence x_n given the current parameters $\theta^{(i)}$. For an OOPS model,

$$\pi(Z_{n,l} = 1 | x_n, \theta^{(i)}) = \frac{\pi(x_n | Z_{n,l} = 1, \theta^{(i)})}{\sum_{l=1}^{\mathcal{T}} \pi(x_n | Z_{n,l} = 1, \theta^{(i)})}, \quad (2.1)$$

where,

$$\pi(x_n | Z_{n,l} = 1, \theta) = \left(\prod_{w=1}^{\mathcal{W}} \theta_{w, x_{n, l+w-1}} \right) \mathbb{I}\{Z_{n,l} = 1\} \left(\prod_{k \in \Delta_{n,l}} \theta_{0, x_{n,k}} \right) \mathbb{I}\{Z_{n,k} = 0\}, \quad (2.2)$$

here $\Delta_{n,l}$ is the set of positions in sequence x_n which lie outside the occurrence of the motif when the motif starts at position l .

The M-step. The M-step of EM consists of updating $\theta^{(i)}$ using the following formula

$$\theta_{w,r}^{(i+1)} = \frac{c_{w,r} + d_{w,r}}{\sum_{r' \in \{\mathbf{A}, \mathbf{C}, \mathbf{G}, \mathbf{T}\}} c_{w,r'} + d_{w,r'}}, \quad 0 \leq w \leq \mathcal{W}, \quad (2.3)$$

where,

$$c_{w,r} = \begin{cases} \sum_{n=1}^{\mathcal{N}} \sum_{l=1}^{\mathcal{T}} \pi(Z_{n,l} = 1 | x_n, \theta^{(i)}) \mathbb{I}\{x_{n,l+w-1} = r\} & \text{if } w > 0 \\ \sum_{n=1}^{\mathcal{N}} \sum_{l=1}^{\mathcal{T}} \mathbb{I}\{x_{n,l+w-1} = r\} - \sum_{w=1}^{\mathcal{W}} c_{w,r} & \text{if } w = 0, \end{cases} \quad (2.4)$$

and $d_{w,r}$ is a fixed value used to stabilize the estimate. As explained in section 2.3.1 the E-step consists of computing the expected value of the complete likelihood and the M-step of maximizing this quantity.

Finding multiple motifs

The three types of sequence model used by MEME assume, *a priori*, that motif occurrences are equally likely at each position l in sequence x_n . That is, initially, MEME assumes that $\pi(Z_{n,l} = 1)$ is a fixed value for all $Z_{n,l}$ in the first pass (the process of discovering one motif). On the second and subsequent passes, MEME changes this assumption to approximate a multiple-motif sequence model. A new probability distribution on each $Z_{n,l}$ is used during the E-step, by multiplying formula (2.1) by the probability that a new motif occurrence starting at position l might overlap occurrences of the motifs found on previous passes of MEME.

2.3.2 Gibbs motif sampler

Gibbs motif sampler (Neuwald et al., 1995) is a popular algorithm that uses Gibbs sampling (Gelfand and Smith, 1990) for sequence motif discovery. Given a set of DNA sequences, assume that within the sequences there are \mathcal{M} blocks (motifs) and each motif (m) has a different width \mathcal{W}_m . Let also $A_{\cdot,n}$ be random variables that record the starting positions of the \mathcal{M} motifs in sequence x_n . The goal of the Gibbs motif sampler is to sample from the posterior distribution of all non-overlapping motifs given the sequence data of density $\pi(a|x) = \int_{\theta} \pi(a, \theta|x) d\theta$ (for conciseness we write here a for $A = a$). Given the position of the various elements within sequence x_n recorded in $a_{\cdot,n}$, the probability of observing sequence x_n is

$$\pi(x_n | A_{\cdot,n}, \theta) = \left(\prod_{m=1}^{\mathcal{M}} \prod_{w=1}^{\mathcal{W}_m} \theta_{m,w,x_{n,A_{m,n}+w-1}} \right) \left(\prod_{k \in \Delta_{n,A_{\cdot,n}}} \theta_{0,x_{n,k}} \right), \quad (2.5)$$

where $\theta_{m,w,r}$ represents nucleotide r in position w in motif m , and $\Delta_{n,A_{\cdot,n}}$ is the set of positions in sequence x_n which lie outside the occurrence of the \mathcal{M} motifs recorded in $A_{\cdot,n}$. The Dirichlet distribution was chosen as the prior for θ . As a result of that choice,

the conditional posteriors are also Dirichlet which facilitates the analysis. Specifically, if the position of motif m is known in all but in sequence x_n , the posterior distribution for the parameters of the corresponding residue frequency model θ_m is the product of \mathcal{W}_m independent 4-parameter Dirichlet distributions,

$$(\theta_{m,w}|x, A) \sim Dir(c_{m,w,\mathbf{A}} + \alpha_{w,\mathbf{A}}, c_{m,w,\mathbf{C}} + \alpha_{w,\mathbf{C}}, c_{m,w,\mathbf{G}} + \alpha_{w,\mathbf{G}}, c_{m,w,\mathbf{T}} + \alpha_{w,\mathbf{T}}), \quad (2.6)$$

where $c_{m,w,r}$ represents the count of residues of type r at position w of motif m in all the sequences, and $\alpha_{w,r}$ is a fixed value used to stabilize the estimate. The Gibbs algorithm permits to sample the joint distribution $\pi(\theta, A|x)$ which is the distribution of interest by iteratively sampling from the complete set of conditionals $\pi(\theta|x, A)$ and $\pi(A|x, \theta)$. To avoid sampling from the Dirichlet distribution, Lawrence et al. (1994) showed how to integrate out θ to sample directly $\pi(A|x) = \int_{\theta} \pi(A, \theta|x) d\theta$ using the following simple form

$$\pi(A_{m,n}|x, A_{m,[-n]}) \propto \left(\prod_{w=1}^{\mathcal{W}_m} \prod_{\mathbf{r}} c_{m,w,\mathbf{r}}[-n] + \alpha_{w,\mathbf{r}} \right) \left(\prod_{\mathbf{r}} c_{0,\mathbf{r}}[-n] + \alpha_{0,\mathbf{r}} \right), \quad (2.7)$$

where $A_{m,[-n]}$ represents the positions of motif m in all sequences except x_n , and $c_{m,w,r}[-n]$ represents the count of residues of type r at position w of motif m in all the sequences except x_n . Equation (2.7) uses the fact that

$$(\theta_{m,w}|x_{[-n]}, A_{[-n]}) \sim Dir(c_{m,w,\mathbf{A}}[-n] + \alpha_{w,\mathbf{A}}, c_{m,w,\mathbf{C}}[-n] + \alpha_{w,\mathbf{C}}, c_{m,w,\mathbf{G}}[-n] + \alpha_{w,\mathbf{G}}, c_{m,w,\mathbf{T}}[-n] + \alpha_{w,\mathbf{T}}), \quad (2.8)$$

sampling using equation (2.7) requires only to maintain up-to-date the residue counts in all motif occurrences at all steps of the algorithm. Marginalizing out θ can also improves the convergence speed of the MCMC algorithm.

2.3.3 Repulsive Parallel MCMC (RPMCMC)

The Gibbs Motif Sampler is only one example of the possible application of the MCMC method to the problem of motif discovery. Ikebata and Yoshida (2015) proposed an extension to search for different motifs at the same time (in a parallel manner). The main idea of the Repulsive Parallel MCMC (RPMCMC) is to have several one-motif discovery Markov chains (replicas in the author's terminology) that run in parallel and interact in order to avoid detecting the same motif (illustrated in fig 2.3). This is done by introducing

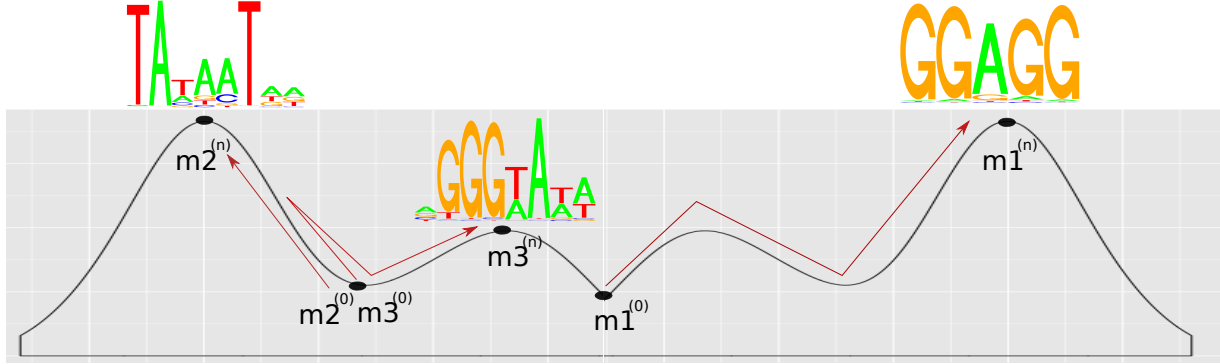


Figure 2.3: A graphical representation of how an algorithm may work to prevent the repetition of detecting the same TFBS without masking the discovered motifs using a repulsion force function. The closer two PWMs get to each other according to a specific distance function, the bigger the repulsion force between them; this will then will prevent the two matrices to converge to the same local Maximum.

a parameter β that tunes the strength of the repulsive force between replicas.

In practice the RPMCMC samples from the adhoc joint distribution defined by a density that depends on x and β , and writes

$$\pi_{\beta,x}(\theta_1, \dots, \theta_{\mathcal{M}}) \propto \psi(\theta_1, \dots, \theta_{\mathcal{M}})^{\beta} \prod_{m=1}^{\mathcal{M}} \pi(\theta_m), \quad \beta \geq 0. \quad (2.9)$$

The function ψ increases with the dissimilarity between the different replicas and the parameter β controls the force severity, i.e. a greater β produces a stronger repulsion. The function ψ is defined by

$$\psi(\theta_1, \dots, \theta_{\mathcal{M}}) = \prod_{m=1}^{\mathcal{M}} \exp(\min_{m' < m} D(\theta_m, \theta_{m'})), \quad (2.10)$$

where D is a dissimilarity between pairs of PWMs. Computing $D(\theta_m, \theta_{m'})$ requires to align θ_m and $\theta_{m'}$.

2.4 Motif discovery incorporating auxiliary data

2.4.1 Making use of ChIP data

The availability of ChIP data has taken the field of motif discovery many steps ahead (Zambelli et al., 2012). ChIP is an experimental technique that permits the genome-wide identification of protein-DNA interactions *in vivo*. ChIP, applied to transcription factors and coupled with genome tiling arrays (ChIP on Chip) or High-throughput sequencing technologies (ChIP-Seq) are the two types of data (Pillai and Chellappan, 2009; Mardis, 2007). ChIP-Seq has rapidly become the *de facto* standard in this field, since it can provide maps of the binding of the TF studied with a much higher resolution than ChIP on Chip in large genomes (Ho et al., 2011).

ChIP allows to single out a set of genomic regions whose binding sites from the same TF are experimentally supported. These regions usually range in size around a few hundred base pairs. Since the regions obtained by ChIP are larger than the actual TFBSs, outputs of ChIP experiments are perfect case study for motif finding in order to identify and model the actual binding specificity of the TF investigated. The sites bound by the TF are more likely to be located near the center of the region extracted (Jothi et al., 2008) or within a few base pairs from the point of maximum enrichment within the ‘peak’ region itself (positional bias within the input sequences being a key factor for assessing the significance of a motif). Analyzing the high number of regions identified by ChIP (typically, thousands of regions) taking into account the peak heights and the position inside the region with respect to the peak has posed new challenges to the developers of algorithms and tools (Zambelli et al., 2012).

Motif discovery tools applied to the output data from ChIP experiments tend to be more successful than the analysis of promoter sequences from co-expressed genes (Krig et al., 2007; Zeller et al., 2006; Loh et al., 2006; Chen et al., 2008). The reason for that is the challenge to delineate clusters in expression dataset and the absence of one-to-one correspondence between co-expression clusters and TF regulons (see subsection 2.4.2). In particular, it is a well-known fact that different regulatory motifs can have overlapping gene sets (Wagner, 1999; Hobert, 2008).

Despite the power of ChIP experiments, methods based on ChIP data have their limitations for *de novo* motif discovery. Indeed they require to know *a priori* the TFs that are relevant for the biological question and dedicated ChIP experiments are needed for

each of these TFs. Furthermore, ChIP analysis of a specific TF–DNA complex is usually a heavy experiment, since it requires either a specific antibody to recognize the TF or genetic engineering of a tagged TF that can be recognized by an already available antibody.

In a system biology framework, when the goal is to unravel globally with limited resources the transcription regulatory network of an organism on which we have limited knowledge, the methods that rely on expression data present some advantages over ChIP-based method: for instance, expression profiles (e.g. RNA-Seq) across biologically relevant growth and stress conditions can be obtained without focusing on specific TFs and without genetic engineering. For this reasons, with the aim of developing generic tool for *de novo* motif discovery in non-model bacteria, this PhD project focuses on making use of expression data.

2.4.2 Making use of expression data

In this subsection we are going to review tools that have been specifically developed to incorporate expression profiles as auxiliary data in motif discovery. Here I decided to review tools that are either popular or related to our work. Given a set of \mathcal{N} promoter regions (sequences) each of which is associated with one or several values that summarize the original expression data of the corresponding gene. These values, hereafter denoted by y , can be either continuous or discrete. A continuous value will typically correspond to the log ratio between expression levels between two biological conditions. A discrete value typically represents the membership of a gene to a co-expression cluster that reflects relative distances in the expression space.

The tools presented below proposed different approaches to take into account this type of data: mutual information, enrichment test, or regression of expression data y given sequence data x . The first two approaches involve transforming the expression data into one-dimensional categorical values, while the third approach can directly accommodate multidimensional continuous expression data.

Using mutual information

FIRE (standing for "Finding Informative Regulatory Elements") is an algorithm that tries to find sequence motifs with maximum mutual information between their patterns of occurrence and expression values (Elemento et al., 2007). It starts by breaking the given set of DNA sequences into all the possible k -mers (seeds) of a given length, say 7 (e.g.,

CGATCAG). Then, it calculates the mutual information between the presence or absence of every seed and the expression profiles of the sequences containing this seed. After that, it sorts all the seeds according to their information scores and those seeds above a specific threshold are then considered for more general motif representation.

The concept of mutual information is well defined, both for continuous and for discrete random variables (Cover and Thomas, 2012). Nonetheless, in practice, estimating the information when continuous variables are involved requires quantizing their values. FIRE does this by transforming continuous expression values into equally populated bins, as described by Slonim et al. (2005). Let's define a random variable Y_n that corresponds to the expression value associated with sequence x_n which can take \mathcal{Y} possible values (the number of bins or clusters), and a random variable $A_{m,n}$ that corresponds to the presence/absence (encoded 1 and 0, respectively) of motif m in sequence n . The mutual information between $A_{m,n}$ and Y_n writes

$$I(A_m; Y) = \sum_{a_m=0:1} \sum_{y=1:\mathcal{Y}} \pi(a_m, y) \log \frac{\pi(a_m, y)}{\pi(a_m)\pi(y)} \quad (2.11)$$

where $\pi(a_m, y)$ is the joint probability density of (A_m, Y) and $\pi(a_m)$ and $\pi(y)$ the corresponding marginals for $A_{m,n} = a_m$ and $Y_n = y$. The estimator of this mutual information is obtained by plugging the empirical estimates $\hat{\pi}(a_m, y)$, $\hat{\pi}(a_m) = \sum_{y=1:\mathcal{Y}} \hat{\pi}(a_m, y)$, and $\hat{\pi}(y) = \sum_{a_m=0:1} \hat{\pi}(a_m, y)$. The joint density estimate $\hat{\pi}(a_m, y)$ equals $c_{m,y}/\mathcal{N}$ for $a_m = 1$ and $(c_y - c_{m,y})/\mathcal{N}$ for $a_m = 0$, where c_y is the number of sequences with expression value y , and $c_{m,y}$ is the number of sequences with expression value y and containing at least one occurrence of motif m .

Using enrichment test

Just like FIRE, GEMS (standing for "Gene Enrichment Motif Searching algorithm") starts by considering all possible k -mers for specific range of k (e.g., $k \in 6, 7, 8$) and then performs a local search starting from the best scoring seeds to define degenerate motifs (Young et al., 2008). However, instead of using mutual information, GEMS uses the hyper-geometric distribution to assess motif enrichment in each cluster, given a set of co-expression clusters. Let $c_{m,y}$ denote the observed number of genes in cluster y containing the motif m . Under the null hypothesis that the occurrences of motif m are distributed independently of the expression data, the random variable $C_{m,y}$, representing the number of occurrences of motif m in cluster y , follows an hyper-geometric distribution such that

$$\Pr(C_{m,y} \geq c_{m,y}) = \sum_{i=c_{m,y}}^{\min(c_y, c_m)} \frac{\binom{c_m}{i} \binom{\mathcal{N}-c_m}{c_y-i}}{\binom{\mathcal{N}}{c_y}}, \quad (2.12)$$

where c_y is the number of genes in cluster y , and c_m is the number of genes with motif m in \mathcal{N} . For a given clustering with \mathcal{Y} clusters, GEMS defines the motif enrichment score of m as

$$s(m) = \max_{y \in \{1:\mathcal{Y}\}} (-\log \pi(C_{m,y} \geq c_{m,y})). \quad (2.13)$$

The criterion is local in the sense that only the cluster y in which the maximum is reached is taken into account, in contrast to FIRE which uses a global criterion in the sense that the score of the motif is based upon the whole dataset.

From clusters to neighborhoods

RED2 (standing for "Regulatory Element Discovery") algorithm (Lajoie et al., 2012) bypasses the need to work with predefined clusters through replacing the clusters by the k nearest neighbors of each gene. This criteria can be applied to both the global criterion of FIRE and the local criterion of GEMS. The formula (2.11) that defines the global criterion of FIRE is replaced by

$$I(A_m; Y) = \frac{1}{\mathcal{N}} \sum_{a_m=0:1} \sum_{n \in \mathcal{N}} \pi(a_m | y_n) \log \frac{\pi(a_m | y_n)}{\pi(a_m)}. \quad (2.14)$$

Here $\pi(a_m = 1 | y_n) = \frac{c_{m,n,k}}{k} = 1 - \pi(a_m = 0 | y_n)$, where $c_{m,n,k}$ is the number of sequences containing the motif m in the k -neighborhood of gene n .

The formula (2.13) that defines the local criterion of GEMS is replaced by

$$s(m) = \max_{n \in \mathcal{N}} (-\log \pi(C_{m,n,k} \geq c_{m,n,k})). \quad (2.15)$$

The probability of seeing $C_{m,n,k} \geq c_{m,n,k}$ sequences containing motif m in the k nearest neighbors of sequence n is derived from the hyper-geometric distribution.

Regression-based methods: modeling the expression data given sequence data ($Y|x$)

REDUCE (standing for "Regulatory Element Detection Using Correlation with Expression") fits a multivariate model to gene expression across one or several conditions indexed by h , where the explanatory variables are occurrences of sequence motifs within regulatory regions (Bussemaker et al., 2001). This is done through the equation

$$y_{n,h} = b + \sum_{m \in \mathcal{M}} e_{m,h} c_{m,n} + \epsilon_{n,h}, \quad (2.16)$$

where b represents a baseline expression level common to all genes, $e_{m,h}$ represents the effect that the presence of motif m in the upstream region has on the expression level, it can be increased or decreased in the value of the expression, $c_{m,n}$ represents the number of occurrence of motif m in the upstream region of gene n , $y_{n,h}$ corresponds to the expression level of gene n , and $\epsilon_{n,h}$ represents the error term.

Whereas, in the original MotifREDUCE (Bussemaker et al., 2001), $c_{m,n}$ simply represents the count of a particular oligonucleotide motif in the regulatory region of gene n , MatrixREDUCE (Foat et al., 2006) uses a more powerful representation of the binding specificity of the transcription factors in the form of a PWM. In this case, the algorithm optimizes a matrix θ_m and the integer counts are replaced by

$$c_{m,n} = \sum_{l=1}^{\mathcal{T}} \prod_{w=1}^{\mathcal{W}_m} \theta_{m,w,x_{n,l}}, \quad (2.17)$$

where $\theta_{m,w,x_{n,l}}$ is the probability of seeing the nucleotide $x_{n,l} \in (\text{A,C,G,T})$ in the w^{th} position of motif m , \mathcal{W}_m is the width of motif m ; and $\mathcal{T} = \mathcal{L} - \mathcal{W} + 1$, where \mathcal{L} is the length of the sequence upstream of gene n .

Modeling the sequence data given expression data ($X|y$)

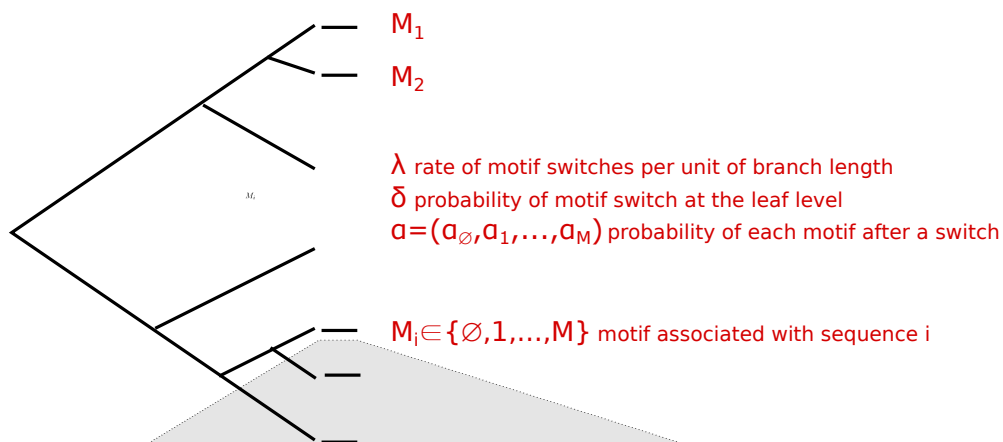
Modeling expression data given sequence data ($Y|x$) as proposed by the regression-based approaches implemented in REDUCE is not the only option that exists to take into account expression data y in *de novo* motif discovery. Beyond the possible difficulty to find appropriate models for expression data, an obvious limitation of this modeling framework is that it entirely focuses on the discovery of sequence motifs that explain the expression data. The alternative viewpoint, which is the one adopted in this PhD thesis (see subsection 2.5.2), consists in modeling the sequence data given the expression data ($X|y$). A key

benefit of this viewpoint is that it builds upon the powerful sequence modeling approaches for *de novo* motif discovery described in section 2.3 which makes it possible to envision algorithms that could simultaneously use the expression data and all the statistical properties of the sequence, establishing a continuum between discovery of motifs related and unrelated to the available expression data.

Nicolas et al. (2012) proposed a statistical model for the discovery of Sigma factor binding sites based on this idea of modeling $X|y$. The details of the model are illustrated in Fig. 2.4. As an alternative to the use for y of a predefined set of co-expression clusters, the statistical model intends to incorporate the full information of an ultra-metric tree obtained by a hierarchical clustering procedure whose topology and branch lengths reflect similarities between activity profiles. The tree was obtained by hierarchical clustering with average link based on pairwise distance between expression profiles of the different genes. Namely, the distance between the expression profiles of genes n_1 and n_2 was defined as $(1 - r_{n_1, n_2})/2$ where r_{n_1, n_2} is the Pearson correlation coefficient. This distance, sometimes referred as the Pearson distance, gives a distance 0 for a perfect positive correlation and 1 for perfect negative correlation.

In order to model the fact that sequences that are close in the expression space (hence in the tree) are more likely to harbor binding sites for the same sigma factors, the occurrences of the different possible motifs are modeled as resulting from an “evolution” process along the branches of the tree (see Fig. 2.4A). This evolution model involves a single parameter corresponding to the rate of motif switches per unit of branch length is introduced for this purpose. The model also account via a second parameter δ for the possibility of outliers. The sequence model given the presence of the motif is illustrated in Fig. 2.4B and accounts for specific properties of Sigma factor binding sites: bipartite motifs with a preferred distance with respect to the TSS. Of note, the model assumes exactly one occurrence of one of the possible motifs. This modeling assumption is probably relevant for Sigma factor binding sites since each TSS is, in principle, associated with exactly one Sigma factor binding site (see subsection 1.1) but does not hold for other types of TFs.

A Model of the distribution of motif types in the promoter correlation tree



B Model of sequence i associated with motif m

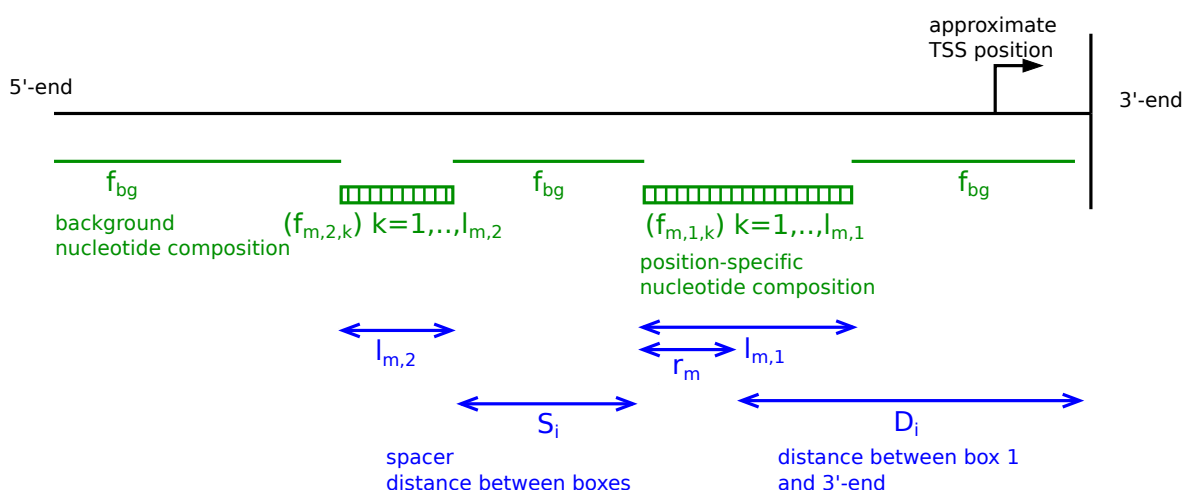


Figure 2.4: The promoter correlation tree (Nicolas et al., 2012). (A) Modeling the non-random distribution of motif types across the promoter activity correlation tree. (B) Modeling of a sequence given its associated motif type.

2.5 Motivations for the novelties in our approach

Two main novelties of our model are to allow overlaps between motif occurrences and to incorporate covariates summarizing transcription profiles into the probability of occurrence in a given promoter region. Each covariate may correspond to the coordinate of the gene on an axis (e.g. obtained by PCA or ICA) or to its position in a tree (e.g. obtained by hierarchical clustering). This subsection is divided into two parts. First, I will discuss the biological and computational motivations for modeling the possibility of overlap between motif occurrences. Second, I will motivate a new way of incorporating expression data in the *de novo* search for regulatory sequence motifs.

2.5.1 Modeling overlaps between motif occurrences

The literature showed in many occasions that the cis-regulatory regions are highly complex and they consist of repetitive as well as overlapping transcription factor binding sites. Hermesen et al. (2006) have performed statistical analysis of the frequency of overlapping and repetitive binding sites in *E.coli* based on a dataset brought from the EcoCyc database² version 9.5 (Keseler et al., 2005). This database included more than 1000 mapped transcription initiation sites, which are regulated by nearly 1400 binding sites for specific transcriptional regulatory factors (TFs). They found that 37% of the genes are mediated by more than one binding site (this situation is taken care of by almost all the methods whether they are model-based or word-based) but they found as well that 39% of the binding sites overlap with at least another site (this situation is rarely taken care of by the model-based methods). Accordingly, these findings indicate that without modeling motif overlap we will miss the discovery of at least 20% of the binding sites in a well studied bacteria like *E.coli*. We find it useful to introduce how these binding sites may work together to regulate the transcription level of some genes. The upper part of Fig 2.5 is taken from Hermesen et al. (2006) and it shows an example of promoters for the bacterium *E.coli*. Ezer et al. (2014) have described in more details how the clusters of transcription factor binding sites may work together.

There are, as well, two main computational advantages for allowing the motif occurrences recorded in hidden variables (see subsection 2.3) to overlap. First, it permits to update the motif one by one without having to change the position of other motifs. Secondly, it allows to update motifs' widths without having collisions between different motifs.

²<http://EcoCyc.org/>

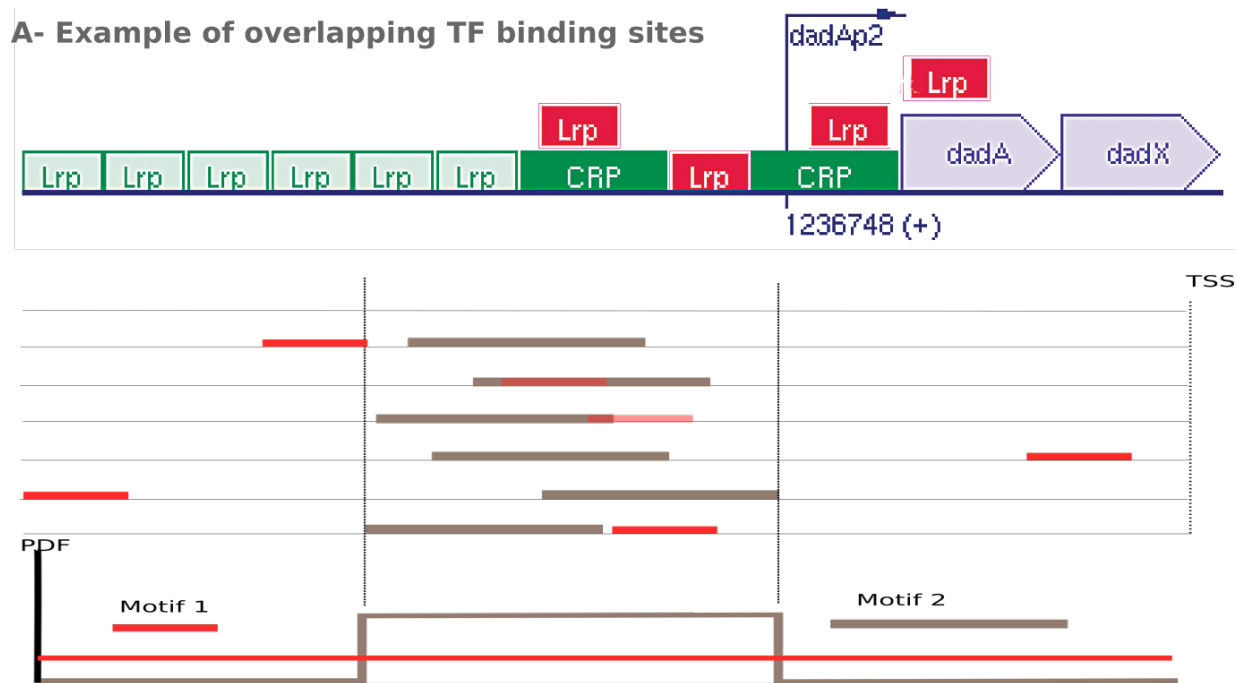


Figure 2.5: The upper part of the figure shows an examples of *E. coli* promoters that shows a complex structures which needs to be taken care of in the process of modeling. Green blocks represent binding sites for activators and red ones represents those for repressors. The lower part of the figure shows a graphical representation of modeling the distance between the motif occurrence and the TSS. The main motivation about this modeling decision is the fact that some motifs tends to bind a reserved site with a reserved distance to the TSS.

Another, but not main, novelty in our approach is that we modeled the distance between the motif occurrence and the TSS. We did so by dividing the sequence into sub-regions and introducing a dedicated set of estimated parameters that record the number of sub-regions and the exact starting and ending position of each one of them, as well as the probability of occurrence of each motif inside each sub-region (Fig 2.5).

2.5.2 Incorporating expression data

I discussed in the last part of subsection 2.4.2 that modeling the sequence data given expression data ($X|y$) is the approach adopted in this PhD thesis and I talked about the benefits of making that modeling choice. As well, I presented the work done by Nicolas et al. (2012) in which they proposed an alternative approach to use y than the conventional use of a predefined set of co-expression cluster. The work done in this PhD thesis can be regarded as an extension to what Nicolas et al. (2012) did in their approach. In this PhD,

we are incorporating covariates that summarize transcription profiles into the probability of TFBS occurrence in a given promoter region. Each covariate corresponds either to a numerical value, such as the coordinate of the gene on an axis obtained by Principal or Independent Component Analysis (PCA or ICA), or to the position of the gene in a tree, such as typically obtained by hierarchical clustering. Incorporating expression data is done using the probit regression framework and the dimensionality of the model is adjusted using trans-dimensional MCMC which allows the algorithms to activate only the covariates that are relevant to each motif.

If the covariate corresponds to a numerical value over a PCA or an ICA axis, the information can be incorporated directly into the probit regression model. Another option on which we worked is to transform the information into binary values, by introducing a cut (value) over the axis that will define two sets of genes, those on the right hand side and those on the left hand side and the two sets of genes will be given different probability of containing the motif.

When the covariate variable is representing a tree, the algorithm selects a branch that will define two sets of genes (those leaves under the branch and those which are not) and the two sets will be given different probability of containing the motif. Since different branches in a same tree could be informative on the probability of occurrence of a same motif we duplicated the two trees in the input of the algorithm (Ward tree on Euclidean distance and average-link tree on Pearson correlation distance). Selecting this way to deal with a clustering tree allows us to handle directly a whole tree and it fits in our model of binarizing the expression covariate.

Among other characteristics, the modeling framework adopted in this PhD avoids the classical way of defining clusters of genes from the transcription data and by turn avoids the drawbacks of hard clustering as explained by Lajoie et al. (2012) and discussed in subsection 2.4.2. Our approach also, beside the aforementioned advantage of avoiding the hard clustering, has the ability to handle different types of ways selected to summarize the expression data.

Chapter 3

A new statistical model for promoter sequences and the associated MCMC algorithm

As stated in the biological and methodological introductions (Chapters 1 and 2), the goal here is to propose a new approach for *de novo* discovery transcription factor binding sites in promoter sequences that can make use of expression data.

The statistical model that was developed for this purpose is composed of two parts. The first part is a model for a data-set of promoter sequences that involves interspersed occurrences of several types of sequence motifs which are allowed to overlap and to have preferred positions (sequences being aligned with respect to the transcription start sites). This sequence model builds directly upon the already existing models for *de novo* motif discovery presented in Section 2.3. As such, it allows already to identify statistically over-represented sequence motifs that can correspond to transcription factor binding sites (TFBS).

The second part of the developed model intends to relate the occurrences of the motifs to the expression data available. As explained in Section 2.4 the point of view that we decided to adopt is to try to use these data as predictors (covariates) that can improve the representation of the promoter sequences by carrying information on where similar motifs are more likely to occur (based on the idea that the genes which are co-regulated by the same TF are more likely to present similarity in their expression profiles).

The chapter is divided into two sections that are dedicated to the two aspects of the model (sequence data and expression data, respectively). These sections explain the mod-

eling and describe the MCMC algorithm that we implemented to carry out the statistical inference in a Bayesian framework. In the first section, after having introduced the ingredients of the sequence model but before entering into the details of the algorithm, we also briefly present the statistical concepts that were used for inference (Bayesian framework and MCMC algorithms).

3.1 Improved models for bacterial promoter sequences

3.1.1 Ingredients of the sequence model

We introduce here the main assumptions of our probabilistic model and the notations used for the data, hidden variables, and parameters. Some of the parameters are of direct interest to us like the Position Weight Matrices (PWMs) that have generated the motifs. The Directed Acyclic Graph represented in Figure 3.1 can be used as a summary of the notations for the different components of our model.

Probabilistic modeling of the sequences

The sequence data set that we consider is composed of a number \mathcal{N} of DNA sequences, each of the same length \mathcal{L} . Let denote $x = (x_n)_{n=1:\mathcal{N}}$ the set of DNA sequences and $x_{n,l}$ the nucleotide in position l of the sequence of index n ¹ ($x_{n,l} \in \{\text{A, C, G, T}\}$). Our probabilistic representation for these sequences involves $\mathcal{M} + 1$ unobserved components that capture the heterogeneity of the nucleotide composition along the sequences. These components consists of \mathcal{M} motif models with respective widths $(w_1, \dots, w_{\mathcal{M}})$ and one background model.

The description of the probabilistic model for the sequences x contains two aspects: the modeling of positions of motif occurrences and the modeling of the sequence given the positions of these occurrences. The modeling of the positions of motif occurrences proceeds as follows.

- The model assumes zero or one occurrence of each of the \mathcal{M} motifs per sequence. In a sequence, the occurrences of the \mathcal{M} motifs are furthermore modeled as mutually independent.

¹We will use the terms “sequence n ” to refer to the “sequence of index n ” and the same for the other components of the model (motifs, regions, ...).

- In this section, the probability that an occurrence of motif m occurs in sequence n is denoted α_m and is considered as the same for any sequence n . The focus of Section 3.2 is the extension of this model to incorporate expression data in this probability of occurrence, thereby bringing an information which is specific of each sequence.
- The probability distribution for the position of motif m in a sequence is defined by a piecewise constant function with k_m breakpoints. The positions of the breakpoints are denoted $d_m = (d_{m,k})_{k=1:k_m}$, where $1 < d_{m,k} < d_{m,k+1} \leq \mathcal{L}$. The probability of occurrence of the motif at a particular position l in region k (i.e. $d_{m,k-1} \leq l < d_{m,k}$) is $\lambda_{m,k}/(d_{m,k} - d_{m,k-1})$ where $d_{m,k} - d_{m,k-1}$ corresponds to the length of region k . Hence, for a given number of occurrences of motif m , the distribution of these occurrences between the $k_m + 1$ regions is described by a multinomial of parameters $\lambda_m = (\lambda_{m,k})_{k=1:k_m+1}$, with $\lambda_{m,k} \geq 0$ and $\sum_{k=1}^{k_m+1} \lambda_{m,k} = 1$, where $\lambda_{m,k}$ is the probability for an occurrence of motif m to be found in the region k .
- For practical reasons that pertain to the implementation of the update of the motif width w_m in the MCMC algorithm we do not always rely on the first position within motif m to index its position. Instead, we use a reference position $r_m \in \{1, \dots, w_m\}$ that will allow the motif occurrences to overlap the sequence boundaries. According to this modeling, a motif can extend up to $r_m - 1$ positions upstream of the sequence (when its indexed position occurs at position 1 in the sequence) and up to $w_m - r_m$ positions downstream of the sequence (when its indexed position occurs at position \mathcal{L} in the sequence).

The modeling of the nucleotide sequence given the positions of motif occurrences involves a set of parameter for each model component: $(\theta_1, \dots, \theta_{\mathcal{M}})$ for the \mathcal{M} motifs and θ_0 for the background.

- In keeping with the simple PWM framework for motif modeling (see Section 2.2), the nucleotide at position $w \in \{1, \dots, w_m\}$ inside an occurrence of motif m is drawn from a multinomial distribution of parameters $\theta_{m,w} = (\theta_{m,w,r})_{r \in \{\text{A,C,G,T}\}}$, with $\theta_{m,w,r} \in [0, 1]$ and $\sum_r \theta_{m,w,r} = 1$, where $\theta_{m,w,r}$ is the probability to observe nucleotide r in the position w of the motif m . As explained below the columns inside a PWM are not necessarily modeled with w_m independent vectors $\theta_{m,w}$'s since we account for the possibility of (fully or partially) palindromic motifs (see page 50).

- Nucleotides outside motif occurrences are modeled as generated by a background model. The background model that we choose is a homogeneous Markov model of order $v \in \{0, 1, \dots\}$. We denote $\theta_0 = ((\theta_{0,s,r})_{s \in \{A,C,G,T\}^k, r \in \{A,C,G,T\}})_{k=0:v}$ the parameters of this model, where $\theta_{0,s,r}$ corresponds to the probability of nucleotide r after the word s of length v . This model involves $4^v \times 3$ independent parameters (since $\sum_{r \in \{A,C,G,T\}} \theta_{0,s,r} = 1$) that allow to account for the composition in word of length $v + 1$; θ_0 also encompasses the transitions matrices of order k from 0 to $v - 1$ that serve to model the nucleotide sequences at position $l \leq v$ (i.e. the initial distribution of the Markov model).
- The assumption of independence between the occurrences of the \mathcal{M} motifs allows overlaps. Two modeling options were envisioned to model nucleotide composition at positions where two or more motif occurrences overlap.
 - The simplest model consisted of considering a simple arithmetic mean of the probability density functions associated with the different motifs that overlap. We refer to this model as the equal weight mixture model for motif overlaps.
 - A more general model was also implemented and consisted in a weighted mean of the probability density functions associated with the different motifs. We refer to this model as a θ -dependent weight mixture model for motif overlaps. According to this model, each density function described by $\theta_{m,w}$ receives a weight that increases with the level of constraint that it imposes on the nucleotide to be generated. We considered for this purpose a quantity denoted $IC(\theta_{m,w})$ expressed in information bits and often referred to as the 'information content' of a column of a PWM in the literature on biological sequence motifs. $IC(\theta_{m,w})$ takes values from 0 when the randomness is maximum (i.e. uniform distribution) to 2 in absence of randomness (i.e. $\theta_{m,w,r} = 1$ for one of the possible nucleotide $r \in \{A, C, G, T\}$). It is computed as

$$IC(\theta_{m,w}) = 2 - \sum_{r \in \{A,C,G,T\}} \theta_{m,w,r} \log_2 \theta_{m,w,r}, \quad (3.1)$$

where the sum term corresponds to the Shannon entropy of the nucleotide distribution described by $\theta_{m,w}$.

Hidden variables

In order to be able to work with this model, we introduced two layers of hidden variables which are:

- $A = (A_{m,n})_{m=1:\mathcal{M}, n=1:\mathcal{N}}$ is a layer of random variables that encode the position of the occurrences of the motifs, with $A_{m,n} \in \{0, 1, \dots, \mathcal{L}\}$ being the position of motif m in sequence n (0 encodes the absence of occurrence).
- $B = (B_{n,l})_{n=1:\mathcal{N}, l=1:\mathcal{L}}$ is a layer of random variables that encode the disambiguation of the overlaps between motifs, with $B_{n,l} \in \{0, 1, \dots, \mathcal{M}\}$ denoting the model component (background or one of the \mathcal{M} motifs) responsible for the probability distribution function of the nucleotide $x_{m,n,l}$. This idea of disambiguation by a dedicated hidden variable is made possible by our choice of modeling the nucleotide emission probability at positions where motifs overlap as a mean since the resulting density can be seen as the marginal of a mixture model (equal weight mixture model or θ -dependent weight mixture model).

Having introduced these hidden random variables (A, B) allows to define the complete data $(x, A = a, B = b)$ whose density function given the parameters $(\theta_0, \theta_{1:\mathcal{M}}, \alpha_{1:\mathcal{M}}, d_{1:\mathcal{M}}, \lambda_{1:\mathcal{M}})$ that we will simply write $(\theta, \alpha, d, \lambda)$ decomposes as

$$\begin{aligned} \pi(x, a, b | \theta, \alpha, d, \lambda) &= \prod_{n=1:\mathcal{N}} \pi(x_n, b_n, a_n | \theta, \alpha, d, \lambda) \\ &= \prod_{n=1:\mathcal{N}} \pi(x_n | b_n, a_n, \theta) \pi(b_n | a_n, \theta) \pi(a_n | \alpha, d, \lambda), \end{aligned} \quad (3.2)$$

where the terms $\pi(x_n | b_n, a_n, \theta)$, $\pi(b_n | a_n, \theta)$, $\pi(a_n | \alpha, d, \lambda)$ are easy to compute as shown below.

The density function of the sequence given the hidden variables writes

$$\begin{aligned} \pi(x_n | b_n, a_n, \theta) &= \prod_{l=1:\mathcal{L}} \pi(x_{n,l} | b_{n,l}, A_n, \theta) \\ &= \prod_{l=1:\mathcal{L}} [\theta_{b_{n,l}, l - a_{n,b_{n,l}} + 1, x_{n,l}}]^{\mathbb{I}\{b_{n,l} \geq 1\}} [\theta_{0, x_{n,l-v:l-1}, x_{n,l}}]^{\mathbb{I}\{b_{n,l}=0\}}, \end{aligned} \quad (3.3)$$

where $\mathbb{I}\{z\}$ is the indicator function that takes value 1 if the Boolean variable z is true and 0 otherwise, and $x_{n,l-v:l-1}$ denotes the word of length v finishing at position $l-1$ of sequence n , i.e. $x_{n,l-v:l-1} = (x_{n,l-v}, \dots, x_{n,l-1})$.

The density function of the disambiguation hidden variables given the position of the motif occurrences writes

$$\pi(b_n|a_n, \theta) = \prod_{l=1:\mathcal{L}} \pi(b_{n,l}|a_n, \theta), \quad (3.4)$$

where to write $\pi(b_{n,l}|a_n, \theta)$ it is easier to introduce a variable $o_{n,l}$ corresponding to the number of motif occurrences that overlap the position (n, l) ,

$$o_{n,l} \triangleq \sum_{m=1:\mathcal{M}} \mathbb{I}\{a_{n,m} \neq 0, a_{n,m} \leq l + r_m - 1 < a_{n,m} + w_m\}. \quad (3.5)$$

If one or several motif occurrences overlap the position (n, l) , we further need to distinguish the case of the equal weight mixture model for motif overlaps in which

$$\begin{aligned} \pi(B_{n,l} = m|a_n, o_{n,l}, \theta) &\propto \mathbb{I}\{m = 0, o_{n,l} = 0\} \\ &\quad + \mathbb{I}\{m > 0, a_{n,m} \neq 0, a_{n,m} \leq l + r_m - 1 < a_{n,m} + w_m\}, \end{aligned} \quad (3.6)$$

and the case of the θ -dependent weight mixture model for motif overlaps in which

$$\begin{aligned} \pi(B_{n,l} = m|a_n, o_{n,l}, \theta) &\propto \mathbb{I}\{m = 0, o_{n,l} = 0\} \\ &\quad + f(IC(\theta_{m,l+r_m-a_{n,m}})) \mathbb{I}\{m > 0, a_{n,m} \neq 0, a_{n,m} \leq l + r_m - 1 < a_{n,m} + w_m\}. \end{aligned} \quad (3.7)$$

The density function of the motif occurrences writes

$$\begin{aligned} \pi(a_{n,m}|\alpha, d, \lambda) &= \mathbb{I}\{a_{n,m} = 0\}(1 - \alpha_m) \\ &\quad + \mathbb{I}\{a_{n,m} \neq 0\} \alpha_m \prod_{k=1:k_m+1} \left[\frac{\lambda_{m,k}}{d_{m,k} - d_{m,k-1}} \right]^{\mathbb{I}\{d_{m,k-1} \leq a_{n,m} < d_{m,k}\}}. \end{aligned} \quad (3.8)$$

Dimension and palindromic constraints

The dimension of the model expressed as the number of parameters that serve for the probabilistic description of the sequence data are not fixed and will be adjusted in the

course of the estimation algorithm. In particular, the dimension of the model change with:

- the number of breakpoints used to describe the distribution of the occurrences of the motifs $(k_m)_{m=1:\mathcal{M}}$,
- the widths of the m motifs $(w_m)_{m=1:\mathcal{M}}$,
- the Markov order v of the background,
- the palindromic constraints that are applied to the PWMs.

We mentioned in Chapter 1 that many of the known binding motifs of TFs are palindromic in the sense that pairs of columns at opposed positions with respect to the center of the motif appeared as mirrored according to Watson-Crick base pairing rule ($\text{A} : \text{T}$ and $\text{C} : \text{G}$). We considered that modeling these palindromic constraints on PWMs could be useful since they reduce the dimension of the model and therefore simultaneously increase the amount of data available to estimate each parameter and decrease the size of the search space for the parameter values.

Instead of imposing a strict palindromic constraint on all or a subset of the motifs we developed a more flexible modeling approach that allows, for each motif m , smooth transitions between palindromic and non palindromic structures. This approach relied on the introduction of the following variables to define the active constraints on θ_m :

- p_m , a binary variable taking value 1 if the motif m is allowed to have a (partially) palindromic structure, 0 otherwise;
- $c_m \in \{1.5, 2, 2.5, \dots, w_m - 0.5\}$, a discrete variable used when $p_m = 1$, to record the position of the center of the palindromic structure; the range for c_m allows two types of palindromic structure (“even” and “odd” types, where the odd type contains a central unpaired column at $w = c_m$);
- $q_{m,w}$, a binary variable used when $w \geq 2c_m - w_m$ and $w < c_m$ to indicate whether columns w and $2c_m - w$ are paired, i.e. $\theta_{m,w,r} = \theta_{m,2c_m-w,\bar{r}}$ when (r, \bar{r}) is a Watson-Crick pair.

The motivations for these modeling choices that allow intermediate levels of constraints between the non palindromic and the fully palindromic structures stem from several considerations. First, our overarching goal was to set up an algorithm that could identify as many motifs as possible simultaneously which is incompatible with the idea of having

all or none of the motifs palindromic. Second, allowing separate sub-populations of motif components to coexist in our model (fully palindromic and non palindromic) would have certainly caused convergence issues due to the size of the dimension jump needed to move one motif from one population to another (dividing or multiplying by two the number of free parameters in θ_m). In contrast, our model that allows smooth transitions between non-palindromic and palindromic motif representation makes it possible to design algorithms that gradually increase or decrease the number of free parameters in θ_m . Finally, it should also be noticed that while some level of palindromic structure are often obvious it is unclear to which extent the biological motifs are fully palindromic or partially palindromic.

3.1.2 Statistical concepts of parameter and dimension estimation

For the task of estimating the parameters of our probabilistic model and adjusting its dimension, we adopted the methodological framework of Bayesian inference coupled with Markov chain Monte Carlo (MCMC) algorithms for its remarkable ability to accommodate complex models such as the one that we just presented for sequence data. The purpose of this subsection is to briefly introduce the statistical concepts that we used. The reader interested in a more detailed and more formal presentation can for instance read Robert (2007). In this subsection we use θ and y as generic notation for the parameters and the data, respectively.

Bayesian inference

The Bayesian inference is based on the "posterior distribution". In order to obtain this distribution, the Bayesian framework starts by placing a probability distribution on the parameters, called the "prior distribution", that allows to account for the initial knowledge on the parameters (if available), the prior is often written as

$$\pi(\theta). \tag{3.9}$$

When new data (y) become available, the information they contain regarding the model parameters is expressed in the "likelihood", which is the probability of the observed data given the model parameters, written as

$$\pi(y|\theta). \tag{3.10}$$

This information is then combined with the prior to produce the "posterior distribution".

Bayes' Theorem, an elementary identity in probability theory, states that the posterior is proportional to the prior times the likelihood. More precisely,

$$\begin{aligned}\pi(\theta|y) &= \frac{\pi(\theta)\pi(y|\theta)}{\pi(y)} \\ &\propto \pi(\theta)\pi(y|\theta),\end{aligned}\tag{3.11}$$

where the symbol \propto means “proportional to” in the sense that the equality holds up to a constant (here $\pi(y)$) which does not depend on θ .

In theory, the posterior distribution is always available, but in realistically complex models, the required analytic computations to obtain the normalizing constant, $\pi(y)$, are often intractable. Over several years, in the late 1980s and early 1990s, concomitantly with the development of the computational resources, it was realized that methods for drawing samples from the posterior distribution could be very widely applicable, since they do not necessarily require to compute the normalizing constant.

MCMC algorithms in Bayesian inference

Markov chain Monte Carlo (MCMC) methods refer to a body of algorithms that to construct Markov chains with an equilibrium distribution that correspond to the distribution of interest, in the context of Bayesian inference $\pi(\theta|y)$. Such a Markovian (hence non-independent) sample $(\theta^{(1)}, \theta^{(2)}, \dots, \theta^{(k)}, \dots)$ allows to study the characteristics of the target probability distribution and in particular to estimate integrals of interest (in particular expected values and variances) since by the law of large numbers

$$\frac{1}{K} \sum_{k=1}^K f(\theta^{(k)}) \xrightarrow{K \rightarrow +\infty} \mathbb{E}_{\pi(\cdot|y)}(f(\theta)).\tag{3.12}$$

In our framework, the dimension of the posterior distribution is high, and thus each sampling step updates one or a subset of the variables (block MCMC algorithm, see page 58) while others stay unchanged, using one of the following MCMC steps:

- Gibbs steps consist of sampling directly from the target conditional distribution (e.g. $\pi(\theta_1|\theta_2, y)$) if the parameter can be divided in two blocks $\theta = (\theta_1, \theta_2)$. Gibbs sampling is applicable when the joint distribution is difficult to sample from directly, but the conditional distribution of each variable is known and easy or easier to sample from.

- Metropolis–Hastings (MH) steps use a proposal density and reject some of the proposed moves. A MH step decomposes into
 1. Generate a candidate $\tilde{\theta}$ for the next sample by drawing from an instrumental distribution of density $q(\theta; \theta^{(k)})$, called the proposal. In multidimensional settings, the proposal can modify only a subset (block) of variables.
 2. Calculate the acceptance ratio

$$\begin{aligned}
 \alpha(\tilde{\theta}, \theta^{(k)}) &= \min \left\{ 1; \frac{\pi(\tilde{\theta}|y)q(\theta^{(k)}; \tilde{\theta})}{\pi(\theta^{(k)}|y)q(\tilde{\theta}; \theta^{(k)})} \right\} \\
 &= \min \left\{ 1; \underbrace{\frac{\pi(y|\tilde{\theta})}{\pi(y|\theta^{(k)})}}_{\text{ratio of likelihoods}} \times \underbrace{\frac{\pi(\tilde{\theta})}{\pi(\theta^{(k)})}}_{\text{ratio of priors}} \times \underbrace{\frac{q(\theta^{(k)}; \tilde{\theta})}{q(\tilde{\theta}; \theta^{(k)})}}_{\text{ratio of proposals}} \right\}.
 \end{aligned} \tag{3.13}$$

which will be used to decide whether to accept or reject the proposed candidate. Of note, a Gibbs step can be seen as a MH step in which the proposal matches the target distribution (acceptation ratio 1).

3. Accept with probability $\alpha(\tilde{\theta}, \theta^{(k)})$, i.e. set $\theta^{(k+1)} = \tilde{\theta}$ if accepted, $\theta^{(k+1)} = \theta^{(k)}$ if rejected.
- Reversible-jump MH is a variant of the Metropolis–Hastings algorithm that allows sampling from the distribution on spaces of varying dimensions (Green, 1995). Thus, the estimation is possible even if the number of parameters in the model is not known, which is the case in our model for cases like motif widths.

3.1.3 Prior settings and Directed Acyclic Graph of the hierarchical Bayesian model for sequence data

Prior for the sequence model

As explained in the previous subsection, the starting point of Bayesian inference is the definition of priors for the parameters of the model which will then be considered as random variables. Furthermore, the Bayesian framework allows to treat the variables that tune the

dimension of the model as other parameters. To avoid enriching the notations we will use the same notation for the random variable and its value (i.e. not use upper case for the random variable).

The priors that we used intend to be non-informative since we have little *a priori* knowledge about the characteristics of the motifs. Their choice also takes into account considerations on the tractability of the MCMC updates and thus, as usual in Bayesian inference, most of these priors are mutually independent conjugate priors.

The priors for the parameters that affect the dimension of the model are defined as

$$k_m \sim \text{Geom}(\text{probability of success} = p_k), \quad (3.14)$$

$$w_m \sim \text{Uniform}(\{w_{\min}, \dots, w_{\max}\}), \quad (3.15)$$

$$v \sim \text{Uniform}(\{0, \dots, v_{\max}\}), \quad (3.16)$$

$$p_m \sim \text{Bernoulli}(p_p), \quad (3.17)$$

and, when $p_m = 1$,

$$\pi(c_m | w_m) \propto a_c^{|c_m - (w_m + 1)/2|} \quad \text{for } c_m \in \{1.5, 2, 2.5, \dots, w_m - 0.5\}, \quad (3.18)$$

$$q_{m,w} | c_m \sim \text{Bernoulli}(p_q) \quad \text{for } w \in \{\max(1, 2c_m - w_m), \dots, \lfloor c_m - 0.5 \rfloor\}. \quad (3.19)$$

The emission parameters that describe the nucleotide composition in the background and in the motifs are modeled as drawn from the following independent Dirichlet distributions

$$\theta_{0,s} \sim \text{Dirichlet}_4(d_{\theta_0}, d_{\theta_0}, d_{\theta_0}, d_{\theta_0}) \quad \text{for } s \in \{\text{A}, \text{C}, \text{G}, \text{T}\}^r, \quad r = 0 : v, \quad (3.20)$$

$$\theta_{m,w} \sim \text{Dirichlet}_4(d_\theta, d_\theta, d_\theta, d_\theta). \quad (3.21)$$

The priors for the parameters that describe the distribution of the motif occurrences are defined as

$$\alpha_m \sim \text{Beta}(a_\alpha, b_\alpha), \quad (3.22)$$

$$\lambda_m | k_m \sim \text{Dirichlet}_{k_m+1}(d_\lambda, \dots, d_\lambda), \quad (3.23)$$

$$d_m | k_m \sim \text{Uniform}(k_m\text{-combinations of } \{2, \dots, \mathcal{L}\}), \quad (3.24)$$

$$r_m | w_m \sim \text{Uniform}(\{1, \dots, w_m\}). \quad (3.25)$$

Of note, α_m does not exist in the final implementation of our motif discovery algorithm since it is replaced by a function of the covariates that summarize the expression data. A typical choice that we used in our applications for the values for the parameters of the priors are $w_{\min} = 3$, $w_{\max} = 25$, $v_{\max} = 6$, $p_k = 0.25$, $d_\lambda = 1$, $p_p = 0.5$, $a_c = 0.5$, $p_q = 0.5$, $d_\theta = 0.25$, $d_{\theta_0} = 1$, $a_\alpha = 1$, $b_\alpha = 100$. The purpose of parameter $a_c \leq 1$ is to express a preference for a center of the palindromic structure near the center of the PWM. The choices of p_k (favoring a small k_m), d_θ (favoring PWM columns with high information content), a_α and b_α (favoring motifs with small number of occurrences) are discussed in Section 4.3.

Directed Acyclic Graph of the hierarchical Bayesian model for sequence data

The Directed Acyclic Graph (DAG) represents the factorization of the joint probability distribution of the variables (parameters, hidden variables, observed data). The DAG shown in Figure 3.1 corresponds to our probabilistic model of sequence data and the priors that we have just defined. Namely, the joint probability distribution writes

$$\begin{aligned}
& \pi(x, a, b, \alpha, d, \lambda, k, r, \theta_{1:\mathcal{M}}, q, c, p, w, \theta_0, v) \\
&= \pi(x|a, b, r, \theta_{1:\mathcal{M}}, \theta_0) \pi(a|\alpha, d, \lambda) \pi(b|a, r, w, \theta_{1:\mathcal{M}}) \\
&\quad \times \pi(r|w) \pi(d|k) \pi(\lambda|k) \pi(k) \\
&\quad \times \pi(\theta_0|v) \pi(v) \pi(\theta_{1:\mathcal{M}}|q, c, p, w) \pi(q|c, w) \pi(c|p, w) \pi(w) \\
&= \prod_{n,l} \pi(x_{n,l}|x_{n,1:l-1}, a, b, r, \theta_{1:\mathcal{M}}, \theta_0) \prod_{m,n} \pi(a_{m,n}|\alpha_m, d_m, \lambda_m) \prod_{n,l} \pi(b_{n,l}|a, r, w, \theta_{1:\mathcal{M}}) \\
&\quad \times \prod_m \pi(r_m|w_m) \prod_m \pi(d_m|k_m) \pi(\lambda_m|k_m) \pi(k_m) \\
&\quad \times \pi(\theta_0|v) \pi(v) \prod_m \pi(\theta_m|q_m, c_m, p_m, w_m) \pi(q_m|c_m, w_m) \pi(c_m|p_m, w_m) \pi(w_m), \quad (3.26)
\end{aligned}$$

where the term in red is specific to the θ -dependent weight mixture model for overlaps.

The DAG representation helps (via the construction of the corresponding undirected Moral graph in which edges have been added between all pairs of nodes with a common child) to visualize the Markov blanket for each variable and by turn the conditional independence relationships useful for the design of the MCMC steps targeting the different blocks of variables.

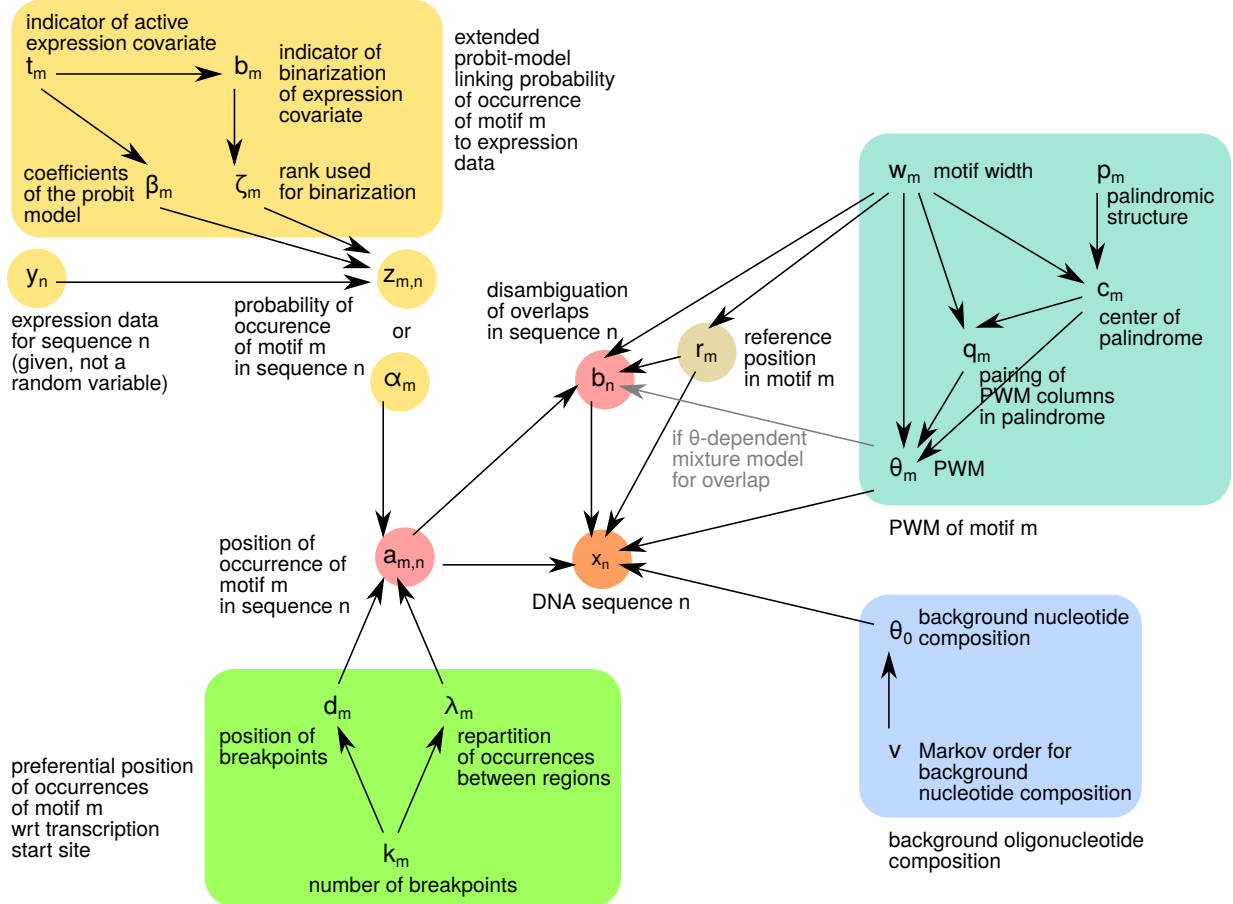


Figure 3.1: DAG of our model. Vertices represent the variables (parameters, hidden variables, observed data) and edges show the factorization of the joint probability distribution (see Equation (3.26)). Colored areas highlight groups of variables that contribute to a same aspect of the model. The upper-left part of the DAG dedicated to the incorporation of expression data in the sequence model which is presented in Section 3.2. In this representation we show the variables for only one motif m and one sequence n . The gray edge that links the disambiguation hidden variable B to θ is specific to the model of overlap that takes into account information content of the columns of the PWM (θ -dependent mixture). Of note, our implementation of the MCMC algorithms makes this option incompatible with the modeling of palindromic structures via the variables (p_m, c_m, q_m) .

3.1.4 Strategy and overview of the MCMC algorithm for the sequence model

A block MCMC sampler

A Markov Chain Monte Carlo (MCMC) algorithm was built with the joint posterior $(a, b, v, \theta, \theta_0, w, k, d, \lambda, \alpha|x)$ as target distribution. Since the dimension of the target is high, this MCMC algorithm is composed of many different steps whose purposes are to update separate subsets of variables. This type of algorithm is known as a block MCMC sampler (Andrieu et al., 2003) consists of a cycle through the different steps such as to allow the update of all the variables. Each step is designed to preserve the target distribution by sampling from the conditional distribution of a block of variables given all the other variables. Usually, the conditional independence properties make that conditional distributions involve only a subset of the other variables. In some cases, variables also disappear from the sampled conditional distribution because they are marginalized out. A subsequent step is then needed to “restore” the status of these variables that were marginalized out by sampling from their conditional distribution.

The steps of the block MCMC sampler that we implemented can be conveniently divided into three types (see also page 53):

- Metropolis-Hastings steps (abbreviated MH steps) that rely on a proposal coupled to an accept/reject procedure to preserve a target distribution (i.e. generating a Markov chain with the target as stationary distribution if repeated);
- Gibbs steps that consists of sampling directly from the target (conditional distribution of the variable or block of variables);
- Reversible-Jump MH steps that are generalization of the Metropolis-Hastings steps when the attempted moves change the dimension.

As already mentioned (page 53), Gibbs step can be seen as a special case of Metropolis-Hastings step where the proposal coincides with the target conditional distribution. Similarly, under circumstances where the probability distribution of the variables whose dimension is modified can be integrated-out, the Reversible-Jump MH steps can be done in a Gibbs manner, i.e. direct sampling from the conditional distribution of the variable that defines the dimension. We refer to this particular case as dimension-changing Gibbs step.

Overview of the 13 steps of the MCMC algorithm

In total, 13 steps have been designed taking into account the structure of dependence summarized in the DAG (Figure 3.1) and computational considerations. Using the notation θ for $(\theta_0, (\theta_m)_{m=1:\mathcal{M}})$ and in their order of appearance in the implemented sweeps, these steps that compose our algorithm are:

- Update the palindromic structure of motif m , described by (p_m, c_m, q_m) according to $p_m, c_m, q_m \mid a_m, b, r_m, w_m, x$ (Dimension-changing Gibbs step, θ_m is marginalized out). Details are provided page 75.
- Update the PWM θ_m , either according to $\theta_m \mid a, b, r, w, p_m, c_m, q_m, x$ (Gibbs step) if the equal weight mixture model for overlaps is used; or preserving $\theta_m \mid \theta_{-m}, a, b, r, w, p_m, c_m, q_m, x$ (MH step) if the θ -dependent weight mixture model for overlaps is used. Here, we use the notation θ_{-m} to refer to all the components of $\theta = (\theta_0, \theta_1, \dots, \theta_{\mathcal{M}})$ but θ_m (the same notation will also be used for other variables). Note that this update restores the status of θ_m which was marginalized out when updating the palindromic structure. Details are provided page 63.
- Update the expected proportion of sequences containing motif m , α_m , according to $\alpha_m \mid a_m$ (Gibbs step). Details are provided page 70.
- Update the position of the occurrences of the motifs, a , according to $a_m \mid a_{-m}, \theta, r, \alpha, x$ (Gibbs step, b is marginalized out). Details are provided page 61.
- Update motif width w_m , preserving $w_m, \theta_m, r_m, c_m, q_m \mid a, r_{-m}, \theta_{-m}, p_m, x$ (Reversible-Jump MH step, b is marginalized out). Details are provided page 65.
- Update the PWM θ_m by shifting motif m , preserving $\theta_m, r_m, c_m, q_m \mid a, r_{-m}, \theta_{-m}, p_m, x$ (Reversible-Jump MH step, with joint update of θ_m and r_m , b is marginalized out). This update which consists of removing one column of the PWM on a side and adding a column on the other side, is simply the coupling of two updates of motif width (decrease on left side coupled with increase on right side, decrease on right side coupled with increase on left side) whose details are provided page 65. It was introduced to allow the motif occurrences to “slide” along the sequences even when $w_m = w_{\max}$.

- Update the disambiguation variables for motif overlaps, b , according to $b \mid a, r, \theta, x$ (Gibbs step). Note that this update restores the status of b which was marginalized out when updating a and w . Details are provided page 62.
- Update the Markov order of the background, v , preserving $v \mid a, r, w, x$ (Reversible-Jump step, θ_0 is marginalized out). Details are provided page 69.
- Update the nucleotide composition of the background, θ_0 , according to $\theta_0 \mid v, a, r, w, x$ (Gibbs step). Note that this update restores the status of θ_0 which was marginalized out when updating v . Details are provided page 68.
- Update the position of the breakpoints defining the piece-wise constant probability density function modeling the positions of occurrences of motif m , d_m , preserving $d_m \mid k_m, a_m$ (MH step, λ is marginalized out). Details are provided page 72.
- Update the number of breakpoints in the piece-wise constant probability density function, k_m , preserving $k_m, d_m \mid a_m$ (Reversible-Jump MH step, note the joint update of d , while λ is marginalized out). Details are provided page 74.
- Update the expected fraction of motif occurrences found in each region of the piece-wise constant pdf, λ , according to $\lambda_m \mid k_m, d_m, a_m$ (Gibbs step). Note that this step restores the status of λ which was marginalized out when updating d and k . Details are provided page 71.
- Update the reference positions motif m , r_m , preserving $r_m, a_m \mid w_m, d_m, \lambda_m$ (MH step, note the joint update of a). Details are provided page 71.

Of note, this ordering of the steps makes that the variables that were marginalized out are restored before their reuse. Other orders would work provided that marginalized-out variables are appropriately restored (which may induce extra-computation).

Detecting errors in the MCMC algorithm

Design and implementation of an MCMC algorithms with so many steps is an error prone process.

In order to detect errors in the equations or implementation of our MCMC algorithm we also implemented a step consisting in simulating sequence x given all the other variables (i.e. $x \mid a, b, \theta$). When activated the MCMC algorithm targets the full joint distribution

$$x, a, b, \alpha, d, \lambda, k, r, \theta_{1:\mathcal{M}}, q, c, p, w, \theta_0, v$$

instead of the conditional

$$a, b, \alpha, d, \lambda, k, r, \theta_{1:\mathcal{M}}, q, c, p, w, \theta_0, v \mid x.$$

The marginal distribution of the parameters under this full joint distribution corresponds to the injected priors. Many errors in the implementation or in the equations skew these marginals and can thus be detected by comparing the marginals to the priors. For this purpose we typically work in a space of small dimension (e.g. taking $\mathcal{N} = 10$ and $\mathcal{M} = 2$) which considerably speeds up the convergence and thereby allows very precise comparisons of the marginals to the priors after a relatively short run (typical length between 1 and 30 minutes).

3.1.5 Details of the steps of MCMC algorithm for the sequence model

Instead of providing the details of the steps in their order of appearance in one sweep of the MCMC algorithm, we adopt here an order that intends to make the presentation easier to understand. We start by describing the update of the hidden variables a and b and then continue by describing, for each group of variables related to a same aspect of the model, the simple updates before the dimension-changing updates.

Update of a , the position of the occurrences of the motifs

This update is conducted as a Gibbs step, separately for each motif and each sequence, in which b is marginalized out. The conditional density of $a_{m,n} \mid a_{-m,n}, \theta, r, \alpha, x$, the position of motif m in sequence n , is computed, up to a normalizing constant, at each point of the

support $\{0, 1, \dots, \mathcal{L}\}$ using the formula

$$\begin{aligned}
\pi(a_{m,n}|a_{-m,n}, \theta, \alpha, r, x) &= \frac{\pi(a_{m,n}, a_{-m,n}, \theta, r, x)}{\pi(a_{-m,n}, \theta, \alpha, r, x)} \\
&\propto \pi(x_n|a_{m,n}, a_{-m,n}, \theta, r) \pi(a_{m,n}|\alpha) \\
&\propto \left[\prod_{l=a_{m,n}-r_m+1}^{a_{m,n}-r_m+w_m} \frac{\pi(x_{n,l}|a_{m,n}, a_{-m,n}, \theta, r)}{\pi(x_{n,l}|A_{m,n}=0, a_{-m,n}, \theta, r)} \right]^{\mathbb{I}_{\{a_{m,n} \neq 0\}}} \pi(a_{m,n}|\alpha),
\end{aligned} \tag{3.27}$$

where the term $\pi(a_{m,n}|\alpha)$ is given by Equation (3.8) and the term of the form $\pi(x_{n,l}|a_{m,n} = k, a_{-m,n}, \theta, r)$ are obtained by summing over all possible values of $b_{n,l}$

$$\begin{aligned}
\pi(x_{n,l}|a, \theta, r) &= \sum_{m=1:\mathcal{M}} \pi(x_{n,l}|B_{n,l} = m, a, \theta, r) \pi(B_{n,l} = m|a, \theta, r), \\
&= \sum_{m=1:\mathcal{M}} \theta_{m,l-(a_{m,n}-r_m)+1,x_{n,l}} \pi(B_{n,l} = m|a, \theta, r),
\end{aligned} \tag{3.28}$$

where $\pi(b_{n,l}|a, \theta, r)$ is given by Equation (3.6) or (3.7) (depending of the type of mixture model chosen for overlaps). The new value of $a_{m,n}$ is then drawn directly according to the conditional density computed at the $\mathcal{L} + 1$ points of the support.

Update of b , the disambiguation variables for motif overlaps

This update is done successively for all positions (n, l) by direct sampling from the conditional density of $b_{n,l}|a, r, \theta, x$ (Gibbs step) obtained as

$$\pi(b_{n,l}|a, r, \theta, x) \propto \pi(x_{n,l}|b_{n,l}, a, r, \theta) \pi(b_{n,l}|a, r, \theta),$$

where the two types of terms are given by Equations (3.3) and by Equations (3.6) or (3.7), depending of the type of mixture model chosen for motif overlaps. The new value of $b_{n,l}$ is then drawn directly according to the conditional density computed at the $\mathcal{M} + 1$ points of the support. Of note, this conditional distribution is trivial if zero or only one motif covers the position (n, l) , with probability of 1 for $b_{n,l} = 0$ if zero motif, and probability of 1 for $b_{n,l} = m$ if only motif m .

Update of θ_m , the PWM of motif m

As already stated, this update can consists of a Gibbs or a MH step depending on the type of mixture model chosen for motif overlaps (equal weight vs. θ -dependent weight). In all cases, the parameters of the different m and of the different columns w within each motif are updated successively (excepted for columns paired in a palindromic structure that requires simultaneous update) and the conditional distribution $\theta_m | \theta_{-m}, a, b, r, w, p_m, c_m, q_m, x$ is preserved.

Let's first describe the simple Gibbs step for the equal weight mixture. In this case, we need to distinguish the case of a column (m, w) that is not paired in a palindromic structure and the case of a paired column. The first case arises when the motif is not palindromic ($p_m = 0$), or the column cannot be paired due to its position relative to the center of the palindrome ($w < 2c_m - w_m$, $w = c_m$ or $w \geq 2c_m$), or the column could have been paired but is not paired ($q_{m, \min(w, 2c_m - w)} = 0$). In this first case, the conditional distribution writes

$$\begin{aligned} \theta_{m,w} | a, b, r, w, p_m, c_m, q_{m, \min(w, 2c_m - w)} = 0, x \\ \sim \text{Dirichlet}_4(\dots, d_\theta + c_{m,w,r}, \dots), \end{aligned} \quad (3.29)$$

where $c_{m,w,r}$ is the count of the nucleotide r emitted from column (m, w) defined as

$$\begin{aligned} c_{m,w,r} &\triangleq \sum_{n,l} \mathbb{I}\{x_{n,l} = r, b_{n,l} = m, a_{m,n} - r_m + w = l\} \\ &= \sum_n \mathbb{I}\{x_{n, a_{m,n} - r_m + w} = r, b_{n,l} = m\}. \end{aligned} \quad (3.30)$$

If the column is paired ($p_m = 1$ and $q_{m, \min(w, 2c_m - w)} = 1$), then $\theta_{m,w,r} = \theta_{m, 2c_m - w, \bar{r}}$. This implies the simultaneous update of the two PWM columns (w and $2c_m - w$) by a single drawing from the conditional distribution obtained by aggregating the counts from the two paired columns

$$\begin{aligned} \theta_{m,w} | a, b, r, w, p_m, c_m, q_{m,w} = 1, x \\ \sim \text{Dirichlet}_4(\dots, d_\theta + c_{m,w,r} + c_{m, 2c_m - w, \bar{r}}, \dots). \end{aligned} \quad (3.31)$$

Equation (3.29) is a direct and well known (see Section 2.3.2) consequence of the choice of a conjugate prior for $\theta_{m,w}$. From Equations (3.3) and (3.21) and using Bayes' rule, this

conditional distribution is obtained as follows

$$\begin{aligned}
\pi(\theta_{m,w}|x, \dots) &\propto \pi(\theta_{m,w}, x, \dots) \\
&\propto \pi(x|\theta_{m,w}, \dots)\pi(\theta_{m,w}|\dots) \\
&\propto \prod_{n,l} \prod_r \theta_{m,w,r}^{\mathbb{I}\{x_{n,l}=r, b_{n,l}=m, a_{m,n-r+w}=l\}} \times \prod_r \theta_{m,w,r}^{d_\theta-1} \\
&\propto \prod_r \theta_{m,w,r}^{d_\theta + \sum_{n,l} \mathbb{I}\{x_{n,l}=r, b_{n,l}=m, a_{m,n-r+w}=l\} - 1}, \tag{3.32}
\end{aligned}$$

where “...” is a convenient notation to refer to all the other variables (i.e. here besides x and $\theta_{m,w}$). Using the notation $c_{m,w,r}$ defined in Equation (3.30), the normalized version of this conditional density writes

$$\pi(\theta_{m,w}|x, \dots) = \frac{\Gamma(\sum_r d_\theta + c_{w,m,r})}{\prod_r \Gamma(d_\theta + c_{w,m,r})} \prod_r \theta_{m,w,r}^{d_\theta + c_{w,m,r} - 1}, \tag{3.33}$$

and corresponds to the Dirichlet distribution of Equation (3.29). The same line of reasoning leads to Equation (3.31) for paired columns in palindromic motifs.

The update of θ_m when the overlap model is a θ -dependent mixture is more complicated due to the gray arrow linking from θ_m to b in the DAG (Figure 3.1). The analogous of Equation (3.32) which gives the conditional density of $\theta_{m,w}$ is then

$$\begin{aligned}
\pi(\theta_{m,w}|x, b, \dots) &\propto \pi(\theta_{m,w}, x, b, \dots) \\
&\propto \pi(x|\theta_{m,w}, b, \dots)\pi(b|\theta_{m,w}, \dots)\pi(\theta_{m,w}|\dots) \\
&\propto \pi(b|\theta_{m,w}, \dots) \prod_r \theta_{m,w,r}^{d_\theta + \sum_{n,l} \mathbb{I}\{x_{n,l}=r, b_{n,l}=m, a_{m,n-r+w}=l\} - 1}, \tag{3.34}
\end{aligned}$$

where “...” refer to all the variables besides x , b and $\theta_{m,w}$. This conditional density can no longer be identified to the density of a Dirichlet distribution which precludes direct sampling (Gibbs step).

The workaround that we implemented is to build a MH step using the Dirichlet of Equation (3.29) as a proposal. In simple words, we propose a new value according to the conditional distribution under the simple model and use an accept-reject method to match it to the conditional under the more complex model. Denoting $q_{b,a,x}$ the density of this

proposal, the probability of acceptance of the proposed value $\tilde{\theta}_{m,n}$ writes

$$\alpha(\theta_{m,n}, \tilde{\theta}_{m,n}) = \min \left\{ 1; \frac{\pi(\tilde{\theta}_{m,w}|b, \dots) q_{b,a,x}(\theta_{m,w})}{\pi(\theta_{m,w}|b, \dots) q_{b,a,x}(\tilde{\theta}_{m,w,r})} \right\}. \quad (3.35)$$

Using Equation (3.34), this probability of acceptance simplifies to

$$\begin{aligned} \alpha(\theta_{m,n}, \tilde{\theta}_{m,n}) &= \min \left\{ 1; \frac{\pi(b|\tilde{\theta}_{m,w}, \dots)}{\pi(b|\theta_{m,w}, \dots)} \right\} \\ &= \min \left\{ 1; \prod_{n,l} \left(\frac{\pi(b_{n,l}|\tilde{\theta}_{m,w}, \dots)}{\pi(b_{n,l}|\theta_{m,w}, \dots)} \right)^{\mathbb{I}\{a_{m,n} \neq 0, l=a_{m,n}-r_m+w\}} \right\}, \end{aligned} \quad (3.36)$$

which involves to compute a product whose number of terms equals to the number of occurrences of motif m where column w is overlapped by an occurrence of another motif (the terms corresponding to the other occurrences are equal to 1). Each of these individual terms is easy to obtain from Equation (3.7).

Update of w_m , the width of motif m

This update relies on a Reversible-Jump MH step in which a new values $\tilde{w}_m, \tilde{\theta}_m, \tilde{r}_m, \tilde{c}_m, \tilde{q}_m$ are proposed for $w_m, \theta_m, r_m, c_m, q_m$. Importantly, the update does not change the position of the motifs (a is given) and each motif are treated successively. The variable b is marginalized out. The update preserves the conditional distribution $w_m, \theta_m, r_m, c_m, q_m | a, r_{-m}, \theta_{-m}, p_m, x$.

The proposed move consists of adding or removing only one column, i.e. $\tilde{w}_m = w_m + 1$ or $\tilde{w}_m = w_m - 1$. Four “directions” of modification of the PWM are possible depending on the sign (increase or decrease the width of the PWM) and on the side (modification on the right side or on the left side of the PWM). In order to preserve the position of the occurrences and since a is kept unchanged, if the modification is done on the left side the reference position needs to be shifted of one bp, i.e. $\tilde{r}_m = r_m + \tilde{w}_m - w_m$. Of note, this implies that a decrease on the left side is automatically rejected if $r_m = 1$ and a decrease on the right side is automatically rejected if $r_m = w_m$. If the modification is done on the left side, r_m is unchanged.

The update of the width preserves palindromic structures (p_m is given). To preserve the pairing of the columns, this implies, if the change is done on the left side of a palindromic motif, to shift the center of the palindrome c_m and the pairing variables $q_{m,w}$, i.e. $\tilde{c}_m = c_m + \tilde{w}_m - w_m$ and $\tilde{q}_{m,w+\tilde{w}_m-w_m} = q_{m,w}$ for $2c_m - w_m \leq w < c_m$ provided that $w + \tilde{w}_m -$

$w_m \geq 0$. This design of proposal means that we cannot accept a modification of the width that would shift the center of the palindrome outside of its allowed range, which happens when $c_m \in \{1.5, 2\}$ and a decrease of the width is attempted on the left side or $c_m \in \{w_m - 1, w_m - 0.5\}$ and a decrease of the width is attempted on the right side. To cope with the palindromic structure, we also need to define a proposal for the pairing status of the added column when an increase of the motif width is proposed and the palindromic structure defined by (p_m, w_m, c_m) allows the pairing of this new column, which happens with left increase when $2c_m < w_m + 1$ and right increase $2c_m > w_m + 1$ (this pairing concerns $\tilde{q}_{m,1}$ if left increase and $\tilde{q}_{2\tilde{c}_m - \tilde{w}_m}$ if right increase). For simplicity, we decided that the new column will be unpaired (i.e. $\tilde{q}_{m,1} = 0$ or $\tilde{q}_{2\tilde{c}_m - \tilde{w}_m} = 0$). A direct consequence of this choice is to forbid decrease moves that remove paired columns (this constraint can be seen in the ratio of proposal in the acceptance probability, Equation (3.40)).

Our proposal gives the same probability $1/4$ to each of the four directions of modification (left vs. right, increase vs. decrease) and it keeps unchanged the values of $\theta_{m,w}$ for those columns that are not directly concerned by the change. Namely, for a change on the right side, we have thus $\tilde{\theta}_m = (\theta_m, \tilde{\theta}_{m,w_m+1})$ if $\tilde{w}_m = w_m + 1$ and $\tilde{\theta}_m = (\theta_{m,-w_m})$ if $\tilde{w}_m = w_m - 1$. Similarly, for a change on the left side, $\tilde{\theta}_m = (\tilde{\theta}_{m,1}, \theta_m)$ if $\tilde{w}_m = w_m + 1$ and $\tilde{\theta}_m = (\theta_{m,-1})$ if $\tilde{w}_m = w_m - 1$. The proposal for a move that increases the width involves thus the drawing of a single four dimensional vector $\tilde{\theta}_{m,w}$ corresponding to the added PWM column. To maximize the chance of accepting the proposed modification, our proposal takes into account the nucleotide composition of the positions in the sequence that will be overlapped by this new PWM column if the move is accepted as follow

$$\begin{aligned}\tilde{\theta}_{m,w_m+1} &\sim \text{Dirichlet}_4(\dots, \max\{d_\theta + \tilde{c}_{m,w_m+1,r} - \tilde{o}_{m,w_m+1,r}, 1\}, \dots) \quad \text{if increase right} \\ \tilde{\theta}_{m,1} &\sim \text{Dirichlet}_4(\dots, \max\{d_\theta + \tilde{c}_{m,-1,r} - \tilde{o}_{m,-1,r}, 1\}, \dots) \quad \text{if increase left,}\end{aligned}\tag{3.37}$$

where $\tilde{c}_{m,w,r}$ corresponds to the count of nucleotide r at the sequence positions covered by the new column w (differing from $c_{m,w,r}$ of Equation (3.30) by the fact that b is not taken into account), and $\tilde{o}_{m,w,r}$ corresponds to the expected contribution of the other motifs that overlap the occurrences of motif m to the count of nucleotide r (according to the equal weight mixture model for motif overlap). Equation (3.37) sets the lower boundary of the parameters of the Dirichlet to 1. This avoids (very rare) cases where $d_\theta + \tilde{c}_{m,w,r} - \tilde{o}_{m,w,r}$ is negative and is therefore incompatible with the range of values allowed for the parameters

of a Dirichlet. The count $\tilde{c}_{m,w,r}$ is obtained as

$$\tilde{c}_{m,w,r} = \sum_n \mathbb{I}\{a_{m,n} \neq 0, x_{n,a_{m,n}-r_m+w} = r\}. \quad (3.38)$$

and the expected count $\tilde{o}_{m,w,r}$ as

$$\begin{aligned} \tilde{o}_{m,w,r} = & \sum_{n,m' \neq m} \left\{ \mathbb{I}\{a_{m,n} \neq 0, x_{n,a_{m,n}-r_m+w} = r\} \theta_{m',a_{m,n}-r_m+w-a_{m',n}-r_{m'},r} \right. \\ & \times \left. \frac{\mathbb{I}\{a_{m',n} \neq 0, 1 \leq a_{m,n} - r_m + w - a_{m',n} - r_{m'} \leq w_{m'}\}}{\sum_{m''} \mathbb{I}\{a_{m'',n} \neq 0, 1 \leq a_{m,n} - r_m + w - a_{m'',n} - r_{m''} \leq w_{m''}\}} \right\}. \end{aligned} \quad (3.39)$$

Denoting by q the proposal described in the preceding paragraphs, the acceptance ratio of the Reversible-Jump MH move writes

$$\begin{aligned} & \alpha((w_m, \theta_m, r_m, c_m, q_m), (\tilde{w}_m, \tilde{\theta}_m, \tilde{r}_m, \tilde{c}_m, \tilde{q}_m)) \\ &= \min \left\{ 1; \frac{\pi(\tilde{w}_m, \tilde{\theta}_m, \tilde{r}_m, \tilde{c}_m, \tilde{q}_m | a, r_{-m}, \theta_{-m}, x) q(w_m, \theta_m, r_m, c_m, q_m)}{\pi(w_m, \theta_m, r_m, c_m, q_m | a, r_{-m}, \theta_{-m}, x) q(\tilde{w}_m, \tilde{\theta}_m, \tilde{r}_m, \tilde{c}_m, \tilde{q}_m)} \right\} \\ &= \min \left\{ 1; \frac{\pi(x | \tilde{w}_m, \tilde{\theta}_m, \tilde{r}_m, a, r_{-m}, \theta_{-m})}{\pi(x | w_m, \theta_m, r_m, a, r_{-m}, \theta_{-m})} \times \frac{\pi(\tilde{w}_m, \tilde{\theta}_m, \tilde{r}_m, \tilde{c}_m, \tilde{q}_m)}{\pi(w_m, \theta_m, r_m, c_m, q_m)} \right. \\ & \quad \times \left. \frac{q(w_m, \theta_m, r_m, c_m, q_m)}{q(\tilde{w}_m, \tilde{\theta}_m, \tilde{r}_m, \tilde{c}_m, \tilde{q}_m)} \right\}. \end{aligned} \quad (3.40)$$

Of note, many terms simplify in this ratio. To illustrate its computation, we can take the example of an attempt to increase the width of the motif on the right side (i.e. $\tilde{w}_m = w_m + 1$, $\tilde{r}_m = r_m$, and $\tilde{\theta}_m = (\theta_m, \tilde{\theta}_{m,w_m+1})$). In this case, the probability of acceptance

$$\alpha((w_m, \theta_m, r_m, c_m, q_m), (\tilde{w}_m, \tilde{\theta}_m, \tilde{r}_m, \tilde{c}_m, \tilde{q}_m)) = \min\{1, A\}, \quad (3.41)$$

is build on acceptance ratio

$$\begin{aligned} A = & \prod_{n:a_{m,n} \neq 0} \frac{\pi(x_{n,a_{m,n}-r_m+w_m+1} | \tilde{w}_m, \tilde{\theta}_m, \tilde{r}_m, a, r_{-m}, \theta_{-m})}{\pi(x_{n,a_{m,n}-r_m+w_m+1} | w_m, \theta_m, r_m, a, r_{-m}, \theta_{-m})} \times \frac{\pi(\tilde{c}_m, \tilde{q}_m | p_m)}{\pi(c_m, q_m | p_m)} \\ & \times \frac{w_m}{w_m + 1} \times \frac{Dir_4(\tilde{\theta}_{m,w_m+1}; \dots, d_\theta, \dots)}{Dir_4(\tilde{\theta}_{m,w_m+1}; \dots, \max\{d_\theta + \tilde{c}_{m,-1,r} - \tilde{o}_{m,-1,r}, 1\}, \dots)}, \end{aligned} \quad (3.42)$$

where the term $w_m/(w_m + 1)$ corresponds to $\pi(\tilde{r}_m|\tilde{w}_m)/\pi(r_m|w_m)$. In this formula, the terms of the form $\pi(x_{n,l}|w_m, \theta_m, r_m, a, r_{-m}, \theta_{-m})$ are obtained by summing over all possible values of $b_{n,l}$ (see Equation (3.28)). The reverse modification from $(\tilde{w}, \tilde{\theta})$ to (w, θ) that decreases the width on the right side has probability of acceptance $\min\{1, 1/A\}$.

Update of θ_0 , the nucleotide composition of the background

The choice of a product of independent 4-dimensional Dirichlet priors for the parameter θ_0 of the Markov model of order v describing the background composition of the sequences (outside motifs) makes that the posterior is also a product of independent 4-dimensional Dirichlet distributions (property of conjugate prior). This allows direct sampling from the conditional distribution $\theta_0 | v, a, r, w, x$ (Gibbs step).

When we include the initial distribution (i.e. transitions for orders $0 \leq k < v$ that serve to model the first positions of the sequences), the number of Dirichlet distributions is $4^0 + 4^1 + \dots + 4^v = (4^{v+1} - 1)/3$, since $\theta_0 = (((\theta_{0,s,r})_{r \in \{A,C,G,T\}})_{s \in \{A,C,G,T\}})^k_{k=0:v}$.

Using the notation $o_{n,l}$ for the number of motif occurrences that overlap position (n, l) defined in Equation (3.5), the conditional density of θ_0 decomposes as the following product of conditional densities for each $\theta_{0,s}$,

$$\begin{aligned}
\pi(\theta_0|v, a, r, w, x) &\propto \pi(x|\theta_0, a, r, w)\pi(\theta_0|v) \\
&\propto \pi(\theta_0|v) \prod_{n,l} \pi(x_{n,l}|x_{n,max(1,l-v):l-1}, \theta_0, a, r, w)^{\mathbb{I}\{o_{n,l}=0\}} \\
&\propto \prod_{s \in (\{A,C,G,T\}^k)_{k=0:v}} \underbrace{\pi(\theta_{0,s}) \prod_{r \in \{A,C,G,T\}} \theta_{0,s,r}^{\sum_{n,l} \mathbb{I}\{o_{n,l}=0\} x_{n,max(1,l-v):l-1}=s, x_{n,l}=r}}_{\propto \pi(\theta_{0,s}|v, a, r, w, x)}.
\end{aligned} \tag{3.43}$$

The conditional density $\pi(\theta_{0,s}|v, a, r, w, x)$ can be rewritten, up to a normalizing constant, as

$$\pi(\theta_{0,s}|v, a, r, w, x) \propto \prod_{r \in \{A,C,G,T\}} \theta_{0,s,r}^{d_{\theta_0} + \sum_{n,l} \mathbb{I}\{o_{n,l}=0\} x_{n,max(1,l-v-1):l-1}=s, x_{n,l}=r}^{-1}, \tag{3.44}$$

which can be identified to the density of the Dirichlet distribution that we use to update

$\theta_{0,s},$

$$\begin{aligned} & \theta_{0,s}|v, a, r, w, x \\ & \sim \text{Dirichlet}_4(\dots, d_{\theta_0} + \sum_{n,l} \mathbb{I}\{o_{n,l} = 0 \mid x_{n,\max(1,l-v):l-1} = s, x_{n,l} = r\}, \dots). \end{aligned} \quad (3.45)$$

Update of v , the Markov order of the background

This dimension changing update is carried out as Reversible-Jump MH step preserving $v \mid a, r, w, x$ in which θ_0 is marginalized out.

Given the current Markov order v , a new value $\tilde{v} \in \{v-1, v+1\}$ is proposed, according to proposal q_v (in practice we use $q_v(v+1) = 0.5$ and $q_v(v-1) = 0.5$). This new value is accepted with probability

$$\begin{aligned} \alpha(v, \tilde{v}) &= \min \left\{ 1; \frac{\pi(\tilde{v}|a, r, w, x)q_{\tilde{v}}(v)}{\pi(v|a, r, w, x)q_v(\tilde{v})} \right\} \\ &= \min \left\{ 1; \frac{\pi(x|\tilde{v}, a, r, w)}{\pi(x|v, a, r, w)} \times \frac{\pi(\tilde{v})}{\pi(v)} \times \frac{q_{\tilde{v}}(v)}{q_v(\tilde{v})} \right\} \\ &= \min \left\{ 1; \frac{\prod_{n,l} \pi(x_{n,l}|x_{n,\max(1,l-\tilde{v}):l-1}, \tilde{v}, a, r, w)^{\mathbb{I}\{o_{n,l}=1\}}}{\prod_{n,l} \pi(x_{n,l}|x_{n,\max(1,l-v):l-1}, v, a, r, w)^{\mathbb{I}\{o_{n,l}=1\}}} \times \frac{\pi(\tilde{v})}{\pi(v)} \times \frac{q_{\tilde{v}}(v)}{q_v(\tilde{v})} \right\}, \end{aligned} \quad (3.46)$$

where $o_{n,l}$ corresponds to the number of motifs overlapping position (n, l) as defined in Equation (3.5). The products of conditional densities in which θ_0 is marginalized out that appear in the probability of acceptance can be obtained in close form. Using the notation $c_{0,s,r}^{(v)} = \sum_{n,l} \mathbb{I}\{o_{n,l} = 0, x_{n,\max(1,l-v-1):l-1} = s, x_{n,l} = r\}$ for the count of word (s, r) with

last character r in the background, these products are computed as

$$\begin{aligned}
& \prod_{n,l} \pi(x_{n,l} | x_{n,max(1,l-v):l-1}, v, a, r, w)^{\mathbb{I}\{o_{n,l}=1\}} \\
&= \int_{\theta_0} \pi(\theta_0) \prod_{n,l} \pi(x_{n,l} | x_{n,max(1,l-v):l-1}, \theta_0, v, a, r, w)^{\mathbb{I}\{o_{n,l}=1\}} d\theta_0 \\
&= \prod_{s \in (\{A,C,G,T\}^k)_{k=0:v}} \int_{\theta_{0,s}} \pi(\theta_{0,s}) \prod_{r \in \{A,C,G,T\}} \theta_{0,s,r}^{c_{0,s,r}^{(v)}} d\theta_{0,s} \\
&= \prod_{s \in (\{A,C,G,T\}^k)_{k=0:v}} \frac{\Gamma(\sum_{r \in \{A,C,G,T\}} d_{\theta_0})}{\prod_{r \in \{A,C,G,T\}} \Gamma(d_{\theta_0})} \int_{\theta_{0,s}} \prod_{r \in \{A,C,G,T\}} \theta_{0,s,r}^{d_{\theta_0} + c_{0,s,r}^{(v)} - 1} d\theta_{0,s} \\
&= \left(\frac{\Gamma(\sum_{r \in \{A,C,G,T\}} d_{\theta_0})}{\prod_{r \in \{A,C,G,T\}} \Gamma(d_{\theta_0})} \right)^{(4^{v+1}-1)/3} \prod_{s \in (\{A,C,G,T\}^k)_{k=0:v}} \frac{\Gamma(\sum_{r \in \{A,C,G,T\}} d_{\theta_0} + c_{0,s,r}^{(v)})}{\prod_{r \in \{A,C,G,T\}} \Gamma(d_{\theta_0} + c_{0,s,r}^{(v)})}.
\end{aligned} \tag{3.47}$$

Update of α_m , the expected proportion of sequences containing motif m

As explained in Section 3.1.3, the parameter α does not exist in the final model which takes into account expression data but its update is presented here for completeness. The expected fraction of the sequences containing the motif α_m are updated successively for each motif m by direct drawing from the conditional distribution $\alpha_m | a_m$. Such a Gibbs step is allowed by the choice of Beta distribution as prior for α_m (property of conjugate prior). Namely,

$$\begin{aligned}
\pi(\alpha_m | a_m, \dots) &\propto \pi(a_m | \alpha_m, \dots) \pi(\alpha_m) \\
&\propto \pi(\alpha_m) \prod_n \pi(a_{m,n} | \alpha_m, \dots) \\
&\propto \alpha_m^{a_\alpha - 1} (1 - \alpha_m)^{b_\alpha - 1} \prod_n \alpha_m^{\mathbb{I}\{a_{m,n} \neq 0\}} (1 - \alpha_m)^{\mathbb{I}\{a_{m,n} = 0\}} \\
&\propto \alpha_m^{a_\alpha + \sum_n \mathbb{I}\{a_{m,n} \neq 0\} - 1} (1 - \alpha_m)^{b_\alpha + \sum_n \mathbb{I}\{a_{m,n} = 0\} - 1},
\end{aligned} \tag{3.48}$$

where \dots denotes here all the variables but α_m and a_m . The left term corresponds to the density function of the Beta distribution that we used for the update of $\alpha_{m,n}$,

$$\alpha_{m,n} | a_{m,n} \sim \text{Beta}(a_\alpha + \sum_n \mathbb{I}\{a_{m,n} \neq 0\}, b_\alpha + \sum_n \mathbb{I}\{a_{m,n} = 0\}). \tag{3.49}$$

Update of r_m , the reference positions motif m .

This update consists of a simple MH step preserving $r_m, a_m \mid w_m, d_m, \lambda_m$ in which new values \tilde{r}_m and \tilde{a}_m are proposed for r_m and a_m . In practice, an attempt is made to increase or decrease r_m by 1 (with equal probabilities) and, simultaneously, to shift the positions of the motif encoded in a_m such as to maintain the positions of the occurrences of the motifs. Namely,

$$\tilde{a}_{m,n} = a_{m,n} + (\tilde{r}_m - r_m)\mathbb{I}\{a_{m,n} \neq 0\} \quad \text{for } n = 1 : \mathcal{N}. \quad (3.50)$$

Importantly, when shifting the $a_{m,n}$, we do not allow motif occurrences to disappear, i.e. are moved outside of the range $\{1, \dots, \mathcal{L}\}$. Therefore, the attempted move is automatically rejected when $\tilde{r}_m = r_m - 1$ if there exists (m, n) such as $a_{m,n} = 1$ and $\tilde{r}_m = r_m + 1$ if there exists (m, n) such as $a_{m,n} = \mathcal{L}$.

The probability of acceptance for this attempted move writes as

$$\begin{aligned} \alpha((r_m, a_m), (\tilde{r}_m, \tilde{a}_m)) &= \min \left\{ 1; \frac{\pi(\tilde{r}_m, \tilde{a}_m | x, \dots) q_{\tilde{r}_m, \tilde{a}_m}(r_m, a_m)}{\pi(r_m, a_m | x, \dots) q_{r_m, a_m}(\tilde{r}_m, \tilde{a}_m)} \right\} \\ &= \min \left\{ 1; \frac{\pi(x | \tilde{r}_m, \tilde{a}_m, \dots) \pi(\tilde{r}_m | \dots) \pi(\tilde{a}_m | \dots) q_{\tilde{r}_m, \tilde{a}_m}(r_m, a_m)}{\pi(x | r_m, a_m, \dots) \pi(r_m | \dots) \pi(a_m | \dots) q_{r_m, a_m}(\tilde{r}_m, \tilde{a}_m)} \right\} \\ &= \min \left\{ 1; \frac{\pi(\tilde{r}_m | w_m) \pi(\tilde{a}_m | d_m, \lambda_m)}{\pi(r_m | w_m) \pi(a_m | d_m, \lambda_m)} \right\}, \end{aligned} \quad (3.51)$$

where the ratio $\pi(\tilde{r}_m | w_m) / \pi(r_m | w_m)$ is 1 if $1 \leq \tilde{r}_m \leq w_m$ and 0 otherwise, and the terms of the form $\pi(a_m | d_m, \lambda_m)$ are given by Equation (3.8).

Update of λ_m , the expected fraction of motif occurrences found in each region of the piecewise constant pdf

The expected fraction of motif occurrences found in each region of the piecewise constant pdf, $\lambda_m = (\lambda_{m,1}, \dots, \lambda_{m,k_m+1})$ are updated for each motif m by direct drawing from the conditional density $\lambda_m \mid k_m, d_m, a_m$. Such a Gibbs step is allowed by the choice of Dirichlet

distribution as prior for $\lambda_m|k_m$ (property of conjugate prior). From Equation (3.8),

$$\begin{aligned}
\pi(\lambda_m|k_m, d_m, a_m) &\propto \pi(\lambda_m|k_m)\pi(a_m|d_m, k_m) \\
&\propto \prod_{k=1:k_m+1} \lambda_{m,k}^{d_\lambda-1} \prod_n \lambda_{m,k}^{\mathbb{I}\{d_{m,k-1} \leq a_{n,m} < d_{m,k}\}} \\
&\propto \prod_{k=1:k_m+1} \lambda_{m,k}^{d_\lambda + \sum_n \mathbb{I}\{d_{m,k-1} \leq a_{n,m} < d_{m,k}\}-1}, \tag{3.52}
\end{aligned}$$

which corresponds to the density of the Dirichlet

$$\lambda_m|k_m, d_m, a_m \sim \text{Dirichlet}_{k_m+1}(\dots, d_\lambda + c_{m,k}, \dots), \tag{3.53}$$

using the notation

$$c_{m,k} = \sum_n \mathbb{I}\{d_{m,k-1} \leq a_{n,m} < d_{m,k}\}. \tag{3.54}$$

Update of d_m , the positions of the breakpoints defining the piecewise constant pdf modeling the positions of occurrences of motif m

The positions $d_m = (d_{m,1}, \dots, d_{m,k_m})$ of the k_m breakpoints defining the piecewise constant pdf modeling the positions of occurrences of motif m are updated by a MH step preserving $d_m | k_m, a_m$ in which λ_m is marginalized out. The proposed \tilde{d}_m is obtained by choosing one of the k_m breakpoints and assigning it a new position in $\{2, \dots, \mathcal{L}\}$ among the $\mathcal{L} - k_m$ positions not occupied by the $k_m - 1$ unchanged breakpoints. This proposal corresponds to an uniform distribution over $k_m(\mathcal{L} - k_m)$ distinct \tilde{d}_m that can be obtained by changing the position of one of the breakpoints of d_m ,

$$q_{d_m}(\tilde{d}_m) = \frac{1}{k_m(\mathcal{L} - k_m)}. \tag{3.55}$$

The MH acceptance ratio simplifies due to the symmetry of the proposal ($q_{d_m}(\tilde{d}_m) = q_{\tilde{d}_m}(d_m)$) and the uniform prior on d_m over all the k_m -combinations in $\{2, \dots, \mathcal{L}\}$,

$$\begin{aligned}
\alpha(d_m, \tilde{d}_m) &= \min \left\{ 1; \frac{\pi(\tilde{d}_m | k_m, a_m, \alpha_m) q_{\tilde{d}_m}(d_m)}{\pi(d_m | k_m, a_m, \alpha_m) q_{d_m}(\tilde{d}_m)} \right\} \\
&= \min \left\{ 1; \frac{\pi(\tilde{d}_m | k_m) \pi(a_m | \tilde{d}_m, \alpha_m) q_{\tilde{d}_m}(d_m)}{\pi(d_m | k_m) \pi(a_m | d_m, \alpha_m) q_{d_m}(\tilde{d}_m)} \right\} \\
&= \min \left\{ 1; \frac{\pi(a_m | \tilde{d}_m, \alpha_m)}{\pi(a_m | d_m, \alpha_m)} \right\}
\end{aligned} \tag{3.56}$$

where the terms of the form $\pi(a_m | d_m, \alpha_m)$ are obtained by marginalizing out λ_m , which is made possible by the Dirichlet prior on this parameter. From Equation (3.8) and using the notation $c_{m,k}$ defined in Equation (3.54), we obtain

$$\begin{aligned}
&\pi(a_m | d_m, \alpha_m) \\
&= \int_{\lambda_m} \pi(a_m | \lambda_m, d_m, \alpha_m) \pi(\lambda_m | d_m) d\lambda_m \\
&= \frac{\alpha_n^{\sum_n \mathbb{I}\{a_{n,m} \neq 0\}}}{\prod_{k=1:k_m+1} (d_{m,k} - d_{m,k-1})^{c_{m,k}}} \times \frac{\prod_{k=1:k_m+1} \Gamma(d_\lambda)}{\Gamma(\sum_{k=1:k_m+1} d_\lambda)} \times \int_{\lambda_m} \prod_{k=1:k_m+1} \lambda_{m,k}^{d_\lambda + c_{m,k} - 1} d\lambda_m \\
&= \frac{\alpha_n^{\sum_n \mathbb{I}\{a_{n,m} \neq 0\}}}{\prod_{k=1:k_m+1} (d_{m,k} - d_{m,k-1})^{c_{m,k}}} \times \frac{\prod_{k=1:k_m+1} \Gamma(d_\lambda)}{\Gamma(\sum_{k=1:k_m+1} d_\lambda)} \times \frac{\Gamma(\sum_{k=1:k_m+1} d_\lambda + c_{m,k})}{\prod_{k=1:k_m+1} \Gamma(d_\lambda + c_{m,k})}.
\end{aligned} \tag{3.57}$$

Using this closed form for the marginalized density, the acceptance ratio of Equation (3.56) writes

$$\begin{aligned}
\alpha(d_m, \tilde{d}_m) &= \min \left\{ 1; \frac{\prod_{k=1:k_m+1} (d_{m,k} - d_{m,k-1})^{c_{m,k}}}{\prod_{k=1:k_m+1} (\tilde{d}_{m,k} - \tilde{d}_{m,k-1})^{\tilde{c}_{m,k}}} \right. \\
&\quad \times \left. \frac{\Gamma(\sum_{k=1:k_m+1} d_\lambda + \tilde{c}_{m,k}) \prod_{k=1:k_m+1} \Gamma(d_\lambda + c_{m,k})}{\Gamma(\sum_{k=1:k_m+1} d_\lambda + c_{m,k}) \prod_{k=1:k_m+1} \Gamma(d_\lambda + \tilde{c}_{m,k})} \right\},
\end{aligned} \tag{3.58}$$

where $\tilde{c}_{m,k}$ is the analogous of $c_{m,k}$ defined using \tilde{d}_m instead of d_m .

Update of k_m , the number of breakpoints in the piecewise constant pdf

The update of k_m is a Reversible-Jump MH step based on the proposition of a couple $(\tilde{k}_m, \tilde{d}_m)$ whose acceptance probability preserves the conditional distribution $k_m, d_m | a_m$ (λ_m is marginalized out). This update is very similar to the update of d_m (page 72) but consists of adding a removing (instead of moving) a breakpoint in d_m . When the addition of a new breakpoint is proposed (i.e. $\tilde{k}_m = k_m + 1$) its position is selected randomly among the $\mathcal{L} - k_m - 1$ positions in $\{2, \dots, \mathcal{L}\}$ that are not occupied by the k_m already existing breakpoints. The proposal for the move that decreases k_m consists of removing a randomly selecting and removing one breakpoint. The proposal density associated to $(\tilde{k}_m, \tilde{d}_m)$ generated by this procedure writes thus

$$q_{k_m, d_m}(\tilde{k}_m, \tilde{d}_m) = \frac{1}{2k_m} \mathbb{I}\{\tilde{k}_m = k_m - 1\} + \frac{1}{2(\mathcal{L} - k_m - 1)} \mathbb{I}\{\tilde{k}_m = k_m + 1\}. \quad (3.59)$$

Analogously to Equation (3.56), the probability of acceptance decomposes into

$$\alpha((k_m, d_m), (\tilde{k}_m, \tilde{d}_m)) = \min \left\{ 1; \frac{\pi(\tilde{k}_m) \pi(\tilde{d}_m | \tilde{k}_m) \pi(a_m | \tilde{d}_m, \alpha_m) q_{\tilde{k}_m, \tilde{d}_m}(k_m, d_m)}{\pi(k_m) \pi(d_m | k_m) \pi(a_m | d_m, \alpha_m) q_{k_m, d_m}(\tilde{k}_m, \tilde{d}_m)} \right\}. \quad (3.60)$$

Taking the example of an increase move (i.e. $\tilde{k}_m = k_m + 1$) and using the result of Equation (3.57), this probability can be written

$$\begin{aligned} \alpha((k_m, d_m), (\tilde{k}_m, \tilde{d}_m)) &= \min \left\{ 1; (1 - p_k) \frac{\binom{\mathcal{L}-1}{k_m}}{\binom{\mathcal{L}-1}{k_m+1}} \times \frac{\pi(a_m | \tilde{d}_m, \alpha_m)}{\pi(a_m | d_m, \alpha_m)} \times \frac{\mathcal{L} - k_m - 1}{k_m} \right\} \\ &= \min \left\{ 1; (1 - p_k) \frac{k_m + 1}{k_m} \right. \\ &\quad \times \frac{\prod_{k=1:k_m+1} (d_{m,k} - d_{m,k-1})^{c_{m,k}}}{\prod_{k=1:\tilde{k}_m+1} (\tilde{d}_{m,k} - \tilde{d}_{m,k-1})^{\tilde{c}_{m,k}}} \times \frac{\Gamma(d_\lambda) \Gamma(\sum_{k=1:k_m+1} d_\lambda)}{\Gamma(\sum_{k=1:\tilde{k}_m+1} d_\lambda)} \\ &\quad \left. \times \frac{\Gamma(\sum_{k=1:\tilde{k}_m+1} d_\lambda + \tilde{c}_{m,k}) \prod_{k=1:k_m+1} \Gamma(d_\lambda + c_{m,k})}{\Gamma(\sum_{k=1:k_m+1} d_\lambda + c_{m,k}) \prod_{k=1:\tilde{k}_m+1} \Gamma(d_\lambda + \tilde{c}_{m,k})} \right\}, \end{aligned} \quad (3.62)$$

where $\tilde{c}_{m,k}$ and $c_{m,k}$ are defined by Equation (3.54).

Update of (p_m, c_m, q_m) , the palindromic structure of motif m

The use of Dirichlet priors for the columns of θ_m allows, in the case of the equal weight mixture model for overlaps, to sample directly from the conditional distribution $p_m, c_m, q_m | a_m, b, r_m, w_m, x$ (θ_m is marginalized out). Of note, this Gibbs-type update changes the dimension of the model. Such a joint update is very efficient but could not be implemented for the θ -dependent mixture model for overlaps which precludes the use of this model when palindromic structures are allowed (alternative updates have not been implemented). In practice, sampling from $p_m, c_m, q_m | a, b, r_m, w_m, x$ is done in two steps:

- sample from $p_m, c_m | a_m, b, r_m, w_m, x$ (marginalizing out q_m),
- sample from $q_m | c_m, p_m, a_m, b, r_m, w, x$.

The densities needed for these two steps are obtained by appropriate summing and renormalization (for marginalization and conditioning) of the joint conditional density

$$\begin{aligned}
& \pi(p_m, c_m, q_m | a, b, r, w_m, x) \\
& \propto \pi(p_m, c_m, q_m | w_m) \pi(x | a, b, r, w_m, p_m, c_m, q_m) \\
& \propto \pi(p_m, c_m, q_m | w_m) \\
& \quad \times \underbrace{\pi((x_{n, a_m, n-r_m+w})_{(n,w): a_m, n \neq 0, w \in \{1, \dots, w_m\}, b_{n, a_m, n-r_m+w} = m} | a, b, r, w_m, p_m, c_m, q_m)}_{L(p_m, c_m, q_m)},
\end{aligned} \tag{3.63}$$

where $L(p_m, c_m, q_m)$ is a likelihood term that decomposes in a product of w_m terms. If $p_m = 0$ (no palindromic structure), c_m and q_m are not defined and the likelihood can be written

$$\begin{aligned}
L(p_m = 0, c_m = \emptyset, q_m = \emptyset) &= \prod_{w=1:w_m} \int_{\theta_{m,w}} \theta_{m,w}^{c_{m,w,r}} \pi(\theta_{m,w}) d\theta_{m,w} \\
&= \prod_{w=1:w_m} \frac{\Gamma(\sum_r d_\theta)}{\prod_r \Gamma(d_\theta)} \int_{\theta_{m,w}} \theta_{m,w}^{c_{m,w,r} + d_\theta - 1} d\theta_{m,w} \\
&= \prod_{w=1:w_m} \frac{\Gamma(\sum_r d_\theta)}{\prod_r \Gamma(d_\theta)} \frac{\prod_r \Gamma(d_\theta + c_{m,w,r})}{\Gamma(\sum_r d_\theta + c_{m,w,r})}.
\end{aligned} \tag{3.64}$$

When $p_m = 1$ (palindromic structure), we need to group the paired columns (where $\theta_{m,w,r} =$

$\theta_{m,2c_m-w,\bar{r}}$) to marginalize over $\theta_{m,w}$, which gives

$$\begin{aligned}
L(p_m = 0, c_m, q_m) &= \prod_{w=1:w_m} \left\{ \left[\frac{\Gamma(\sum_r d_\theta)}{\prod_r \Gamma(d_\theta)} \frac{\prod_r \Gamma(d_\theta + c_{m,w,r})}{\Gamma(\sum_r d_\theta + c_{m,w,r})} \right]^{\mathbb{I}\{w < 2c_m - w_m\} + \mathbb{I}\{w = c_m\} + \mathbb{I}\{w \geq 2c_m\}} \right. \\
&\quad \times \left[\mathbb{I}\{q_{m,w} = 0\} \left(\frac{\Gamma(\sum_r d_\theta)}{\prod_r \Gamma(d_\theta)} \right)^2 \frac{\prod_r \Gamma(d_\theta + c_{m,w,r})}{\Gamma(\sum_r d_\theta + c_{m,w,r})} \frac{\prod_r \Gamma(d_\theta + c_{m,2c_m-w,r})}{\Gamma(\sum_r d_\theta + c_{m,2c_m-w,r})} \right. \\
&\quad \left. \left. + \mathbb{I}\{q_{m,w} = 1\} \frac{\Gamma(\sum_r d_\theta)}{\prod_r \Gamma(d_\theta)} \frac{\prod_r \Gamma(d_\theta + c_{m,w,r} + c_{m,2c_m-w,\bar{r}})}{\Gamma(\sum_r d_\theta + c_{m,w,r} + c_{m,2c_m-w,\bar{r}})} \right]^{\mathbb{I}\{2c_m - w_m \leq w < c_m\}} \right\}. \tag{3.65}
\end{aligned}$$

Using the formulas of Equations (3.64) and (3.65) for the likelihood terms, we obtain the density needed for the step 1 sampling (p_m, c_m) by summing the joint density of Equation (3.63) over all possible values of q_m ,

$$\begin{aligned}
\pi(p_m, c_m | a, b, r, w, x) &\propto \sum_{q_m} \pi(q_m, c_m, p_m | a, b, r, w, x) \\
&\propto \pi(p_m) \pi(c_m | p_m) \sum_{q_m} \pi(q_m | c_m) L(p_m, c_m, q_m) \\
&\propto \pi(p_m) \pi(c_m | p_m) \prod_{w=1:w_m} \left\{ \left[\frac{\Gamma(\sum_r d_\theta)}{\prod_r \Gamma(d_\theta)} \frac{\prod_r \Gamma(d_\theta + c_{m,w,r})}{\Gamma(\sum_r d_\theta + c_{m,w,r})} \right]^{\mathbb{I}\{w < 2c_m - w_m\} + \mathbb{I}\{w = c_m\} + \mathbb{I}\{w \geq 2c_m\}} \right. \\
&\quad \times \left[(1 - p_q) \left(\frac{\Gamma(\sum_r d_\theta)}{\prod_r \Gamma(d_\theta)} \right)^2 \frac{\prod_r \Gamma(d_\theta + c_{m,w,r})}{\Gamma(\sum_r d_\theta + c_{m,w,r})} \frac{\prod_r \Gamma(d_\theta + c_{m,2c_m-w,r})}{\Gamma(\sum_r d_\theta + c_{m,2c_m-w,r})} \right. \\
&\quad \left. \left. + p_q \frac{\Gamma(\sum_r d_\theta)}{\prod_r \Gamma(d_\theta)} \frac{\prod_r \Gamma(d_\theta + c_{m,w,r} + c_{m,2c_m-w,\bar{r}})}{\Gamma(\sum_r d_\theta + c_{m,w,r} + c_{m,2c_m-w,\bar{r}})} \right]^{\mathbb{I}\{2c_m - w_m \leq w < c_m\}} \right\}. \tag{3.66}
\end{aligned}$$

For those columns that can be paired (i.e. w such as $\min(1, 2c_m - w_m) \leq w < c_m$), the conditional density used in the step 2 of the update (sampling $q_{m,w}$ given c_m) is obtained

from Equations (3.63), (3.65), and (3.19) as

$$\begin{aligned}
& \pi(q_{m,w} | c_m, p_m, a, b, r, w, x) \\
& \propto \mathbb{I}\{q_{m,w} = 0\} (1 - p_q) \left(\frac{\Gamma(\sum_r d_\theta)}{\prod_r \Gamma(d_\theta)} \right)^2 \frac{\prod_r \Gamma(d_\theta + c_{m,w,r})}{\Gamma(\sum_r d_\theta + c_{m,w,r})} \frac{\prod_r \Gamma(d_\theta + c_{m,2c_m-w,r})}{\Gamma(\sum_r d_\theta + c_{m,2c_m-w,r})} \\
& \quad + \mathbb{I}\{q_{m,w} = 1\} p_q \frac{\Gamma(\sum_r d_\theta)}{\prod_r \Gamma(d_\theta)} \frac{\prod_r \Gamma(d_\theta + c_{m,w,r} + c_{m,2c_m-w,\bar{r}})}{\Gamma(\sum_r d_\theta + c_{m,w,r} + c_{m,2c_m-w,\bar{r}})}.
\end{aligned} \tag{3.67}$$

3.2 Incorporating expression data in the sequence model

3.2.1 An extended probit model to use expression data as covariates

We already discussed our choice of incorporating expression data in our model as covariates that can carry information on the probability of occurrence of the motifs in the different sequences. To establish this link between expression data and probability of occurrence of a motif, we choose to adopt the methodological framework of the probit regression which presents the advantage of being relatively simple to implement, compared to the logit regression, in our context of Bayesian and MCMC-based inference. This simplicity stems from the availability of a data augmentation scheme in which a Gaussian latent variable model is introduced (Albert and Chib, 1993). We also realized that this model could easily accommodate an extension that could be seen as a binarization of expression covariates according to an automatically adjusted breakpoint which is very appealing in our modeling context for two reasons. First, it allows to model sharp switches in the probability of occurrences as a function of the position of the sequence in the expression space without imposing the probability of occurrence to jump between 0 and 1 between the sides of the breaks. Second, this binary representation of the covariate makes it possible to incorporate whole tree structures in the regression model. In this case, the binarization breakpoint is located along the branches of the tree instead of along a simple axis. In this section we start by describing our model for incorporation of covariates that have the form of a vector of continuous variables. Then, we describe (page 79) how the version of the model in which the covariate is binarized can also handle trees.

Vectors of continuous variables

We consider the case of a number \mathcal{C} of vectors of continuous variables denoted $y = (y_{n,c})_{n=1:\mathcal{N}, c=1:\mathcal{C}}$ that are to be used as covariates in the modeling of the events $\mathbb{I}\{A_{m,n} > 0\}$ (i.e. an occurrence of motif m is found in sequence n) for $n = 1 : \mathcal{N}$. Each motif m is modeled independently. According to the standard probit model, we would write,

$$\pi(A_{m,n} > 0|y) = \Phi(\beta_{m,0} + \sum_{c=1:\mathcal{C}} \beta_{m,c} y_{n,c}), \quad (3.68)$$

where the β 's are the regression coefficients ($\beta_{m,c} \in \mathbb{R}$) and Φ is the cumulative distribution function of the standard normal distribution which serves to map any value of $\beta_{m,0} + \sum_{c=1:\mathcal{C}} \beta_{m,c} y_{n,c}$ into a probability between 0 and 1.

To allow the automatic selection of the covariates that are relevant to predict the occurrences of motif m (choice of model dimension) and the aforementioned binarization of the covariates, our model involves the following parameters (variables in our Bayesian inference context),

- $t_m = (t_{m,1}, \dots, t_{m,C})$ where $t_{m,c} \in \{0, 1\}$ indicates whether covariate c should be taken into account in the probability of occurrence of motif m (i.e. $A_{m,n} > 0$);
- $\beta_m = (\beta_{m,0}, \beta_{m,1}, \dots, \beta_{m,C})$ where $\beta_{m,c} \in \mathbb{R}$ represents when $t_{m,c} = 1$ the parameter used in a probit regression that relates $y_{n,c}$ to the probability that $A_{m,n} > 0$, $\beta_{m,0}$ is the intercept parameter;
- $b_m = (b_{m,1}, \dots, b_{m,C})$ where $b_{m,c} \in \{0, 1\}$ indicates, when $t_{m,c} = 1$, whether the values in the vector $y_{.,c}$ are binarized ($b_{m,c} = 1$) or enter directly as they are in the probit model ($b_{m,c} = 0$);
- $\zeta_m = (\zeta_{m,1}, \dots, \zeta_{m,C})$ where $\zeta_{m,c} \in \{1, \dots, \mathcal{N} - 1\}$ indicates, when $t_{m,c} = 1$ and $b_{m,c} = 1$, the rank in $y_{.,c}$ of the value used for binarization; we use the notation $y_{[\zeta_{m,c}],c}$ for the corresponding cut-off.

In keeping with the probit regression framework, we write the probability of occurrence of motif m in sequence n as

$$\pi(A_{m,n} > 0|y, t, \beta, b, \zeta) = \Phi(\beta_{m,0} + \sum_c \beta_{m,c} \mathbb{I}\{t_{m,c} = 1\} \tilde{y}_{n,c}), \quad (3.69)$$

where $\tilde{y}_{n,c}$ corresponds to $y_{n,c}$ after an eventual binarization and writes

$$\begin{aligned} \tilde{y}_{n,c} \triangleq & \mathbb{I}\{b_{m,c} = 0\}y_{n,c} \\ & + \mathbb{I}\{b_{m,c} = 1\} \left[\left(\frac{\zeta_{m,c}}{\mathcal{N}} - 1 \right) \mathbb{I}\{y_{n,c} \leq y_{[\zeta_{m,c}],c}\} + \frac{\zeta_{m,c}}{\mathcal{N}} \mathbb{I}\{y_{n,c} > y_{[\zeta_{m,c}],c}\} \right]. \end{aligned} \quad (3.70)$$

When binarization is active (case $b_{m,c} = 1$), this formula maps $y_{n,c} > y_{[\zeta_{m,c}],c}$ to $\zeta_{m,c}/\mathcal{N}$ and $y_{n,c} \leq y_{[\zeta_{m,c}],c}$ to $\zeta_{m,c}/\mathcal{N} - 1$. Any mapping to other values than $\zeta_{m,c}/\mathcal{N}$ and $\zeta_{m,c}/\mathcal{N} - 1$ would be equivalent in terms of the distribution of $\mathbb{I}\{A_{m,n} > 0\}$ that it allows, provided that the β are readjusted. However, this mapping was chosen because it presents a two-fold advantage: it ensures the centering to 0 of $\tilde{y}_{n,c}$ whatever the value of $\zeta_{m,c}$ when $b_{m,c} = 1$ and it ensures a difference of 1 between the two possible values of the binarized covariates (hence preserving the interpretation of β across the possible values for $\zeta_{m,c}$). The centering is important for the mixing of the MCMC algorithm since the \mathcal{C} groups of variables $(t_{m,c}, b_{m,c}, \beta_{m,c}, \zeta_{m,c})$ are updated successively each c (see algorithm in Section 3.2.2).

Figures 3.2 and 3.3 intend to illustrate the shapes of the relationship between expression covariates and the probability of motif occurrence that are allowed by the simple probit model and by our extension based on covariate binarization. In one dimension (Figure 3.2), the simple probit model can account for a sharp switch between a region of low probability of occurrence and a region of high probability of occurrence but these regions have then probabilities close to 0 and 1. In contrast, the extended probit model with its three parameters $\zeta_{m,c}$, $\beta_{m,0}$ and $\beta_{m,c}$ can describe freely the position of the switch and the probability of motif occurrence on both side of switch. In two dimensions (Figure 3.3), the five parameters of the extended probit model can describe freely the position of the switches on each axis, but only three parameters are used to describe the probability of motif occurrence in the four regions defined by the switches. Indeed, when dimension \mathcal{C} increases the number ($2^{\mathcal{C}}$) of regions delineated by the switches increases exponentially but the number ($1 + 2\mathcal{C}$) of parameters increases only linearly as a result of the assumption of additive effects (on the probit scale). This slow increase makes it possible to use this model even when dimension \mathcal{C} is relatively high.

Trees: branch lengths and topology

Further extending the previous model described by Equations (3.69) and (3.70), we consider that covariates can come not only in the form of vectors of continuous variables but also in

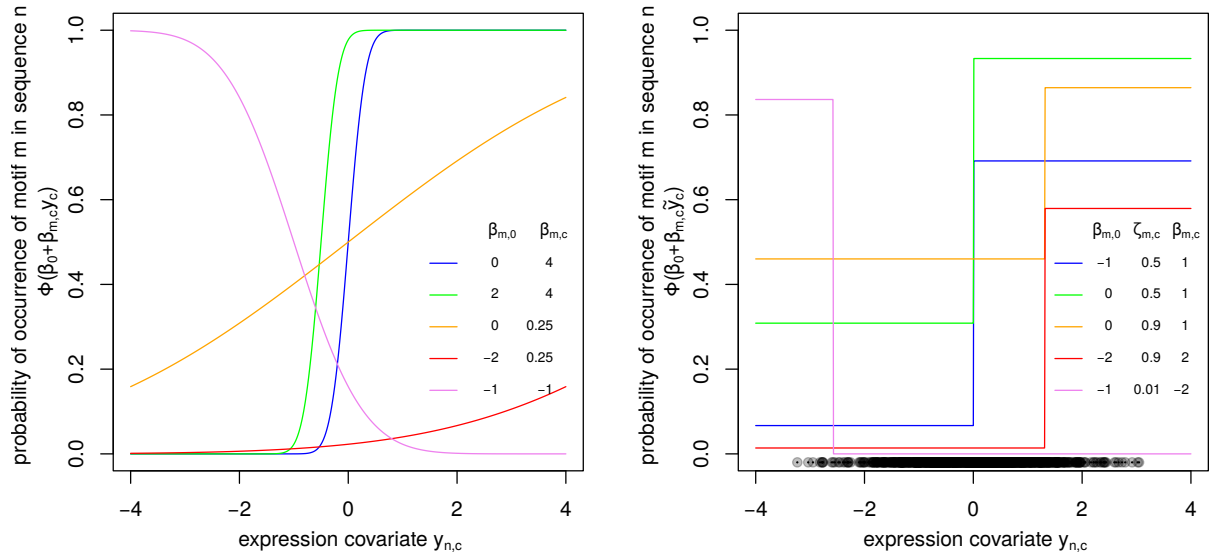


Figure 3.2: Graphical illustration of our probit model of probability of motif occurrence for one expression covariate. The probability of motif occurrence in sequence n is represented as a function of the value of the covariate $y_{n,c}$ for five sets of parameters. Left: simple probit regression model with two parameters $\beta_{m,0}$ and $\beta_{m,c}$ that relate the value of expression covariate c available for gene n ($y_{n,c}$) to the probability of occurrence of motif m . Right: extended probit regression model binarizing the expression covariate c according to a cut-off parameter $\zeta_{m,c}$ expressed as a rank (value of the cumulative distribution function in the insert legend). The horizontally aligned gray points at the bottom of the plot represents the values of $y_{n,c}$ that served for the binarization according to $\zeta_{m,c}$.

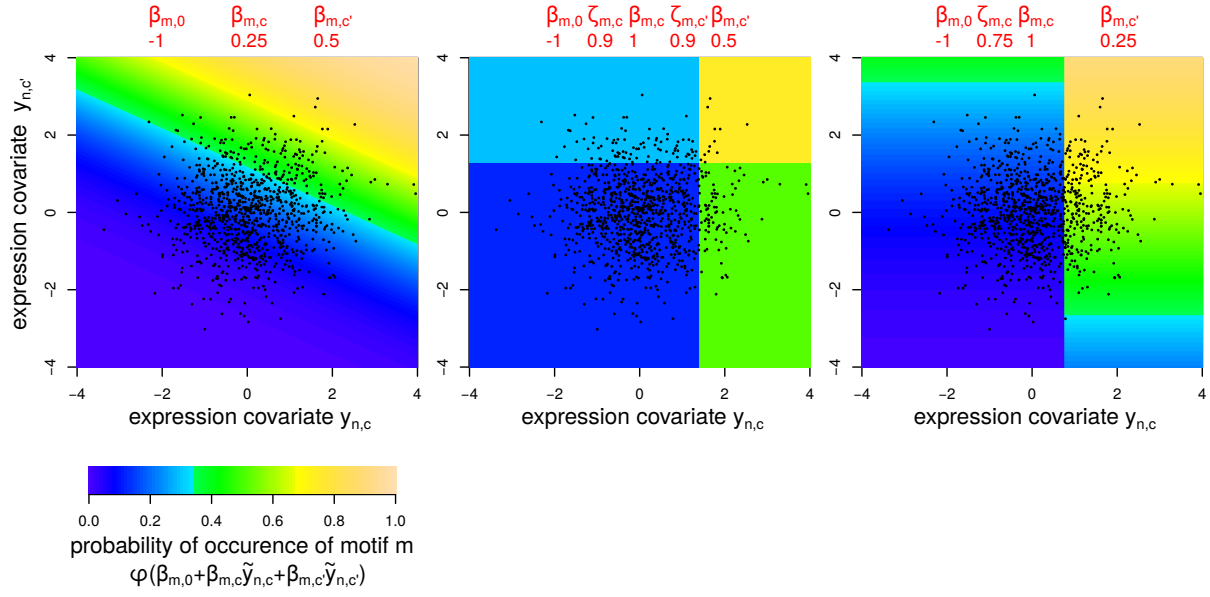


Figure 3.3: Graphical illustration of our probit model of probability of motif occurrence for two expression covariate. The probability of motif occurrence in sequence n is represented as a function of the value of the covariates $y_{n,c}$ and $y_{n,c'}$ for three sets of parameters. Left: simple probit regression model with three parameters $\beta_{m,0}$, $\beta_{m,c}$ and $\beta_{m,c'}$ that relate the value of expression covariates c and c' to the probability of occurrence of motif m . Middle: extended probit regression model binarizing the expression covariates c and c' according to cut-off parameters $\zeta_{m,c}$ and $\zeta_{m,c'}$ expressed as a rank (quantile in the insert legend). Right: extended probit regression model binarizing the expression covariates c but not c' . The black dots represents the values of $y_{n,c}$ and $y_{n,c'}$ that served for the binarization according to $\zeta_{m,c}$ and $\zeta_{m,c'}$.

the form of trees. For simplicity, we consider here rooted binary trees in which all the leaves are at a same distance from the root. In our context, these trees are obtained by hierarchical clustering of gene expression. Such a tree covariate c , contains a total of $2\mathcal{N} - 1$ nodes that decompose into \mathcal{N} terminal nodes or leaves and $\mathcal{N} - 1$ internal nodes. It can be entirely encoded (topology and branch lengths) as a vector internal node heights $h_c = (h_{c,i})_{i=1:\mathcal{N}-1}$ ordered such as $h_{c,i} \leq h_{c,i+1}$ and a $(\mathcal{N} - 1) \times 2$ matrix $s_c = (s_{c,i,1}, s_{c,i,2})_{i=1:\mathcal{N}-1}$ that identifies the subtrees merged at each internal node. In practice the $2\mathcal{N} - 2$ subtrees hanging below each node are identified by values in the ranges $\{-\mathcal{N}, \dots, -1\} \cup \{1, \mathcal{N} - 1\}$, with negative values corresponding to terminal nodes (in our cases the indexes of the sequences) and positive values corresponding to internal nodes ordered by height as in vector h_c .

Taken together, the vector h_c and the two-columns matrix s_c replace the vector y_c when the covariate c is a tree. Furthermore, the trees can be accommodated by our model only in the form of binarized covariates, hence $b_{m,c} = 1$. Compared to a covariate of type “vector”, the interpretation of the $\zeta_{m,c}$ is slightly changed such as it identifies a subtree (indexed as in the rows of matrix s_c). The mapping of the covariate to $\tilde{y}_{n,c}$, is also adapted. We used, $\tilde{y}_{n,c} = |\zeta_{m,c}|/\mathcal{N} - 1$ if sequence n belongs to the subtree $\zeta_{m,c}$ and $|\zeta_{m,c}|/\mathcal{N}$ otherwise, where we use the notation $|\zeta_{m,c}|$ for the number of sequences in the subtree $|\zeta_{m,c}|$. Figure 3.4 illustrates how the selection of a subtree allows to associate different probability of motif occurrence to different sequences.

Data augmentation

Following the data augmentation scheme described by Albert and Chib (1993), we introduce a unit variance Gaussian random variable $Z_{m,n}$ such that the probability modeled by the probit regression corresponds to the probability of $Z_{m,n} > 0$. The distribution of this random variable is

$$Z_{m,n}|y, t, \beta, b, \zeta \sim \mathcal{N}(\text{mean} = \beta_{m,0} + \sum_c \beta_{m,c} \mathbb{I}\{t_{m,c} = 1\} \tilde{y}_{n,c}, \text{var} = \sigma_z^2), \quad (3.71)$$

where σ_z^2 is set to 1 to match the variance of the standard Gaussian distribution whose probability density function is used to define the probability of motif occurrence in Equation (3.68) and (3.69). In the context of our model, $Z_{m,n} > 0$ means that an occurrence of motif m is found in sequence n and is thus equivalent to $A_{m,n} > 0$ while $Z_{m,n} \leq 0$ is

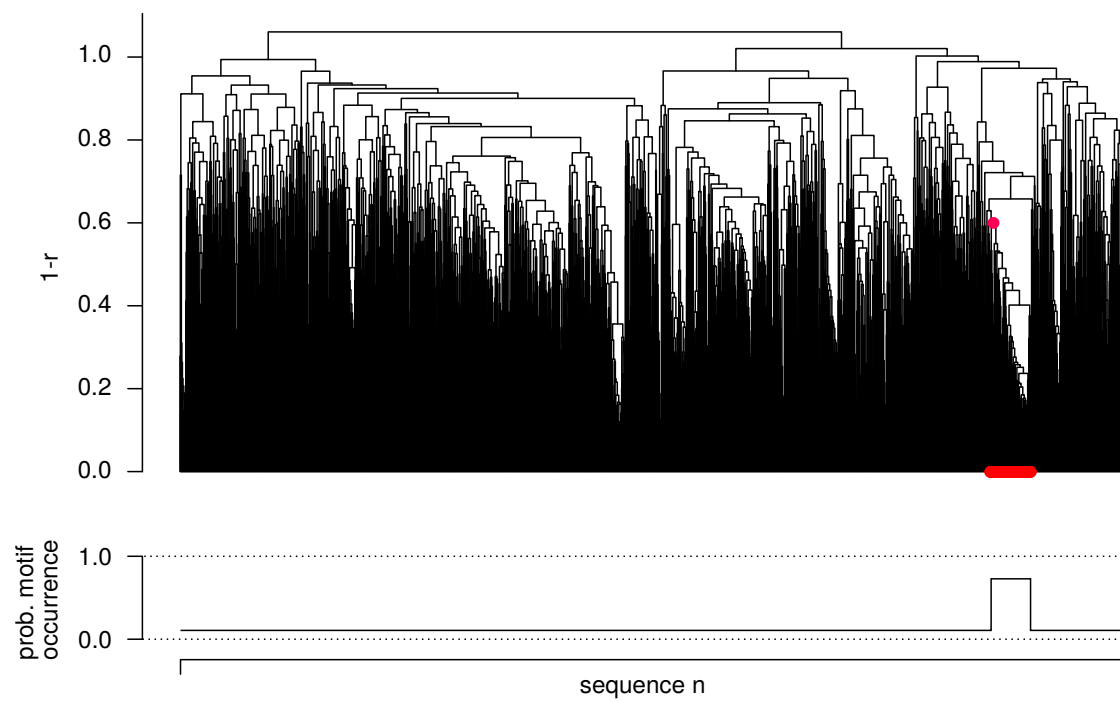


Figure 3.4: A graphical representation of how the process of selecting a cut on the PCA or the ICA can as well be applied on the clustering tree. In the case of the tree we are selecting a specific subtree (branch or merge) and all the sequences hanging from this subtree will have a different probability of having the motif compared to the rest of the tree.

equivalent to $A_{m,n} = 0$. This data augmentation scheme is justified by the fact that

$$\begin{aligned}
\pi(Z_{m,n} > 0 | y, t, \beta, b, \zeta) &= 1 - \Phi_{\text{mean}=\beta_{m,0}+\sum_c \beta_{m,c}\mathbb{I}\{t_{m,c}=1\}\tilde{y}_{n,c}, \text{var}=\sigma_z^2}(0) \\
&= 1 - \Phi_{\text{mean}=0, \text{var}=\sigma_z^2}(-\beta_{m,0} - \sum_c \beta_{m,c}\mathbb{I}\{t_{m,c}=1\}\tilde{y}_{n,c}) \\
&= \Phi_{\text{mean}=0, \text{var}=\sigma_z^2}(\beta_{m,0} + \sum_c \beta_{m,c}\mathbb{I}\{t_{m,c}=1\}\tilde{y}_{n,c}), \quad (3.72)
\end{aligned}$$

where the right-hand side term is the same as in Equation (3.69) when $\sigma_z^2 = 1$.

3.2.2 Inference

Priors and DAG for the extended probit

We typically have many covariates of type “vector”, each summarizing one aspect of the expression profiles; but few covariates of type “tree”, each providing a global summary of the expression profiles. For this reason, our prior concerning the active covariates (status encoded in variable t_c) of the extended probit distinguishes these two types of covariates.

For the covariates of type “tree”, we simply use independent Bernoulli priors

$$t_{m,c} \sim \text{Bernoulli}(p_{t,\text{tree}}). \quad (3.73)$$

For the covariates of type “vector”, we wanted a prior that favors the activation of a small number of covariates which conducted us to use a model in which activation of the different covariates are not independent. Namely, we use a geometric prior on the number of active covariates coupled with a uniform prior on which covariates are active (given the number of active covariates). The corresponding joint density function for the \mathcal{C}_v covariates of type “vector” is

$$\pi((t_{m,c})_{c=1:\mathcal{C}_v}) \propto p_{t,\text{vector}}^{\sum_{c=1:\mathcal{C}_v} t_{m,c}} \frac{1}{\binom{\mathcal{C}_v}{\sum_{c=1:\mathcal{C}_v} t_{m,c}}}, \quad (3.74)$$

in which the sum is over the covariates of type “vector”, and $p_{t,\text{vector}}$ corresponds to the parameter “probability of success” of the geometric prior. From Equation (3.74), the conditional distribution for any particular $t_{m,c}$ given the status of all other covariates of type

“vector” (denoted here $t_{m,-c}$) writes

$$\pi(t_{m,c}|t_{m,-c}) \propto \mathbb{I}\{t_{m,c} = 1\} p_{t,\text{vector}} \frac{\mathcal{C}_v - \sum_{-c} t_{m,c'} - 1}{\sum_{-c} t_{m,c'} + 1} + \mathbb{I}\{t_{m,c} = 0\}, \quad (3.75)$$

where $\sum_{-c} t_{m,c'}$ corresponds to the number of active components among all covariates of types “vector”, excepted c .

The prior used for the parameter $\beta_{m,c}$ are independent Gaussian distributions

$$\beta_{m,0} \sim \mathcal{N}(\text{mean} = m_{\beta_0}, \text{sd} = s_{\beta_0}), \quad (3.76)$$

$$\beta_{m,c} \sim \mathcal{N}(\text{mean} = m_{\beta}, \text{sd} = s_{\beta}). \quad (3.77)$$

A different prior is used for $\beta_{m,0}$ because it correspond to the intercept parameter whose meaning differs from those of the other coefficients of the probit model. In fact, the mean of the prior for the intercept $\beta_{m,0}$ tunes the expected number of occurrences for motif m in absence of link to expression covariates, with 0 corresponding to a presence of the motif in one half of the sequences (see Equation (3.69)). In contrast, m_{β} , the mean of the prior for $\beta_{m,c}$, will be set to 0, since the covariates are centered at 0 and have no reason to favor positive or negative links between expression covariates and probability of motif occurrence (the sign of a covariate such as obtained by PCA or ICA is arbitrary).

The prior used for the binarization of a covariate c of type “vector” is

$$b_{m,c}|t_{m,c} = 1 \sim \text{Bernoulli}(p_b). \quad (3.78)$$

The prior used for cut-off value used in the binarization process, $\zeta_{m,c}$, which is encoded as rank of the value in the vector or the index of the node in the tree, correspond to a uniform with respect to the values taken by the covariate (in case of covariate of type “vector”) or the length of the tree (in case of covariate of type “tree”). Hence, for a covariate of type “vector”,

$$\pi(\zeta_{m,c}|t_{m,c} = 1, b_{m,c} = 1) \propto y_{[\zeta_{m,c}]+1,c} - y_{[\zeta_{m,c}],c}. \quad (3.79)$$

For a covariate of type “tree”, the prior writes

$$\pi(\zeta_{m,c} | t_{m,c} = 1, b_{m,c} = 1) \propto (h_{p(\zeta_{m,c}),c} - h_{\zeta_{m,c},c}) \mathbb{I}\{n(\zeta_{m,c}) \geq n_{\zeta,\min}, \mathcal{N} - n(\zeta_{m,c}) \geq n_{\zeta,\min}\}, \quad (3.80)$$

where $p(\zeta_{m,c})$ is the parent node of $\zeta_{m,c}$ and $n(\zeta_{m,c})$ is the number of leafs in the subtree hanging below $\zeta_{m,c}$. In this prior, $n_{\zeta,\min}$ is a parameter that avoids locating a breakpoint in the tree at a position that would delineate a singleton or coexpression cluster that we judge too small.

In practice we used $p_{t,\text{tree}} = 0.5$, $p_{t,\text{vector}} = 0.5$, $p_b = 0.5$, $m_{\beta_0} = \Phi_{0,1}^{-1}(0.01)$ (i.e. $m_{\beta_0} \approx -2.326$), $s_{\beta_0} = \sqrt{0.2}$, $m_{\beta} = 0$, $s_{\beta} = 1$, and $n_{\zeta,\min} = 10$.

The Directed Acyclic Graph presented in Figure 3.1 encompasses the random variables t_m , β_m , b_m , ζ_m , and Z_m .

Overview of MCMC updates for the extended probit

In total, 3 MCMC steps have been designed to update the $z_{m,n}$'s, $\beta_{m,c}$'s, $\zeta_{m,c}$'s, $t_{m,c}$'s, $b_{m,c}$'s taking into account the structure of dependence summarized in the DAG (Figure 3.1) and computational considerations. These 3 steps replace the update of α_m in the sweep of the model without expression data described page 59. The update of a is also modified to be conditioned on z . In their order of appearance in the sweep these steps are:

- Update the data augmentation variable $z_{m,n}$ of the probit model according to $z_{m,n} | a_{m,n}, \beta_m, t_m, b_m, \zeta_m$ (Gibbs step). Details are provided page 87.
- Update the dimension changing variables $t_{m,c}, b_{m,c}, \zeta_{m,c}$ according to $t_{m,c}, b_{m,c}, \zeta_{m,c} | z_m, \beta_{m,-c}, t_{m,-c}, b_{m,-c}, \zeta_{m,-c}$ (dimension changing Gibbs step in which $\beta_{m,c}$ is marginalized out). Details are provided page 89.
- Update the coefficients of the probit regression $\beta_{m,c}$ according to $\beta_{m,c} | z_m, \beta_{m,-c}, t_m, b_m, \zeta_m$ (Gibbs step). Details are provided page 87.
- Update the position of the occurrences of the motif m , a_m , according to $a_m | a_{-m}, t_m, \beta_m, b_m, \zeta_m, \theta, r, \alpha, x$ (Gibbs step, b and z_m are marginalized out). Details are provided page 87.

The 3 first steps are relatively fast (in particular compared to the update of a) but they consider each covariate separately and they are done conditionally on $a_{m,n}$ which can cause

relatively slow mixing. This sequence of 3 steps is repeated 10 times at each sweep of the algorithm in our MCMC runs.

3.2.3 Details of the MCMC steps relative to the extended probit model

As in Section 3.1.5, instead of providing the details of the steps in their order of appearance in one sweep of the MCMC algorithm, we adopt for their detailed description an order that intends to make the presentation easier to understand.

Update of a , the position of the occurrences of the motifs

This step is identical to the step presented in Section 3.1.5, excepted that during the update of $a_{m,n}$ (the $a_{m,n}$ are updated successively), α_m is replaced by the probability of occurrence of motif m in sequence n given by the probit model. This probability of occurrence is $\Phi(\beta_0 + \sum_c \mathbb{I}\{t_{m,c} = 1\} \beta_{m,c} \tilde{y}_{n,c})$ (see Equation (3.69) and (3.70)).

Update of z , the data augmentation variable of probit model

The $z_{m,n}$'s are updated successively by direct sampling from the conditional distribution of $z_{m,n}$ given $a_{m,n}, \beta_m, t_m, b_m, \zeta_m$ (Gibbs step). This distribution is a truncated Gaussian since

$$\begin{aligned} & \pi(z_{m,n} | a_{m,n}, \beta_m, t_m, b_m, \zeta_m, \dots) \\ & \propto \pi(a_{m,n} | z_{m,n}, \beta_m, t_m, b_m, \zeta_m, \dots) \pi(z_{m,n} | \beta_m, t_m, b_m, \zeta_m) \\ & \propto (\mathbb{I}\{a_{m,n} > 0, z_{m,n} \geq 0\} + \mathbb{I}\{a_{m,n} = 0, z_{m,n} < 0\}) \\ & \quad \times \exp \left\{ -\frac{1}{2\sigma_z^2} (z_{m,n} - \beta_0 - \sum_c \mathbb{I}\{t_{m,c} = 1\} \beta_{m,c} \tilde{y}_{n,c})^2 \right\}. \end{aligned} \quad (3.81)$$

Hence, given that $\sigma_z^2 = 1$ (see Equation (3.71), but is still written as σ_z^2 to make apparent the homogeneity of the formulas), we draw $z_{m,n}$ from $\mathcal{N}(\text{mean} = \beta_0 + \sum_c \mathbb{I}\{t_{m,c} = 1\} \beta_{m,c} \tilde{y}_{n,c}, \text{var} = 1)$ truncated to \mathbb{R}^- if $a_{m,n} = 0$ and truncated to \mathbb{R}^+ if $a_{m,n} > 0$.

Update of β , the coefficients of probit regression

The $\beta_{m,c}$ are updated successively for the active covariates ($t_{m,c} = 1$) by direct sampling from the conditional distribution of $\beta_{m,c}$ given $z_m, \beta_{m,-c}, t_m, b_m, \zeta_m$. This update is made

possible by the choice of a Gaussian prior for β which is the conjugate prior for the mean of a Gaussian distribution. The possibility of such a Gibbs step is the fundamental purpose of the introduction of z . Of note, we update here each $\beta_{m,c}$ separately for simplicity but a joint update would also be possible.

The conditional distribution of $\beta_{m,c}$ is easy to write up to a constant,

$$\begin{aligned}
& \pi(\beta_{m,c} | z_m, \beta_{m,-c}, t_m, b_m, \zeta_m) \\
& \propto \pi(z_m | \beta_{m,c}, \beta_{m,-c}, t_m, b_m, \zeta_m) \pi(\beta_{m,c}) \\
& \propto \prod_n \exp \left\{ -\frac{1}{2\sigma_z^2} (z_{m,n} - \beta_0 - \sum_{c'} \mathbb{I}\{t_{m,c'} = 1\} \beta_{m,c'} \tilde{y}_{n,c'})^2 \right\} \\
& \quad \times \exp \left\{ -\frac{1}{2s_\beta^2} (\beta_{m,c} - m_\beta)^2 \right\} \\
& \propto \exp \left\{ -\frac{1}{2s_\beta^2} (\beta_{m,c} - m_\beta)^2 \right. \\
& \quad \left. - \frac{1}{2\sigma_z^2} \sum_n \left(z_{m,n} - \beta_0 - \mathbb{I}\{t_{m,c} = 1\} \beta_{m,c} \tilde{y}_{n,c} - \sum_{c' \neq c} \mathbb{I}\{t_{m,c'} = 1\} \beta_{m,c'} \tilde{y}_{n,c'} \right)^2 \right\} \\
& \propto \exp \left\{ -\frac{1}{2} \beta_{m,c}^2 \left[\frac{1}{s_\beta^2} + \frac{\mathbb{I}\{t_{m,c} = 1\}}{\sigma_z^2} \sum_n \tilde{y}_{n,c}^2 \right] \right. \\
& \quad \left. - \beta_{m,c} \left[\frac{m_\beta}{s_\beta^2} + \frac{\mathbb{I}\{t_{m,c} = 1\}}{\sigma_z^2} \sum_n \tilde{y}_{n,c} \left(z_{m,n} - \beta_0 - \sum_{c' \neq c} \mathbb{I}\{t_{m,c'} = 1\} \beta_{m,c'} \tilde{y}_{n,c'} \right) \right] \right\}.
\end{aligned} \tag{3.82}$$

For the conciseness of the subsequent formulas, we will use the notation

$$z_{m,n,-c} \triangleq z_{m,n} - \beta_0 - \sum_{c' \neq c} \mathbb{I}\{t_{m,c'} = 1\} \beta_{m,c'} \tilde{y}_{n,c'}. \tag{3.83}$$

The right-hand side term of Equation (3.82) corresponds to the density of the Gaussian distribution that we use to update $\beta_{m,c}$,

$$\pi(\beta_{m,c} | z_m, \beta_{m,-c}, t_m, b_m, \zeta_m) \sim \mathcal{N}(\text{mean} = \mu_{\beta_{m,c}|\dots}, \text{var} = \sigma_{\beta_{m,c}|\dots}^2), \tag{3.84}$$

where

$$\begin{aligned}\mu_{\beta_{m,c}|\dots} &= \left[\frac{m_\beta}{s_\beta^2} + \frac{\mathbb{I}\{t_{m,c} = 1\}}{\sigma_z^2} \sum_n \tilde{y}_{n,c} z_{m,n,-c} \right] \times \left[\frac{1}{s_\beta^2} + \frac{\mathbb{I}\{t_{m,c} = 1\}}{\sigma_z^2} \sum_n \tilde{y}_{n,c}^2 \right]^{-1} \\ \sigma_{\beta_{m,c}|\dots}^2 &= \left[\frac{1}{s_\beta^2} + \frac{\mathbb{I}\{t_{m,c} = 1\}}{\sigma_z^2} \sum_n \tilde{y}_{n,c}^2 \right]^{-1}.\end{aligned}\quad (3.85)$$

The update of $\beta_{m,0}$ relies on analogous equations in which the $\tilde{y}_{n,c}$'s are replaced by 1,

$$\pi(\beta_{m,0}|z_m, \beta_{m,-0}, t_m, b_m, \zeta_m) \sim \mathcal{N}(\text{mean} = \mu_{\beta_{m,0}|\dots}, \text{var} = \sigma_{\beta_{m,0}|\dots}^2), \quad (3.86)$$

with

$$\begin{aligned}\mu_{\beta_{m,0}|\dots} &= \left[\frac{m_{\beta_0}}{s_{\beta_0}^2} + \frac{1}{\sigma_z^2} \sum_n \left(z_{m,n} - \sum_c \mathbb{I}\{t_{m,c} = 1\} \beta_{m,c} \tilde{y}_{n,c} \right) \right] \times \left[\frac{1}{s_{\beta_0}^2} + \frac{\mathcal{N}}{\sigma_z^2} \right]^{-1} \\ \sigma_{\beta_{m,0}|\dots}^2 &= \left[\frac{1}{s_{\beta_0}^2} + \frac{\mathcal{N}}{\sigma_z^2} \right]^{-1}.\end{aligned}\quad (3.87)$$

Update of the dimension changing variables $t_{m,c}, b_{m,c}, \zeta_{m,c}$ of the extended probit

The possibility to carry out at a relatively small computational cost this update in a Gibbs manner (i.e. by direct sampling of the conditional) is a key ingredient of the usefulness of the extended probit model. This part of the algorithm consists of the joint update of the three variables $t_{m,c}, b_{m,c}, \zeta_{m,c}$ successively for each motif m and selected number of covariates c . In practice, we randomly select one tenth of the covariates of type “vector” and one covariate of type “tree”. The variable $\beta_{m,c}$ is marginalized out.

The conditional density needed for this update decomposes in three terms corresponding to the mutually exclusive cases of a covariate c which is not active ($t_{m,c} = 0$), a covariate c which is active and not binarized ($t_{m,c} = 1, b_{m,c} = 0$), and a covariate c which is active

and binarized ($t_{m,c} = 1, b_{m,c} = 1$). Namely, we can write

$$\begin{aligned}
& \pi(t_{m,c}, b_{m,c}, \zeta_{m,c} | z_m, \beta_{m,-c}, t_{m,-c}, b_{m,-c}, \zeta_{m,-c}) \\
& \propto \pi(z_m | t_{m,c}, b_{m,c}, \zeta_{m,c}, \beta_{m,-c}, t_{m,-c}, b_{m,-c}, \zeta_{m,-c}) \pi(t_{m,c}, b_{m,c}, \zeta_{m,c} | t_{m,-c}) \\
& \propto \mathbb{I}\{t_{m,c} = 0\} \pi(t_{m,c} | t_{m,-c}) \pi(z_m | t_m, \beta_{m,-c}, b_{m,-c}, \zeta_{m,-c}, t_{m,c} = 0) \\
& \quad + \mathbb{I}\{t_{m,c} = 1, b_{m,c} = 0\} \pi(t_{m,c} | t_{m,-c}) \pi(b_{m,c} | t_{m,c}) \\
& \quad \times \int_{\beta_{m,c}} \pi(z_m | t_m, b_m, \beta_m, \zeta_{m,-c}, t_{m,c} = 1, b_{m,c} = 0) \pi(\beta_{m,c} | t_{m,c}) d\beta_{m,c} \\
& \quad + \mathbb{I}\{t_{m,c} = 1, b_{m,c} = 1\} \pi(t_{m,c} | t_{m,-c}) \pi(b_{m,c}, \zeta_{m,c} | t_{m,c}) \\
& \quad \times \int_{\beta_{m,c}} \pi(z_m | t_{m,c}, b_{m,c}, \zeta_m, \beta_m, t_{m,c} = 1, b_{m,c} = 1) \pi(\beta_{m,c} | t_{m,c}) d\beta_{m,c}.
\end{aligned} \tag{3.88}$$

We will now see how these three terms can be computed, and in particular the third term which needs to be computed efficiently for every possible value of $\zeta_{m,c}$ to be able to draw directly from the conditional distribution of $t_{m,c}, b_{m,c}, \zeta_{m,c}$. The term needed for the first case ($t_{m,c} = 0$) writes simply

$$\begin{aligned}
\pi(z_m | t_m, \beta_{m,-c}, b_{m,-c}, \zeta_{m,-c}, t_{m,c} = 0) &= \prod_n \frac{1}{\sigma_z \sqrt{2\pi}} \exp \left\{ -\frac{1}{2\sigma_z^2} z_{m,n,-c}^2 \right\} \\
&= \frac{1}{\sigma_z^{\mathcal{N}} (2\pi)^{\mathcal{N}/2}} \exp \left\{ -\frac{\sum_n z_{m,n,-c}^2}{2\sigma_z^2} \right\},
\end{aligned} \tag{3.89}$$

where $z_{m,n,-c}$ corresponds to the definition given by Equation (3.83).

The term needed for the second case ($t_{m,c} = 1, b_{m,c} = 0$) is obtained as follow

$$\begin{aligned}
& \pi(z_m | t_m, \beta_{m,-c}, b_{m,-c}, \zeta_{m,-c}, t_{m,c} = 1, b_{m,c} = 0) \\
&= \int_{\beta_{m,c}} \pi(z_m | t_m, b_m, \beta_m, \zeta_{m,-c}, t_{m,c} = 1, b_{m,c} = 0) \pi(\beta_{m,c} | t_{m,c}) d\beta_{m,c} \\
&= \int_{\beta_{m,c}} \left[\prod_n \frac{1}{\sigma_z \sqrt{2\pi}} \exp \left\{ -\frac{1}{2\sigma_z^2} (z_{m,n,-c} - \beta_{m,c} y_{n,c})^2 \right\} \right. \\
&\quad \left. \times \frac{1}{s_\beta \sqrt{2\pi}} \exp \left\{ -\frac{1}{2s_\beta^2} (\beta_{m,c} - m_\beta)^2 \right\} \right] d\beta_{m,c} \\
&= \frac{1}{s_\beta \sigma_z^{\mathcal{N}} (2\pi)^{(\mathcal{N}+1)/2}} \times \exp \left\{ -\frac{\sum_n z_{m,n,-c}^2}{2\sigma_z^2} - \frac{m_\beta^2}{2s_\beta^2} \right\} \\
&\quad \times \int_{\beta_{m,c}} \exp \left\{ -\frac{1}{2} \beta_{m,c}^2 \left(\frac{\sum_n y_{n,c}^2}{\sigma_z^2} + \frac{1}{s_\beta^2} \right) + \beta_{m,c} \left(\frac{\sum_n y_{n,c} z_{m,n,-c}}{\sigma_z^2} + \frac{m_\beta}{s_\beta^2} \right) \right\} d\beta_{m,c} \\
&= \frac{1}{s_\beta \sigma_z^{\mathcal{N}} (2\pi)^{\mathcal{N}/2}} \exp \left\{ -\frac{\sum_n z_{m,n,-c}^2}{2\sigma_z^2} \right\} \times \left(\frac{\sum_n y_{n,c}^2}{\sigma_z^2} + \frac{1}{s_\beta^2} \right)^{-1/2} \\
&\quad \times \exp \left\{ -\frac{m_\beta^2}{2s_\beta^2} + \frac{1}{2} \left(\frac{\sum_n y_{n,c} z_{m,n,-c}}{\sigma_z^2} + \frac{m_\beta}{s_\beta^2} \right)^2 \left(\frac{\sum_n y_{n,c}^2}{\sigma_z^2} + \frac{1}{s_\beta^2} \right)^{-1} \right\}.
\end{aligned} \tag{3.90}$$

The derivation of the term needed for the third case ($t_{m,c} = 1, b_{m,c} = 1$) is similar to Equation (3.90), excepted that $\tilde{y}_{n,c}$ replaces $y_{n,c}$. However, it is important to note that $\tilde{y}_{n,c}$ depends on $\zeta_{m,c}$ (see Equation (3.70)) and will therefore be written here $\tilde{y}_{n,c,\zeta_{m,c}}$ to make this dependence apparent.

$$\begin{aligned}
& \pi(z_m | t_m, \beta_{m,-c}, b_{m,-c}, \zeta_m, t_{m,c} = 1, b_{m,c} = 1) \\
&= \frac{1}{s_\beta \sigma_z^{\mathcal{N}} (2\pi)^{\mathcal{N}/2}} \exp \left\{ -\frac{\sum_n z_{m,n,-c}^2}{2\sigma_z^2} \right\} \times \left(\frac{\sum_n \tilde{y}_{n,c,\zeta_{m,c}}^2}{\sigma_z^2} + \frac{1}{s_\beta^2} \right)^{-1/2} \\
&\quad \times \exp \left\{ -\frac{m_\beta^2}{2s_\beta^2} + \frac{1}{2} \left(\frac{\sum_n \tilde{y}_{n,c,\zeta_{m,c}} z_{m,n,-c}}{\sigma_z^2} + \frac{m_\beta}{s_\beta^2} \right)^2 \left(\frac{\sum_n \tilde{y}_{n,c,\zeta_{m,c}}^2}{\sigma_z^2} + \frac{1}{s_\beta^2} \right)^{-1} \right\}.
\end{aligned} \tag{3.91}$$

Using the expressions obtained for the three terms in Equations (3.89), (3.90), and (3.91),

we can rewrite the conditional joint density of $t_{m,c}, b_{m,c}, \zeta_{m,c}$ (Equation (3.88)) as

$$\begin{aligned}
& \pi(t_{m,c}, b_{m,c}, \zeta_{m,c} | z_m, \beta_{m,-c}, t_{m,-c}, b_{m,-c}, \zeta_{m,-c}) \\
& \propto \pi(z_m | t_{m,c}, b_{m,c}, \zeta_{m,c}, \beta_{m,-c}, t_{m,-c}, b_{m,-c}, \zeta_{m,-c}) \pi(t_{m,c}, b_{m,c}, \zeta_{m,c} | t_{m,-c}) \\
& \propto \mathbb{I}\{t_{m,c} = 0\} \pi(t_{m,c} | t_{m,-c}) \\
& \quad + \mathbb{I}\{t_{m,c} = 1, b_{m,c} = 0\} \pi(t_{m,c} | t_{m,-c}) \pi(b_{m,c} | t_{m,c}) \\
& \quad \quad \left(\frac{\sum_n y_{n,c}^2}{\sigma_z^2} + \frac{1}{s_\beta^2} \right)^{-1/2} \\
& \quad \quad \times \exp \left\{ -\frac{m_\beta^2}{2s_\beta^2} + \frac{1}{2} \left(\frac{\sum_n y_{n,c} z_{m,n,-c}}{\sigma_z^2} + \frac{m_\beta}{s_\beta^2} \right)^2 \left(\frac{\sum_n y_{n,c}^2}{\sigma_z^2} + \frac{1}{s_\beta^2} \right)^{-1} \right\} \\
& \quad + \mathbb{I}\{t_{m,c} = 1, b_{m,c} = 1\} \pi(t_{m,c} | t_{m,-c}) \pi(b_{m,c} | t_{m,c}) \pi(\zeta_{m,c} | t_{m,c}, b_{m,c}) \\
& \quad \quad \times \left(\frac{\sum_n \tilde{y}_{n,c,\zeta_{m,c}}^2}{\sigma_z^2} + \frac{1}{s_\beta^2} \right)^{-1/2} \\
& \quad \quad \times \exp \left\{ -\frac{m_\beta^2}{2s_\beta^2} + \frac{1}{2} \left(\frac{\sum_n \tilde{y}_{n,c,\zeta_{m,c}} z_{m,n,-c}}{\sigma_z^2} + \frac{m_\beta}{s_\beta^2} \right)^2 \left(\frac{\sum_n \tilde{y}_{n,c,\zeta_{m,c}}^2}{\sigma_z^2} + \frac{1}{s_\beta^2} \right)^{-1} \right\},
\end{aligned} \tag{3.92}$$

where the terms expressing the priors ($\pi(t_{m,c} | t_{m,-c})$, $\pi(b_{m,c} | t_{m,c})$ and $\pi(\zeta_{m,c} | t_{m,c}, b_{m,c})$) are given by Equations (3.73), (3.75), (3.78), (3.79), and (3.80).

As already mentioned, a key point is to be able to sample directly from this conditional density in order to compute efficiently (i.e. avoiding repeated summing over \mathcal{N}) the terms corresponding to $t_{m,c} = 1, b_{m,c} = 1$ for all possible values of $\zeta_{m,c}$. The computation of $\sum_n \tilde{y}_{n,c,\zeta_{m,c}}^2$ does not necessitate summing over \mathcal{N} since, from Equation (3.70), we have, in the case of a covariate of type “vector” where $\zeta_{m,c}$ corresponds to the rank of the cut-off value used for binarization,

$$\sum_n \tilde{y}_{n,c,\zeta_{m,c}}^2 = \zeta_{m,c} \left(\frac{\zeta_{m,c}}{\mathcal{N}} - 1 \right)^2 + (\mathcal{N} - \zeta_{m,c}) \left(\frac{\zeta_{m,c}}{\mathcal{N}} \right)^2 \tag{3.93}$$

and, in the case of covariate of type “tree” where $\zeta_{m,c}$ represents the index of a branch delineating a subtree whose size is denoted $|\zeta_{m,c}|$,

$$\sum_n \tilde{y}_{n,c,\zeta_{m,c}}^2 = |\zeta_{m,c}| \left(\frac{|\zeta_{m,c}|}{\mathcal{N}} - 1 \right)^2 + (\mathcal{N} - |\zeta_{m,c}|) \left(\frac{|\zeta_{m,c}|}{\mathcal{N}} \right)^2. \tag{3.94}$$

Similarly, the sum $\sum_n \tilde{y}_{n,c,\zeta_{m,c}} z_{m,n,-c}$ can be divided into two parts by separating the $y_{n,c}$

that are mapped to $\frac{\zeta_{m,c}}{\mathcal{N}} - 1$ and those that are mapped to $\frac{\zeta_{m,c}}{\mathcal{N}}$. In the case of a covariate c of type tree, the binarization relies on the cut-off value denoted $y_{[\zeta_{m,c}],c}$. Hence,

$$\begin{aligned} \sum_n \tilde{y}_{n,c,\zeta_{m,c}} z_{m,n,-c} &= \left(\frac{\zeta_{m,c}}{\mathcal{N}} - 1 \right) \sum_{n: y_{n,c} \leq y_{[\zeta_{m,c}],c}} z_{m,n,-c} \\ &\quad + \left(\frac{\zeta_{m,c}}{\mathcal{N}} \right) \left(\sum_n z_{m,n,-c} - \sum_{n: y_{n,c} \leq y_{[\zeta_{m,c}],c}} z_{m,n,-c} \right), \end{aligned} \tag{3.95}$$

where the partial sums $\sum_{n: y_{n,c} \leq y_{[\zeta_{m,c}],c}} z_{m,n,-c}$ corresponding to all possible values of $\zeta_{m,c}$ can be computed in one pass, by adding the $z_{m,n,-c}$'s in the order of increasing $y_{n,c}$'s. Similarly, in the case of a covariate of type “tree”, all the relevant partial sums are computed in a single bottom-up recursion (i.e. taking the subtrees defining the binarization in the order of increasing height as indexed in the rows of matrix s_c).

Importantly, the time complexity of the computation and sampling of the joint conditional of $t_{m,c}, b_{m,c}, \zeta_{m,c}$ is therefore only $O(\mathcal{N})$, and not $O(\mathcal{N}^2)$ as a first look at Equation (3.91) might have suggested.

Chapter 4

Data collection and method implementation

This chapter is dedicated to data collection, data preparation before making use of it, and the practical implementation of our methodology (C++ and settings for Bayesian priors). I discussed in Section 1.4 that the objective of the developed tool is to make simultaneous use of expression data and sequence data around the Transcription Start Sites (TSSs). In the first section, I discuss how we collected these data for *L. monocytogenes* as a first step toward our goal of detecting the main regulons of this bacterium (see Section 1.3). The second section is dedicated to expression data preparation, which includes applying hierarchical clustering to summarize the relation between genes in the form of a tree, and applying matrix factorization to summarize the relation between genes in the form of coordinates in a space of smaller dimensions. In the third section, I review analysis that were conducted in order to help us set the prior values on some important to-be-estimated parameters. Section four is dedicated to the computational implementation of the methodology discussed in Chapter 3, explaining how to access and run the tool, and detailing the contents of the output files.

4.1 Data collection

In order to run our tool for finding TFBSs, we needed to collect and prepare the input data. The tool handles two types of data as an input:

- Sequence data; the promoter regions around the TSSs (see subsection 4.1.1 for details).

- Expression data consisting of a large set of expression profiles for all the sequences considered in the search process.

4.1.1 Promoter sequences

To define promoter sequences and to align them with respect to TSSs, we used the large repertoire of 2,299 TSSs mapped by Wurtzel et al. (2012) at 1 bp resolution on *L. monocytogenes* EGDe genome sequence (see Section 1.2). In practice, data were extracted from Table S1 available online¹. Each TSS was given a unique ID defined by its exact position in the genome (TSS.4611.1 standing for TSS at position 4,611 on strand +1). The promoter sequences were defined as the 121 bp spanning from position -100 to +20 relative to each TSS in keeping with the size of the promoter regions analyzed for presence/absence of motifs recognized by TFs in Mäder et al. (2016) on *Staphylococcus aureus*.

We further trimmed the list of TSSs to avoid overlaps of these sequences on a same strand that could be misinterpreted as shared sequence features and thereby cause false positive motif predictions. Since each TSS is accompanied by a read count (Wurtzel et al., 2012) reflecting its level of experimental support and transcriptional activity, we decided that when two promoter regions overlap, the trimming procedure should keep the TSS with the highest read count. In practice, we used a simple greedy procedure that incorporated non-overlapping promoter regions one-by-one in the order of decreasing read-count which led us to a set of 1,545 non-overlapping promoter regions (67% of the initial list of TSSs).

4.1.2 Expression data set

The second step of data preparation was to collect expression profiles. For this purpose, we relied on the work done by Bécavin et al. (2017) to aggregate many published data sets for the listerionics website². As downloaded, the data had dimensions (3,159*254) where each row corresponds to a gene of *L.monocytogenes* EGDe and each column corresponds to the log2 of an expression ratio (log2 fold-change) comparing a pair of experimental conditions that could correspond to different mutants, growth conditions, or strains measured in a same study. The different studies aggregated in this data set used different technologies (one-color or two-color microarrays, RNA-Seq) but always the genome and annotation of *L.monocytogenes* EGDe as reference.

¹http://www.weizmann.ac.il/molgen/Sorek/listeria_browser/

²<https://listerionics.pasteur.fr/Listerionics/#bacnet.Listeria>

Since the studies used different technologies and versions it was not possible to always assess the complete list of genes, which resulted in having some columns with many missing values. To overcome that, the number of columns (pairs of conditions) was reduced from 254 to 165 after computing the median of the number of missing values per column and removing the columns with a number of missing values higher than 1.5 times this median. In parallel, rows (genes) were also trimmed based on the same criterion (decreasing the number of genes from 3,159 to 2,825. Figure 4.1 shows a heat map of the missing values inside the expression data matrix, highlighting the rows and columns that were kept and those that were removed.

From this table, and using the set of promoter regions that Wurtzel et al. (2012) associated to each of the TSSs, we matched the 1,545 non overlapping promoter regions to the 2,825 genes with expression data and obtained our final expression data-set representing 1,512 TSSs with 165 expression values.

4.2 Building covariates summarizing expression data

As mentioned in section 2.5.2 and explained in technical details in chapter 3, our developed tool incorporates covariates that summarize transcription where each covariate corresponds either to a numerical value, that may corresponds to the coordinate of the gene on an axis obtained by Principal or Independent Component Analysis (PCA or ICA), or to the position of the gene in a tree, typically obtained by hierarchical clustering. In this section, I explain the different approaches that were applied to summarize the expression data in order to use them as inputs to our motif discovery tool.

4.2.1 Hierarchical clustering trees

Hierarchical clustering is an iterative procedure for forming hierarchical groups of mutually exclusive subsets, each of which has members that are maximally similar with respect to specified characteristics. Given n sets, this procedure permits their reduction to $n - 1$ mutually exclusive sets by considering the union of all possible $n(n-1)/2$ pairs and selecting a union having a maximal value for an objective function, that reflects the criterion chosen by the investigator. This process that starts with singleton groups containing only one element is repeated until only one group containing all the element remains, the final result of the process is usually represented in the form of a rooted tree (dendrogram). Two very popular agglomerative algorithms, based on two different objective functions,

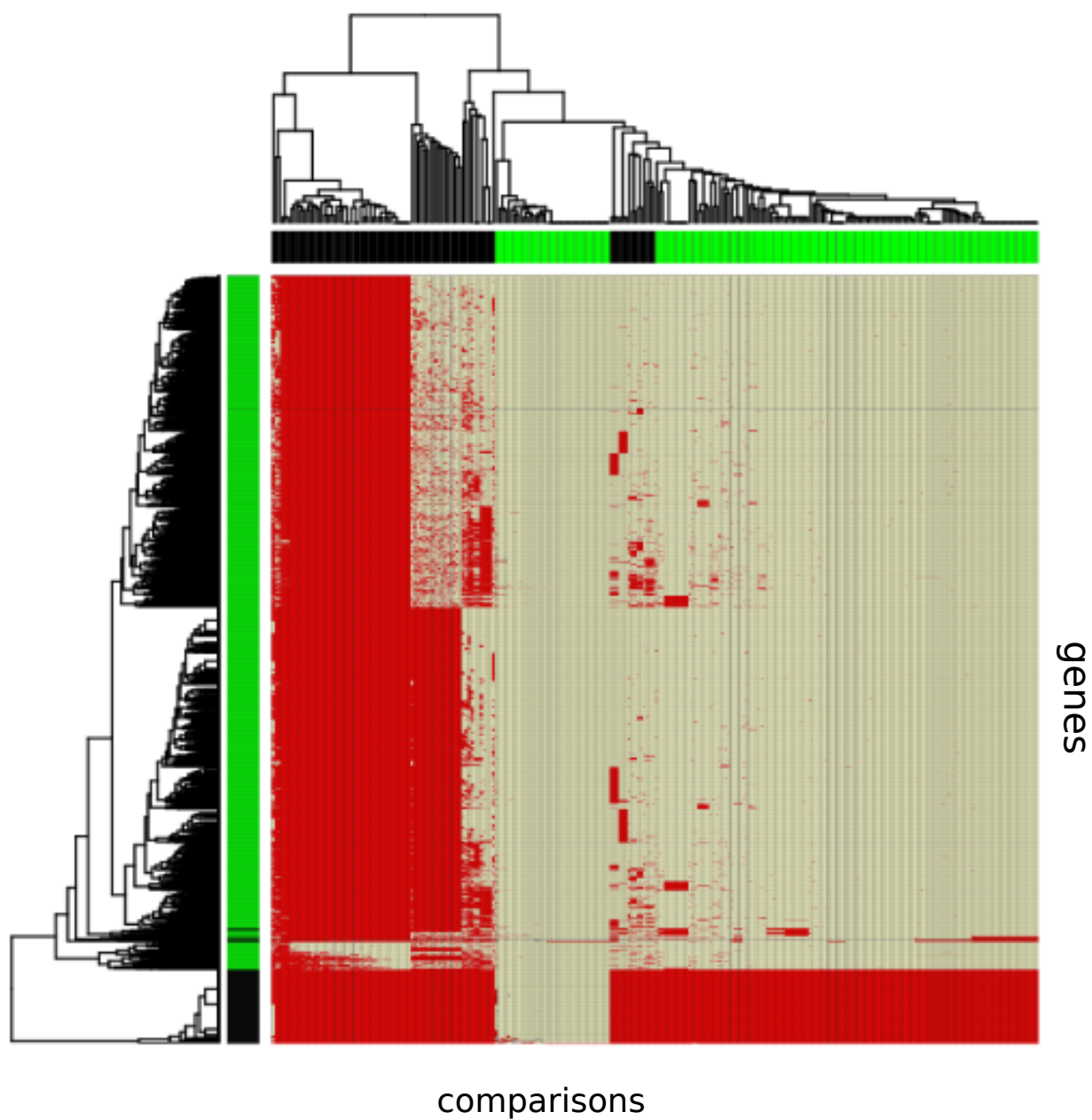


Figure 4.1: Heat map of the missing values inside the expression data matrix. The red color represents the missing values. The side bars are used to indicate rows and columns that were kept in the expression matrix (green for kept, black for removed).

were selected to perform hierarchical clustering on our data, Ward's and average linkage clustering. The explanation of the two methods is detailed below, and the resulting trees are shown in Figure 4.2 on page 100.

Ward's method (Ward Jr, 1963) or Ward's minimum variance algorithm performs hierarchical clustering by minimizing the total within-cluster variance. This method is implemented by finding, at each step, the pair of clusters that leads to minimum increase in total within-cluster variance after merging. This increase is a weighted squared distance between cluster centers. To apply Ward's algorithm, the initial distance between individual elements must be (proportional to) the Euclidean distance.

Unweighted average linkage clustering (Sokal, 1958) or unweighted pair group method with arithmetic mean (UPGMA) is the second method selected by us. The UPGMA algorithm constructs a tree in which distances between leaves (genes) are approximation of the values present in the initial pairwise distance matrix. To this end, at each step, the nearest two clusters (based on a specific distance function) are combined into a higher-level cluster. The distance between any two clusters A and B is given by:

$$\frac{1}{|A||B|} \sum_{a \in A} \sum_{b \in B} D(a, b). \quad (4.1)$$

$D(a, b)$ is a distance function specified by the user (i.e. values in a pairwise distance matrix). We selected D to be the "Pearson distance" $D = 1 - r$, where r is the Pearson correlation coefficient. The Pearson distance can be connected to the Euclidean distance (d) between centered and scaled vectors (here gene expression profiles) through equation

$$\begin{aligned} D &= 1 - r \\ &= \frac{d^2}{2n}. \end{aligned} \quad (4.2)$$

In practice, Ward's method was implemented by running "Ward.D2" method of the R function "hclust" on the matrix of Euclidean distances between genes after centering and scaling expression profiles across conditions. Average-link clustering was implemented by running the "average" method of the same R function on the matrix of Pearson correlation distances ($1 - r$), the relation between the two distance functions is shown in equation (4.2).

Before calculating the distance matrices, each column was duplicated with a reverse sign. This was done because each column in the original data matrix corresponds to a comparison between two conditions (log ratio), and the sign of the log ratio is arbitrary. As

well, rows were centered and scaled, since we did not want to give overwhelming importance to the magnitudes of the changes. Scaling is done by dividing the (centered) rows of the matrix by their standard deviations. The missing values that were not removed from the expression matrix in the cleaning process were replaced by "NA". When computing the Euclidean distance between two vectors, pairs containing "NA" values are excluded, and the sum is scaled up proportionally to the number of columns used. In the case of computing the Pearson correlation between two vectors, the "NA" values are excluded from each vector before the calculation.

Instead of cutting the tree at a given height such as to define a set of non-overlapping clusters, our approach to incorporate expression data, based on the binarization of the expression covariate, allows to directly handle a whole tree (see chapter 3). In this tree context, a cut is made automatically at branch-level to define two sets of genes that differ by the probability of occurrence of a motif. Since different branches in a same tree could be informative on the probability of occurrence of a same motif, we duplicated the two trees in the input file to the algorithm.

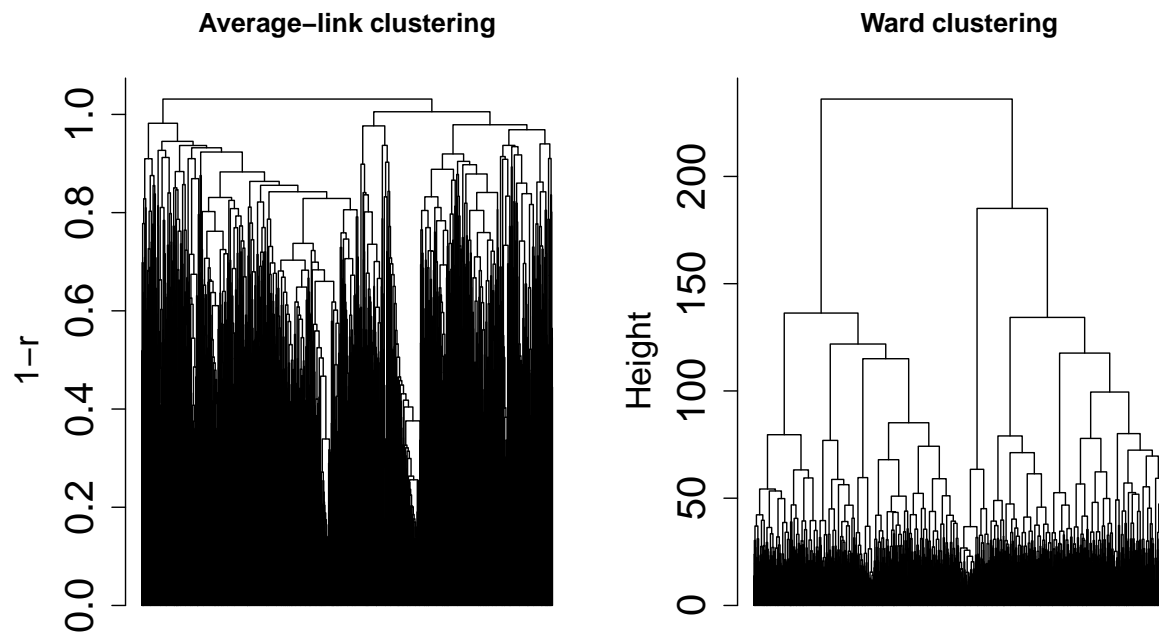


Figure 4.2: The final two trees that were used by our MCMC algorithm to produce the results on chapter 5. The left sub-figure shows the clustering tree that resulted from applying average-link clustering on Pearson distances between gene expression profiles. The sub-figure on the right shows the tree that resulted from applying Ward clustering on Euclidean distances between centered and scaled gene expression profiles.

4.2.2 Principle Component Analysis (PCA)

Principal component analysis (PCA) is a powerful and very popular dimensionality reduction procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. See Alter et al. (2000) for an early application of PCA on expression data in the search process for co-regulated genes.

PCA was one of the techniques applied by us to summarize expression data in the form of vector covariates. Since the original data matrix had dimensions $(1,512 * 165)$, the output of the PCA analysis is 165 components decreasingly ordered according to the variance percentage captured by each component out of the total variance in the data. We choose to incorporate only the first 20 components (capturing almost 70% of the variance of the original data). Figure 4.3 illustrates how somewhat similar information in the expression data can be captured either by applying PCA or hierarchical clustering.

In practice, we used the "prcomp" function in R that applies PCA by a singular value decomposition

$$Y = U\Sigma W^T, \quad (4.3)$$

where Σ is a $(1,512 * 165)$ rectangular diagonal matrix of positive numbers, called the singular values of Y ; U is a $(1,512 * 1,512)$ matrix, the columns of which are orthogonal unit vectors of length 1,512 called the left singular vectors of Y ; and W is $(165 * 165)$ whose columns are orthogonal unit vectors of length 165 and called the right singular vectors of Y . The function "prcomp" outputs two matrices $Y^p = U\Sigma$ and W , equation (4.3) holds if we consider all the principle components ($K = 165$). Using the notation Y_K^p for the first K columns of Y^p , equation (4.3) writes

$$Y = Y_K^p W_K^T + \epsilon, \quad (4.4)$$

where Y_K^p has dimensions $(1,512 * K)$ and corresponds to the coordinates of the genes in the subspace defined by the K first PC axes, W_K has dimensions $(165 * K)$ and corresponds to the coefficients of the linear transformation used to define the PC axes, and ϵ represents the residuals. Figure 4.4 shows the relation between the choice of K and the value of ϵ . Of note, The missing values that were not removed in the cleaning process were replaced by 0 since the function "prcomp" accepts only numeric values.

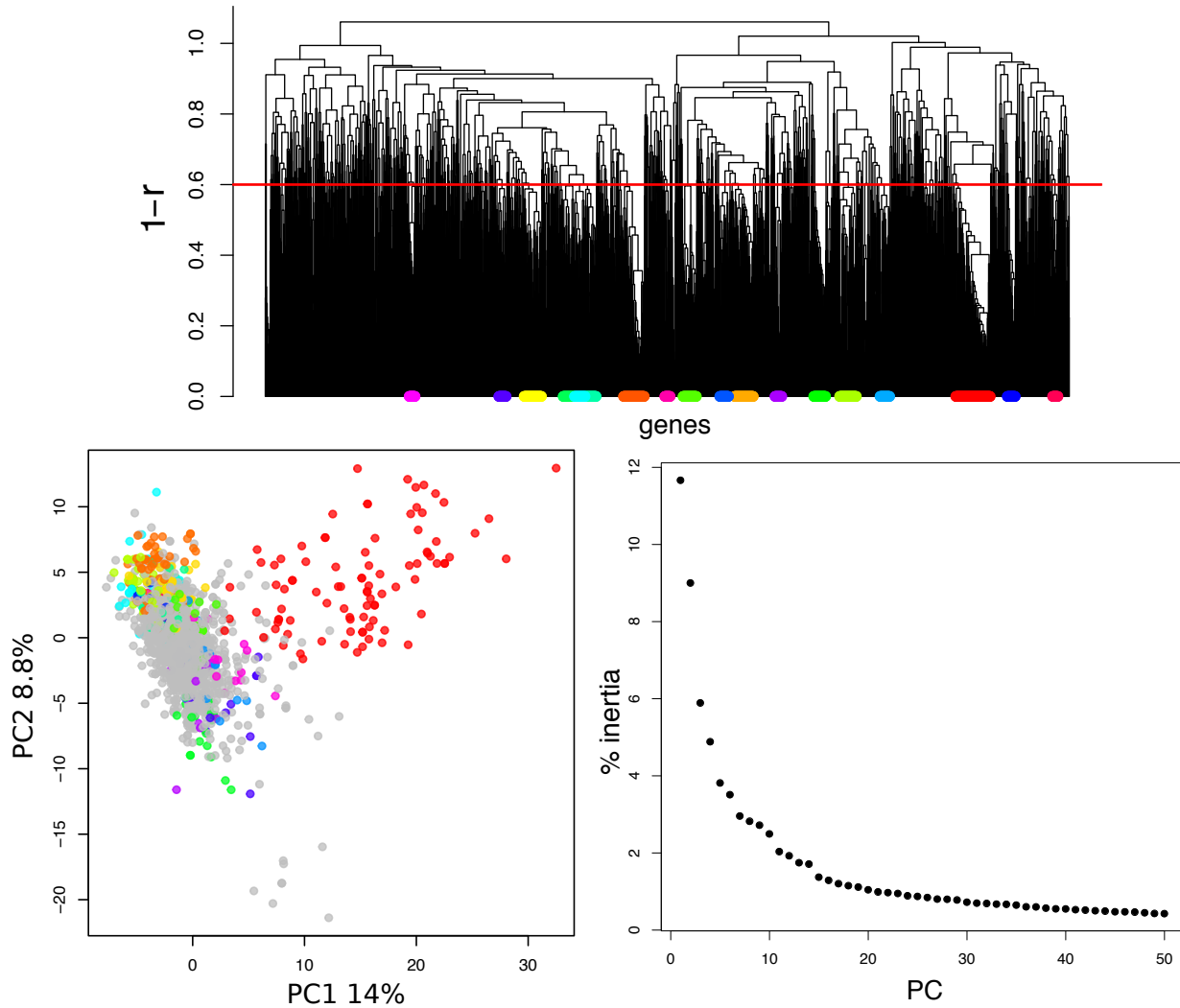


Figure 4.3: On the top of the figure is a clustering tree of the whole set genes, before removing genes with many missing values or those which do not match with the list of TSSs (see subsection 4.1.2), the red line represents the correlation value at which we cut to define clusters containing 20 or more genes (colored leaves). The figure on the bottom left shows the coordinates of the genes on the first two axes of the Principle Component Analysis. The red dots on this figure correspond to the big red cluster at the bottom of the tree. The figure on the bottom right shows the decreasing percentage of the variance that is captured by each component of the PCA, this figure can help us to decide how many components to keep according to how much variance a new component adds to the overall captured variance.

4.2.3 Independent Component Analysis (ICA)

Independent Component Analysis (ICA) is another matrix factorization method for data dimension reduction. ICA defines a new coordinate system in the multi-dimensional space such that the distributions of the data point projections on the new axes become as mutually independent as possible (Hyvärinen and Oja, 2000). This independence between axes can be achieved either by minimizing the mutual information or by maximizing the non-Gaussianity. The recognized good performances of ICA to address source separation problems, for instance in acoustic signals, has attracted the attention for blind separation of biological, environmental and technical factors affecting gene expression. The first work to apply ICA to define linear modes of genes expression was done by Liebermeister (2002). Some authors have claimed ICA to be more biologically relevant than other dimension reduction techniques, including PCA, in the context of expression data analysis (Carpentier et al., 2004).

In practice, ICA makes a decomposition analogous to PCA (see subsection 4.2.2), assuming that the data has been generated from independent K sources. It is usually written

$$Y = SA + \epsilon, \quad (4.5)$$

where Y has dimensions $(1, 512 * 165)$, S is the source matrix $(1, 512 * K)$, A is the mixing matrix $(K * 165)$, and ϵ represents the residuals. The goal of ICA is to find the source matrix S with number of columns K which correspond to number of sources that has generated this data and a mixing matrix A with number of columns equals to 165.

Despite the similarity between equations (4.4) and (4.5), applying ICA differs from PCA in two practical aspects. First, ICA requires defining the number of independent components before applying the algorithm. Second, the output components of ICA algorithms are not stable since they rely on local optimization from a random starting point, which suggests to run the algorithm multiple times and applying some stability analysis in order to identify the stable components (Kairov et al., 2017).

In order to determine the optimal number of independent components we conducted some analysis to understand the relation between selected number of components and both mean square error and stability of components. We started by calculating the mean square error between the input matrix Y and the matrix that results from multiplying the source matrix S and the mixing matrix A . In practice, we applied the "fastICA" algorithm (Hyvarinen, 1999) for every possible value of K (from 1 to 165). Figure 4.4 shows the

relationship between the mean square error and number of columns of the source matrix. When calculating the mean square error for ICA we ended up having exactly the same values as for PCA, which is due to the first step of the algorithm FastICA that consists of applying PCA to reduce the dimensionality of the data to the number of components selected by the user.

To assess the relationship between the number of selected components (K) and their stability we applied some analysis for different values of K . For every $K \in (5, 10, \dots, 120)$ we ran the FastICA 100 times and clustered the output $100 * K$ components using average-link clustering based on pairwise correlation ($1 - |r|$, the absolute value account here for the arbitrary sign of the ICA components) computed on gene-specific coefficients found in the columns of matrix S . The clustering tree was then cut at a Pearson correlation value of 0.8 to define clusters of similar components. Figure 4.5 presents the number and the ratio of clusters containing a number of components greater than or equals to 80, thereby defining "stable" components found in at least 80% of the runs. The figure shows that the number of stable components increases steadily until $K = 40$ where it starts to plateau. We had the goal of selecting a K which is as a good trade off between stability and mean square error and $K = 40$ seemed to satisfy this goal.

To produce the input covariates for our motif discovery tool, we ran the FastICA 100 times setting $K = 40$, the resulted $(40 * 100)$ components were then clustered by applying the same technique as used in the stability analysis. Cutting the tree at 0.8 lead to 26 clusters with more than 80 components. We kept a single representative component for each of these 26 clusters. To select a representative component, we gave a score for every component inside the cluster which reflects the correlation with the rest of the components in the cluster. The representative component was the one that has maximum correlation score with the rest of the cluster members. Mathematically, the selected component c^* satisfies the equation

$$c^* = \arg \max_c \sum_{c'} |r_{c,c'}|, \quad (4.6)$$

where $r_{c,c'}$ is the pairwise correlation between the two components, and c' is an index for components belonging to the cluster of interest except component c .

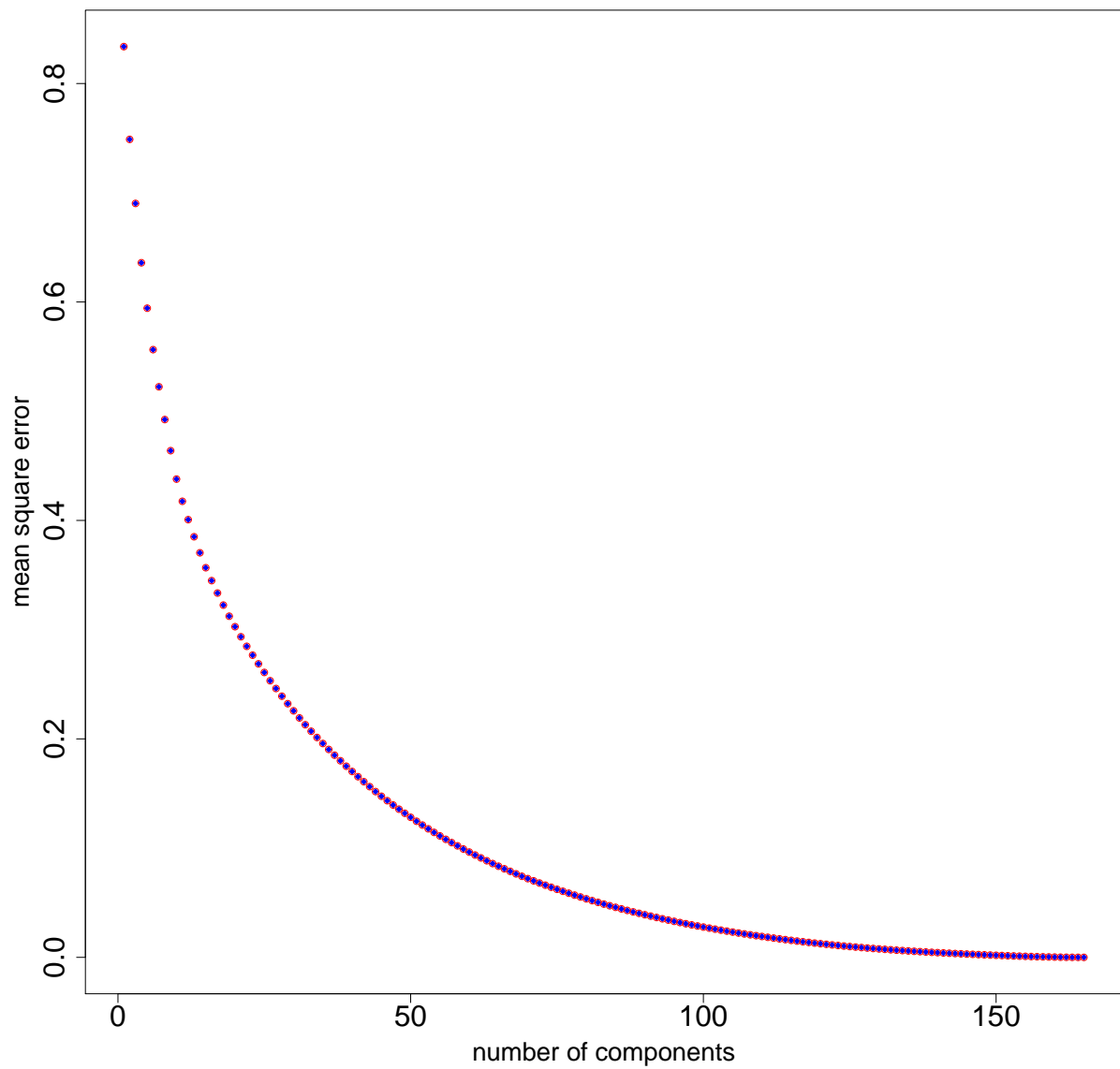


Figure 4.4: Mean square error between the original expression matrix and the approximated matrix using ICA and PCA for different values of K . The figure shows the evolution of the mean square error in relation to selected number of components (K). Results are similar for PCA and ICA (see subsection 4.2.3).

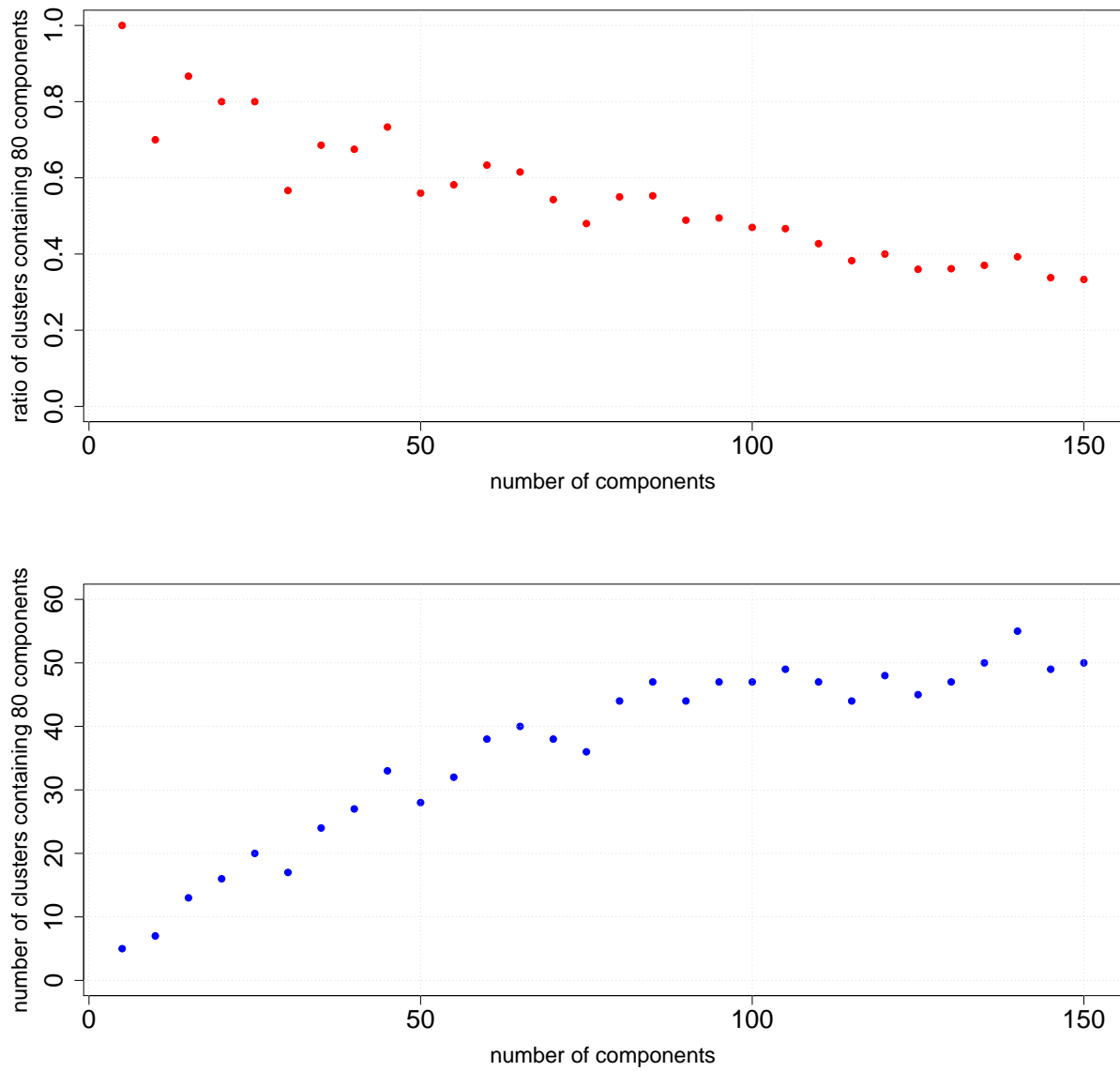


Figure 4.5: Relationship between component stability and selected number of components in ICA. For every $K \in (5, 10, \dots, 120)$ we ran the FastICA 100 times. The resulting $100 \cdot K$ components of each K were clustered together, the dots in the figures correspond to ratio (upper) and number (lower) of clusters containing at least 80 components (i.e. the number of stable components).

4.3 Prior distribution and parameter tuning

The Bayesian framework allows a “learning” capability so that “historical” data can be used in modeling a new, but similar problem. This can be important in building prior distributions based on partial information for known motifs and improving estimation of novel motifs. When this information is not available, it is intuitive to select one form of uninformative prior. To avoid choosing uninformative prior that may cause the algorithm to misbehave in the process of parameters estimation, including the uniform prior in some cases, we analyzed the impact of the prior for some of the model parameters.

The first prior that we examined was the prior distribution for the number of regions modeling the pdf of the motif occurrence positions in terms of piece-wise constant function (hereafter referred to as position pdf). We tested two prior distributions (uniform and geometric) as illustrated in figure 4.6. In order to analyze the impact of the prior on the posterior distribution, we examined three very different scenarios for the distribution of the motif occurrences in the sequences. These scenarios are: uniformly distributed, tend to localize in a specific region of the sequence (piece-wise constant pdf), or occur with a probability that varies linearly with the position in the sequence (see figure 4.6).

In practice, we simulated variable numbers of motif occurrences according to the three scenarios, and implemented the relevant part of the algorithm (updating iteratively only the set of parameters related to estimating the number of regions, considering the motif occurrences as given). Figure 4.7 shows under the two priors, and for different numbers of motifs, how accurately the model allows to estimate the position pdf. We analyzed as well the posterior distribution of the number of regions. Figure 4.8 shows the posterior of the estimated number of regions in the piece-wise case.

We verified (unsurprisingly) that the accuracy of our estimated position pdfs is improved and the impact of the prior choice decreases as the sample size increases. For small number of occurrences, we noticed that the geometric prior provided smoother position pdf than the uniform prior (Figure 4.6), which can be explained by the fact that the geometric prior favors smaller number of regions than the uniform prior. Our main conclusion is that the choice of geometric prior on the number of regions is more relevant in our case because it leads to smoother estimated posteriors for the position pdf.

Another main prior to select was the prior put on the PWMs, $\theta_{m,w}$ representing the composition of motif m at position w . We have chosen the Dirichlet distribution as a prior on $\theta_{m,w}$ since it is the conjugate prior of the multinomial distribution (see section 2.3.2).

Yet, in practice we need to choose the concentration parameters γ of this Dirichlet. An usual choice for γ in the case for the four DNA nucleotides is the uniform prior given by $(1, 1, 1, 1)$. We also tested the non-informative (objective) prior distribution referred to as Jeffreys prior. Jeffreys prior in our case is $(0.25, 0.25, 0.25, 0.25)$. When fixing the value of γ to 0.25, we noticed that the produced results were substantially better. We interpreted this as a consequence of focusing the search on motifs with higher information content (see figure 4.9).

The prior set on the number of occurrence of each motif was also found to have a significant impact on the behavior of the algorithm. If we are not taking expression data into account, this prior value is represented by the two parameters of a Beta distribution. These two parameters (α_1, α_2) incorporate our prior belief about the ratio $(\frac{\alpha_1}{\alpha_2})$ between the number of sequences containing the motif and those not containing the motif. At the beginning we set $(\frac{\alpha_1}{\alpha_2})$ to 1 ($\alpha_1 = 1, \alpha_2 = 1$), this choice caused very slow mixing of the algorithm which limited the exploration of the search space and caused the algorithm to get stuck on motifs with high number of occurrences and low information content. To overcome this slow mixing and to allow the algorithm to discover motifs with high information content, we made the choice of reducing this ratio to a much smaller value, such as 0.01 ($\alpha_1 = 1, \alpha_2 = 100$). When we are incorporating expression data, the role of the aforementioned ratio is played by β_0 , which is the intercept of the Probit regression model, linking motif occurrences to expression data.

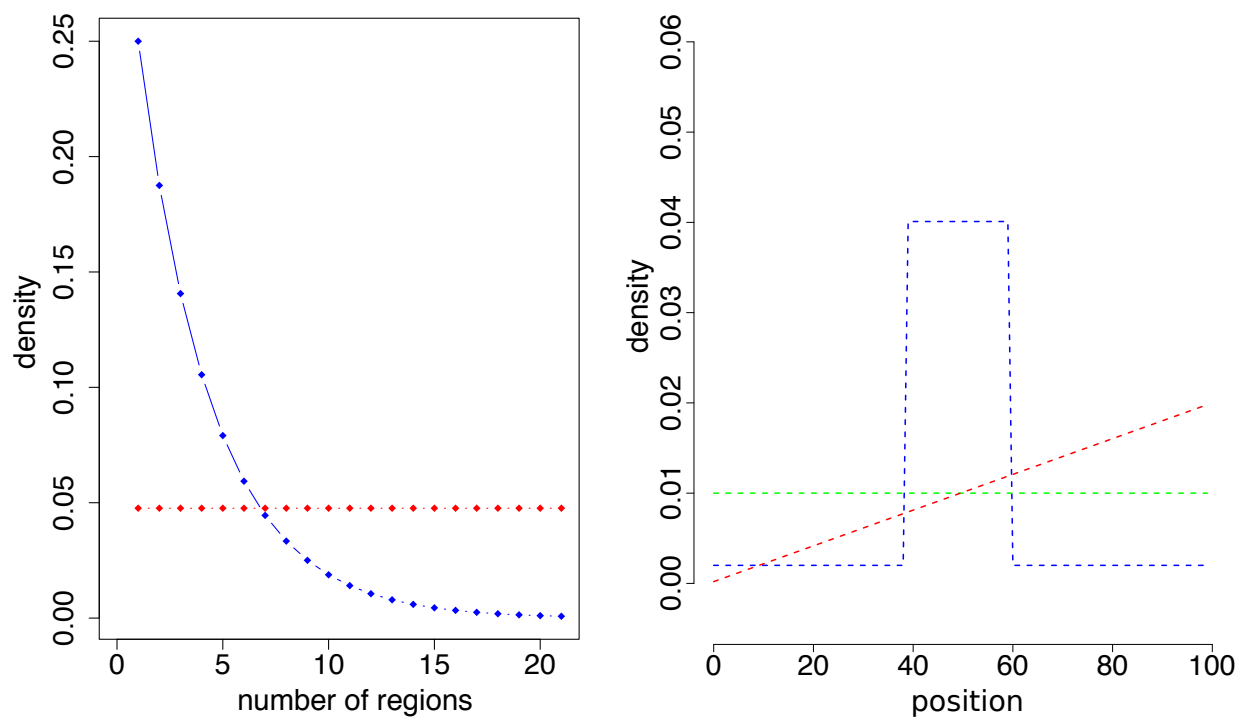


Figure 4.6: Set up of the simulation study to explore the impact of the prior on the number of regions. The sub-figure on the left shows the two prior distributions that were tested, $Uniform(\{1, \dots, 21\})$ and $Geom(mean = 4)$. The sub-figure on the right shows the three scenarios for pdfs of motif occurrence posteriors.

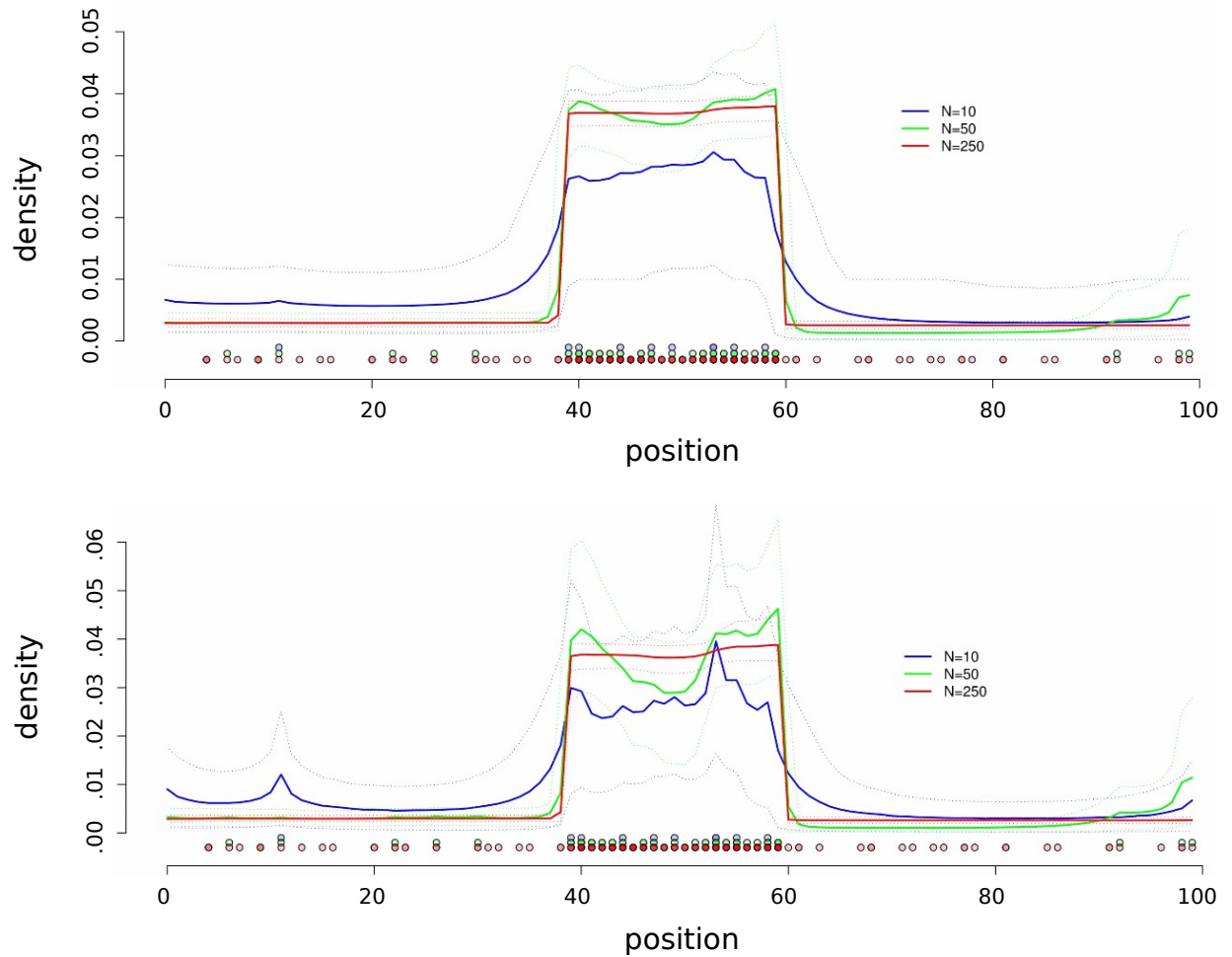


Figure 4.7: Impact of the prior on the estimation of pdf for motif occurrence posteriors. Only the scenario that correspond to piece-wise constant pdf is represented here. The different colors correspond to estimates obtained for different sample sizes (blue for 10, green for 50, and red for 250), with a uniform prior for the number of regions (lower), and geometric prior (upper).

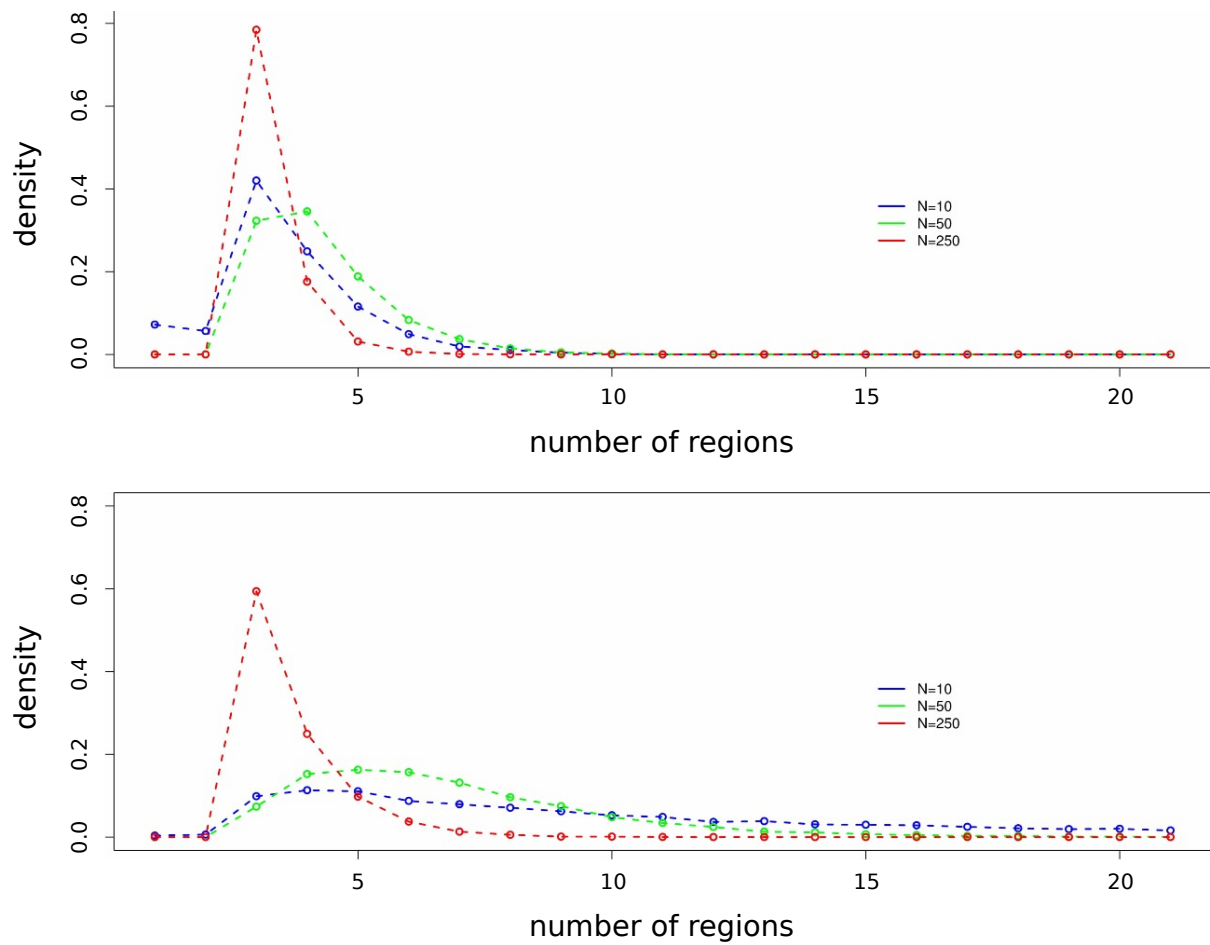


Figure 4.8: Posterior probability on K if the prior on motif positions is set according to a piecewise constant function, with a uniform prior for the number of regions (lower), and geometric prior (upper).

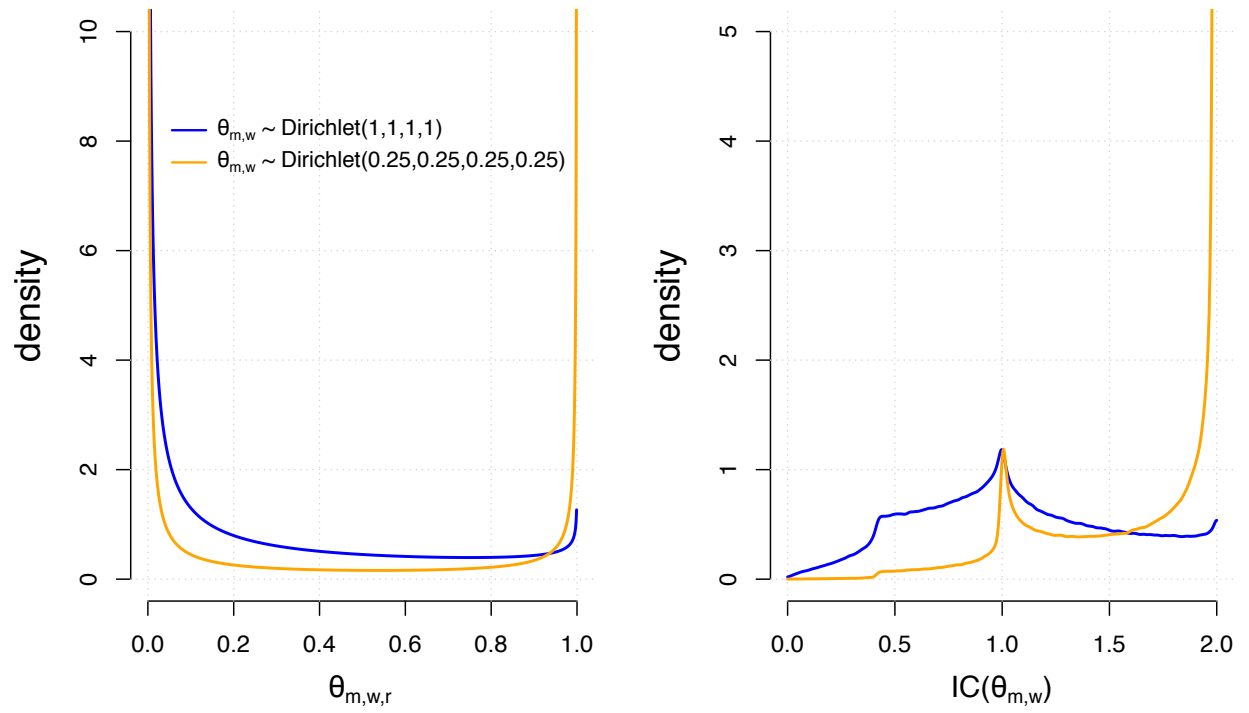


Figure 4.9: Impact of the choice of the concentration parameter of the Dirichlet prior distribution for PWMs. The sub-figure on the left shows the pdf of $\theta_{m,w,r}$ in the two cases of $\gamma = 1$ and $\gamma = 0.25$. The sub-figure on the right shows the resulting pdf of the information content (IC) of $\theta_{m,w}$ for the two choices of γ .

4.4 The developed tool

The methodology was implemented in C++ programming language. The C++ source code is available online at <https://github.com/eisultan/Multiple>. It can be downloaded and reused under GNU General Public License.

Our developed tool for finding TFBSs is called Multiple. Multiple is a program that searches for TFBSs. It requires as input the sequence data (DNA promoter regions around the Transcription Start Sites (TSSs)) and the expression data (the expression profiles across the conditions). The user have the choice to input the expression data either in its original form or by first summarizing it into covariates. Each covariate may correspond to the coordinate of the gene on an axis (e.g. obtained by PCA or ICA) or to its position in a tree (e.g. obtained by hierarchical clustering).

4.4.1 Description of the command line arguments

Usage : multiple [-seq | [-xvectors | [-xtrees | [-prefix | [-nmotif <#(default 2)>] [-wmin <#(default 3)>] [-wmax <#(default 25)>] [-breakpmax <#(default 20)>] [-Nsweep <#(default 10000)>] [-rngseed <#(default 2)>] [-ICbg <bool(default false)>] [-widthbg <#(default wmin+3)> <double(default 1)>] [-validate <bool(default false)>] [-TSS <bool(default true)>] [-geo_prior_breakpoints <double(default 0.25)>] [-z <double(default 1)> <double(default 1)> <double(default 1)>] [-probit <double(default 1)> <double(default 0.2)> <double(default -2.3)>] [-palprior <double(default 0)> <double(default 0.5)> <#(default 1)> <double(default 0.5)>]

Main:

-seq <file> : the name of the sequence file, formatted as shown below.

Optional:

- -xvectors <file> : name of the expression file that contains the PCA/ICA covariates.
- -xtree <file> : name of the expression file that contains the clustering tree, see format below.
- prefix <string> : the prefix to be added before every file name, the prefix allows the user to run many jobs with different seeds or with different set of parameters without having to change the directory by changing the prefix before starting the job.

- -nmotif <#(default 2)> : number of motifs.
- -wmin <#(default 3)> : minimal allowed width for the motifs.
- -wmax <#(default 25)> : maximum allowed width for the motifs.
- -breakpmax <#(default 20)> : we divide the promoter regions into subregions with different probability of finding the algorithm, this parameter will set an upper bound for number of sub-regions.
- -Nsweep <#(default 10000)> : number of MCMC iterations.
- -rngseed <#(default 2)> : this parameter gives the freedom to change the seed which is useful in case of performing stability analysis.
- -ICbg <bool(default false)> : In the case of overlap between two or more motifs, there is a parameter that assign a membership of this nucleotide to one of the intersecting motifs. By adjusting this parameter the user indicates whether the algorithm will use the information content of the intersecting motifs for this assignment, or just assume an equal probability for the nucleotide to be generated by any of the motifs.
- -widthbg <#(default wmin+3)> <double(default 1)> : prior for the width of the motif, the motif will have piece-wise distribution the first part of the motif will have a uniform distribution and the second part will have a geometric distribution. The user will enter two values, the first will corresponds the width of the first part with uniform distribution and the second value will be the parameter of the geometric distribution. For example, if the user entered 8 and 0.9 this will mean the motif will have a uniform probability to have a width between wmin and 8, and will be less likely to extend after 8 and this likelihood for extension will be controlled by the 0.9 value.
- -validate <bool(default false)>] : this parameter may not be useful for the user, it was used in the building of the code to validate that it is working as supposed to.
- -TSS <bool(default true)> : this parameter if deactivated, the algorithm will not divide the sequence into sub-sequences with different probability of motif occurrence, but rather will treat the whole sequence as one piece with uniform probability of seeing a motif.

- `-geo_prior_breakpoints <double(default 0.25)>` : parameter of the geometric distribution to be put over the number of break points that divide the promoter region between 0 (the sequence as one piece) and `breakpmax`. You can adjust this value to 1 to have a uniform distribution.
- `-z <double(default 1)> <double(default 1)> <double(default 1)>` : This is to adjust the pseudo count of the of three Dirichlet distribution used in the algorithm. The first is for the motif nucleotides' count, the second is for the background nucleotides' count, and the third is for motif counts in the subregions.
- `-probit <double(default 1)> <double(default 0.2)> <double(default -2.3)>` : adjusting the values for the regression model (probit) used to summarize the expression values into probabilities of occurrence. The first is for the probit model variance, the second and the third are for for the variance and the mean values of the intercept respectively. The mean of the intercept may thought of as the prior value on the number of motifs without looking at the expression data, while the variance of the intercept may thought of as how much we are expecting this value to vary from the initial value. We can think of the variance of the probit model as how much we want the shape of the model to change.
- `-palprior <double(default 0)> <double(default 0.5)> <#(default 1)> <double(default 0.5)>` : In the case that the user is interested specifically in the palindromic motif, where the four parameters are respectively: probability for a motif to be "palindrome", probability for a pair of columns in a palindrome to be coupled, minimum length for a half palindrome, a parameter that account for the centering of the palindrome (1 means no centering, all positions are equally likely for the center).

The sequence file should be formatted as:

```
1512 121
>lmo0001
TTAACTGGCTGTGGACAACCGTTTTTCACATCTGGACAGTTTTGTGGATAGA
>lmo0002
GCAGCATGGCTTGTAACCTACTTATCCACAAATCCACAGCGCCTATTACTATT
...
```

where 1512 specifies the number of sequences to be analyzed and 121 indicates the nucleotide length of each sequence. The subsequent lines provide the sequence information:

they are composed of a sequence identifier separated from the sequence itself by a tabulation character.

Notice: all the sequences need to have the same length and that missing data is not allowed.

The topology and branch length information about the tree should be provided in a format as shown below:

```

1
-20      -1231    0.0025
...
-982     6        0.0038
...
353      434      0.0155
...
1509     1510     0.5587

```

The first row has only the number of the inputted trees. The rows afterwards corresponds to nodes, such as, the second row corresponds to the first node and so on. The first two columns indicate the two children of this node and the last column gives the height of the node. The n leaves (here 1512) are numbered negatively from -1 to - n , the $n-1$ internal nodes are numbered positively from 1 to $n-1$. The nodes need to be ordered by increasing height such that all the descendants of a node are found above the line that describes the node.

Notice: this input format supposes an ultra-metric binary tree as all the leaves are implicitly assumed to have height 0. These requirements are naturally fulfilled if the tree was obtained by hierarchical clustering of the matrix of pair-wise correlation coefficients using the average linkage algorithm. Modifying the program to allow other types of tree should not be too difficult.

4.4.2 Output files

The standard output includes the user-specified parameters that were used.

The parameters file:

The parameter file is edited every ten sweeps by adding the new values for these parameters respectively:

- sweep: the number of the sweep in which the file is being edited.
- motif: the index of the motif whose its value are written in the row.
- nb_regions: how many subregions correspond to the given motif.
- phi: in the case of not inputting expression data phi will indicate the probability of seeing the motif in the input data.
- refpos: the reference position, since the position of the motif inside the sequence does not refer to the first position but rather to a reference position and by turn the reference position is need in order to align the motif occurrence. This kind of referring to the motif gives the freedom to the motif to go outside the sequence.
- width: the width of the motif at the given sweep.
- bg_order: the order Markov model describing the background
- track_left: this value is increased or decreased by 1 according to the change happening to the most left nucleotide of the motif, this value combined with the width allows us to imagine how the motif moved over the sequence across the iterations which is good if we want to align the PWMs and also to access the convergence of the motif.
- prob_x: This value is basically the multiplication of the probabilities of the nucleotides in the data. If we initialize all the PWM for all the motifs and the parameters for the background to 0.25 which means equal probabilities to see the different nucleotides (A,C,G,T) inside the motifs as well as in the background. The prob_x value will equal to $(\log(0.25) * \text{number of sequences} * \text{number of nucleotides in the sequence})$ This value should improve across the iterations since the estimation of the PWMs is improving and by turn the information content is improving. This value will give us an indication on how fast this improvement is happening across the iterations.
- T_probit: The next K columns in the parameter files are titles T_probit corresponding to K-1 covariate and the intercept of the regression mode. The cells of the T_probit columns take values between 0 and 3, where 0 means that the corresponding covariate is inactive, 1 is given if the corresponding covariate represent a

PCA/ICA axis and the inputs in this axis should be treat as they are without binarizing them to only 2 values (having a motif or not) instead of degrees of believe. 2 As well is for the PCA/ICA axis but in case that the corresponding component will be binarized. 3 is given if the corresponding component is representing the height of the gene on a tree.

- beta: After the K T_probit column, the file contains K beta columns where the cells of these columns contain the value that the covariate is influencing the regression model final value (the value that tells if the gene has the motif or not).

The PWM file:

The PWM file is edited every ten sweeps by adding the new values for the position weight Matrices describing the motifs and some other parameters related to the motif shape, these parameters are respectively:

- sweep: the number of the sweep in which the file is being edited.
- motif: the index of the motif whose values are written in the row.
- width: the width of the motif in concern.
- refpos: the reference position (explained above).
- centerpal: tells which nucleotide within the motif represents the center of the palindrome (if there is any).
- npairedcolpal: the number of paired columns for the current motif in the current iteration (even number).
- prob: after that, there are ($\text{maxw} * 4$) columns describing the probability of seeing (A,C,G,T) respectively. For example, the first four columns are the probability to see (A,C,G,T) in the first position and so on.

The position pdf file:

The position pdf file is edited every 100 sweeps and it contains a column for every nucleotide in the sequence, this column represents the probability of seeing the motif (the reference position) at this nucleotide. There are four other columns in the file, these columns represent respectively:

- sweep: the number of the sweep in which the file is being edited.
- motif: the index of the motif whose values are written in the row.
- phi: explained above.
- nb_regions: represent the amount of sub regions that the sequence has been divided into, each sub region has a different probability of seeing the motif.
- pos_prob: after that, there are L columns corresponding to L nucleotides forming the sequence. The values in the cells of these columns are the probability of seeing the motif (the reference position) at this nucleotide.

The motif sequence file:

The motif sequence file is edited every 100 sweeps and it contains the positions of the motif occurrences in all the sequences. There are another two columns describing the sweep and the motif as before. The columns in the file are ordered in the following manner:

- sweep: the number of the sweep in which the file is being edited.
- motif: the index of the motif whose values are written in the row.
- position: After that, there are N columns called position for the N sequences in the input data. The values in these columns are the the position of the motif in concern in the corresponding sequence. When the motif is absent, the value (-1) is given.

Chapter 5

Results

This chapter is dedicated to the results and performance of our developed tool. In the first section, I show how we process the output files in order to detect the stable motifs. The second section is dedicated to results on the bacterium *L. monocytogenes* where I discuss the discovered motifs and their links to known regulons. In the third section, I review some analysis performed by us to assess the quality of results in relation to the type of information incorporated in the search process (TSS position data, expression data). The fourth and final section of this chapter is dedicated to the relative performance of our tool in comparison to other popular tools.

5.1 Processing of the output files to detect stable motifs

The procedures explained in this section were developed to process the output files that resulted from applying our tool on the *L. monocytogenes* data set (presented in chapter 4) but they should be applicable to output files produced from any data set. The data set is composed of sequence data (1,512 DNA sequences, each of length 121 bp), and expression data (a matrix of 1,512 rows and 165 columns).

The output files explained in section 4.4 contain all the necessary information, not only to summarize our final findings, but also to make plots that help us to visualize and understand the algorithm behavior. For instance, to produce the final PWM that is used to plot the sequence logo, we needed to align the 100 matrices produced in the last 10,000 sweeps (thinning step = 100). To perform this alignment we needed to record the changes that occurred on the 5'- and 3'-sides of the PWM. The width of the motif aside with a variable called "track_left" provide us with these information. The value of the variable

"track_left" is incremented by 1 each time the motif is shifted to the right or its width is reduced from left, and decremented by 1 each time the motif width is shifted to the left or its width extended from left. Figure 5.1 provides a graphical representation of this alignment procedure, where changes that occurred on the 5'- and 3'-sides of the PWM are represented respectively by the lower and higher borders of the polygon.

The total number of PWM columns covered by motif m throughout the last 10,000 sweeps (referred to as K) is given by

$$\max_k(track_left_m^{(k)} + w_m^{(k)}) - \min_k(track_left_m^{(k)}). \quad (5.1)$$

Figure 5.1 demonstrates that some components after discovering a motif and staying stable for a number of sweeps disappear and converge to another motif, while some other components have not converged to a specific motif by the time that maximum number of sweeps is reached. To cope with this randomness, we decided to run the MCMC algorithm multiple times with a different random seed for each run, then cluster the identified motifs and keep only those that were found in more than one MCMC run.

We performed 10 independent parallel runs of our algorithm for *de novo* motif discovery, each run consisting of 50,000 MCMC sweeps. Only the last 10,000 sweeps were used for our analysis to estimate posterior distributions while the first 40,000 sweeps served as a burn-in period to forget the starting point. The number of components corresponding to possible motifs \mathcal{M} in each run was fixed to 75 (however the prior gives a significant probability for each motif to be empty). These runs produced information on 750 (10x75) possible motifs that differed widely: with or without occurrences, stable across the last 10,000 sweeps or not, found in several independent runs or not. We analyzed and compared these components to extract distinct well supported motifs (stable across the last 10,000 sweeps and found in at least two runs). For this purpose, we identified the positions in each sequence that are predicted to be covered by each motif with estimated posterior probability at least 0.5 and computed pairwise distances between motifs i and j as

$$d_{mean}(i, j) = 1 - \frac{2O(i, j)}{N(i) + N(j)}, \quad (5.2)$$

where $N(i)$ and $N(j)$ denote the total number of positions covered by each motif and $O(i, j)$ is the number of positions covered by both motifs. Mathematically, the total number of positions covered by motif i with posterior probability at least 0.5 is defined by can be written as

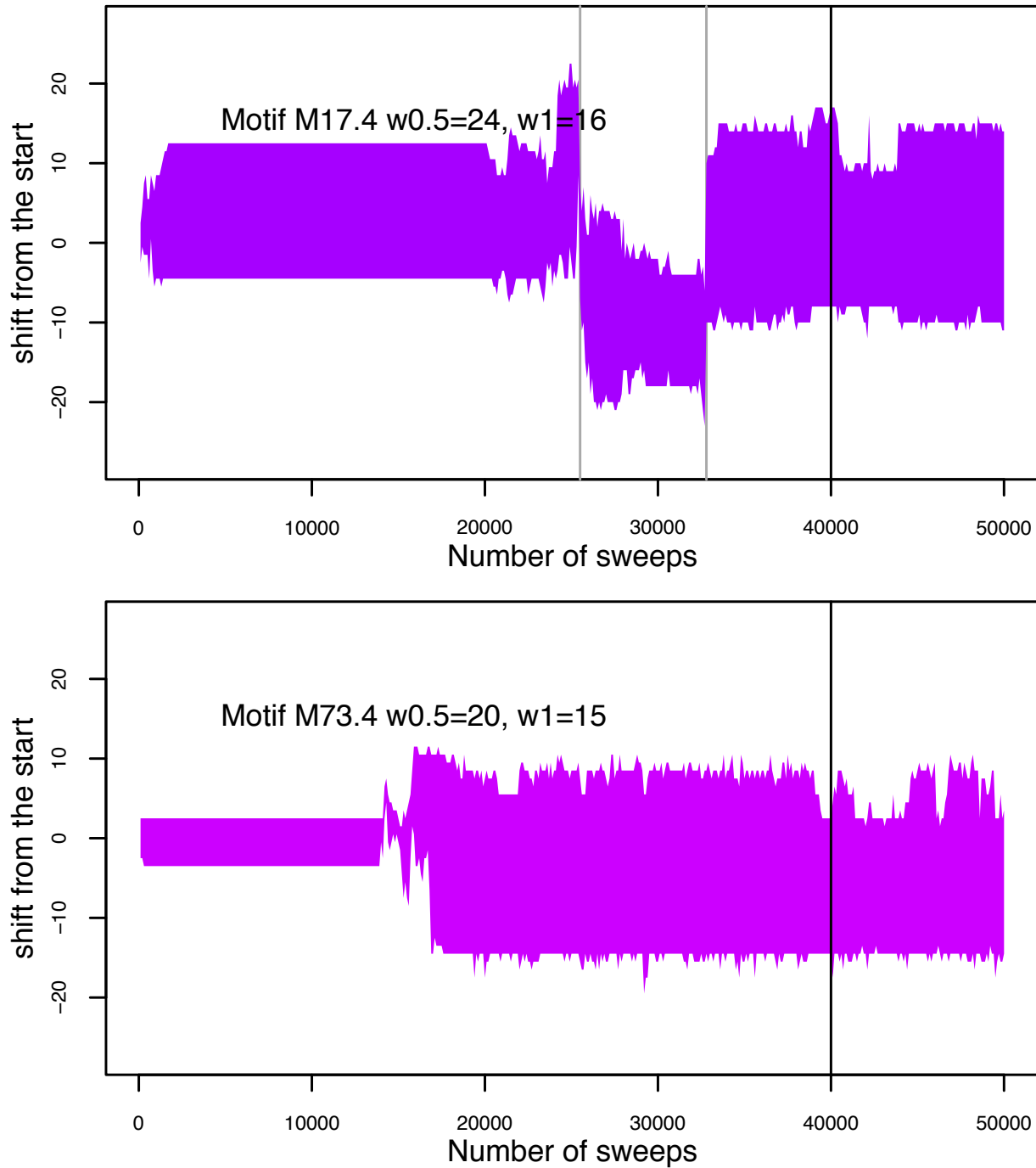


Figure 5.1: Two different convergence plots that show the changes that occurred in the widths of the corresponding PWMs across the 50,000 sweeps of the algorithm. Changes that occurred on the 5'- and 3'-sides of the PWM are represented respectively by the lower and higher borders of the polygon. A vertical line is drawn at sweep number 40,000 that delimited the end of our burn-in period. For the sake for better presentation, the polygon is centered in the figure whenever its center become greater than 15 or less than -15 and a dark grey vertical line is drawn to indicate the specific iteration in which this happened.

$$N(i) = \sum_n \sum_l \mathbb{I}\{\hat{p}_{i,n,l} \geq 0.5\}, \quad (5.3)$$

where,

$$\hat{p}_{i,n,l} = \frac{1}{K} \sum_k \mathbb{I}\{a_{i,n}^{(k)} - r_i^{(k)} \leq l < a_{i,n}^{(k)} - r_i^{(k)} + w_i^{(k)}\}, \quad (5.4)$$

where K is the number of iterations used for the analysis (10,000 in this case), and $\hat{p}_{i,n,l}$ is the estimated posterior probability for position n, l to be covered by motif m . Similarly,

$$O(i, j) = \sum_n \sum_l \mathbb{I}\{\hat{p}_{i,n,l} \geq 0.5\} \mathbb{I}\{\hat{p}_{j,n,l} \geq 0.5\}. \quad (5.5)$$

With these formula, $d(i, j) = 0$ if the positions covered are exactly the same and $d(i, j) = 1$ if they do not overlap (including if one motif has no occurrence). Motifs were then compared using hierarchical clustering based on this distance and the average-link method. To select only well distinct motifs found at least in two runs, we used procedure relying on two levels of clustering obtained by cutting the tree at two different heights: 0.75 for high-level clusters, 0.25 for low-level clusters. The high-level clusters serve to ensure that final stable motifs are not redundant. For each high-level cluster we examined if it contained a low-level clusters of size at least two and selected in these low-level clusters a single representative motif ($\arg \max_i N(i)$) per high-level cluster (when no low-level cluster existed, the high-level cluster was not represented in our final list of motifs). Figure 5.2 shows a slice of the clustering tree along with, the number of occurrences, and the width for each motif. The output result of these selection criteria is presented in section 5.2

Although the distance function of equation (5.2) was used to calculate the distance between motifs, it is worth mentioning that we have tried alternative distance functions before finally selecting this one. The other functions that were tested are:

$$d_{frac}(i, j) = 1 - \frac{O(i, j)}{N(i) + N(j) - O(i, j)} \quad (5.6)$$

$$d_{max}(i, j) = 1 - \frac{O(i, j)}{\max(N(i), N(j))} \quad (5.7)$$

$$d_{min}(i, j) = 1 - \frac{O(i, j)}{\min(N(i), N(j))} \quad (5.8)$$

$$d_{Euclidean}(i, j) = \sqrt{N(i) + N(j) - 2O(i, j)} \quad (5.9)$$

We chose the d_{mean} as our final distance function since its behavior was the easiest to interpret. For instance, using d_{min} tends to cluster two motifs together if one of them represent a part of the second. On the contrary, d_{max} will tend to distinguish such motifs. The distance $d_{Euclidean}$ does not give results easily understandable between motifs of different lengths and different numbers of occurrences. Another approach that we decided to avoid is to compare the PWMs.

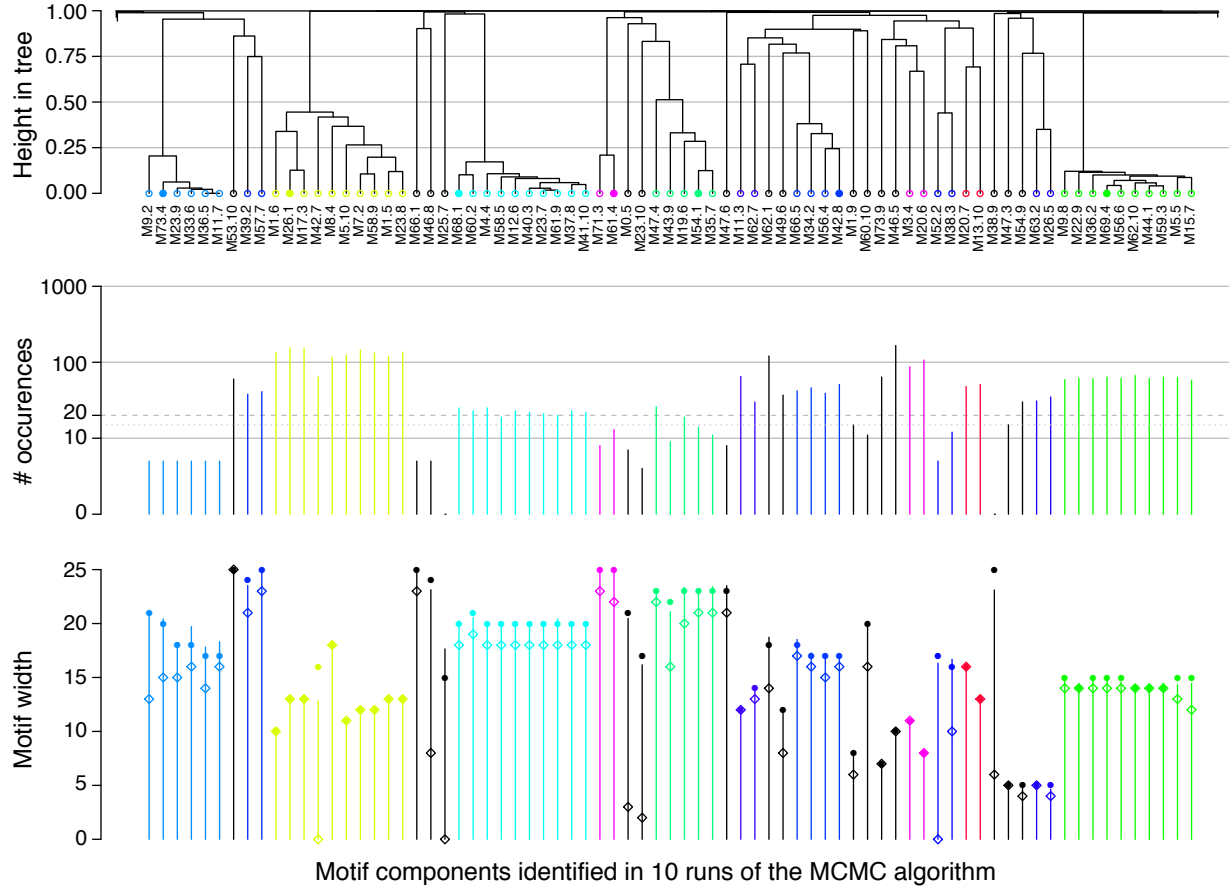


Figure 5.2: Hierarchical clustering of motif components to extract stable motifs. Only a fraction of the tree is represented here (75 motif components out of 750). First row: hierarchical clustering tree. Colors distinguish the high-level clusters obtained by cutting the tree at height 0.75. The selected representative stable motifs are indicated by closed circles (open circles for non selected motifs). Second row: number of occurrences of each motif components across the 1,512 promoter sequences (posterior probability cut-off 0.5). Third row: motif width as obtained by three different approaches: average motif width across the last 10,000 MCMC sweeps (vertical bar), number of columns included in the PWM with probability cut-off 0.5 (closed circle) or cut-off 1.0 (open losange).

5.2 Results on *L. monocytogenes*

5.2.1 Main characteristics of the sequence motifs and the corresponding predicted regulons

The procedure described in subsection 5.1 identified 40 stable motifs. Table 5.1 summarizes the main characteristics of the 40 motifs which were found to exhibit considerable diversity in terms of abundance, preferred position with respect to the TSS, link with expression data, and PWM structure (palindromic vs. non-palindromic). Figure 5.3 illustrates 3 of these motifs. Similar figures for all motifs, as well, the list of genes associated with each of the 40 motifs are made accessible to the community of biologist working on *L. monocytogenes*. The two files are supplementary file S1 and supplementary file S2, respectively.

5.2.2 Links to known transcription factors

Three complementary approaches were used to identify links between the 40 motifs discovered by our *de novo* approach and known regulons. The first was the comparison with lists of genes collected from tables published in several expression studies and positions of transcription factor binding sites recorded in the RegPrecise database; the second approach was the systematic comparison with 188 reference PWMs derived from sequence alignments extracted from the propagated RegPrecise database (Novichkov et al., 2013) for different taxonomic groups in the Firmicutes phylum: *Listeriaceae* (25 PWMs) and *Staphylococcaceae* (39 PWMs) and *Bacillales* (124 PWMs).

The third approach consisted in specific literature searches associated with a careful manual examination of (i) the set of genes downstream of TSSs in which the motif was predicted to occur (ii) the comparisons of conditions in which the log2 fold-change deviated the most from 0 (iii) the characteristics of the PWM and the preferred position of motif occurrences with respect to the TSS. For many of the identified transcription factors, these lines of observation provided several convergent clues. The links to known regulons that were identified with this combination of approaches are reported in the rightmost column of Table 5.1. Most of the motifs with high number of occurrence were found to describe general characteristics of promoter regions (variations on the themes of SigA -10 and -35 boxes, nucleotide composition around TSS, Ribosome Binding Site). Systematic comparison with RegPrecise PWMs was particularly informative for the identification of BglR2 CcpA, CcpB, Fur, LexA, LiaR, and Rex. Among the other identified transcription

factors, SigB and SigL were identified based on position with respect to the TSS.

Comparison with sigma factor consensus defined for *B. subtilis* (Nicolas et al., 2012), as well as overlap with previous experimental data on SigL (also known as RpoN or sigma54) and SigB regulons in *L. monocytogenes* (Arous et al., 2004; Chaturongakul et al., 2011; Palmer et al., 2011). Spx was identified based on (i) its position with respect to the TSS just upstream of the SigA -35 box, (ii) sequence properties reported for Spx in *B. subtilis* described as an AGCA element at position -44 (Rochat et al., 2012; Lin et al., 2013), and (iii) literature data on the Spx-regulon of *L. monocytogenes* (Whiteley et al., 2017). PWMs found for PrfA and VirR, two key transcription regulator involved in *L. monocytogenes* virulence, were in line with previously described sequence properties (Mandin et al., 2005; Scortti et al., 2007; de las Heras et al., 2011). The correspondences that we identified with known motifs validate our approach for *de novo* discovery and suggest that other motifs that we identified may also correspond to biologically relevant motifs.

Among the predicted regulons that we have not been able to connect to literature data, the most spectacular by its size and the functional homogeneity of the regulated genes (genes encoding ribosomal proteins), is associated with motif M58.7.

Importantly, even when motifs could be linked to previously known transcription factors, being able to identify them in a *de novo* and automated manner can shed a new light on the corresponding regulons. For instance, our results suggest that LiaR regulon may be about twice larger than previously identified by differential expression analysis of the $\Delta liaS$ mutant vs. wild-type (Fritsch et al., 2011). Ordered by the number of TSSs, the predicted regulons for which we have identified a transcription factor are: SigB, CcpA, Rex, LiaR, Fur, LexA, VirR, Spx, BglR2, SigL, PrfA.

Table 5.1: Summary of the 40 stable motifs identified in *L. monocytogenes* EGD-e

motif ^a	#TSSs ^b	#TSS(0.8-0.2) ^c	w ^d	IC ^e	Pal ^f	Position ^g	FC ^h	#r ⁱ	Comment ^j
M71.2	1,325	1,174-1,435	9	3	1.3	-9 [1]	0.44	7:10	SigA -10
M71.8	1,308	848-1,506	6	2	1.6	1 [1]	0.46	13:16	TSS (A)
M36.4	1,033	552-1,382	6	3	0	-32 [2]	0.48	10:10	SigA -35 (TTG)
M55.7	836	117-1,486	5	0	1.7	-25 [40]	0.58	3:7	SigA -10 extension (CT) ?
M8.3	792	331-1,273	5	1	0	4 [1]	0.53	10:15	TSS (G)
M9.10	735	447-1,165	8	4	0.6	-13 [1]	0.50	7:8	SigA (extended -10, TG)
M61.5	561	171-1,153	5	5	0	14 [10]	0.41	6:8	RBS (GGAGG)
M59.9	297	96-844	21	7	0	-60 [42]	0.36	4:10	T-rich element
M26.1	252	107-640	13	5	2.1	-79 [23]	0.73	6:10	SigA -10 on reverse strand
M27.6	240	153-590	6	5	0.4	-33 [2]	0.48	6:6	SigA -35 (TTGAC)
M29.8	122	91-150	12	4	2.2	-13 [2]	3.06	10:10	SigB -10
M9.1	116	43-368	22	2	5.6	-75 [37]	1.18	2:8	loose
M18.2	108	47-355	6	6	0.3	9 [12]	0.52	2:8	RBS (GAGGTG)
M12.4	97	71-109	7	4	0.6	-32 [3]	3.87	10:10	SigB -35
M69.4	87	59-128	15	8	10.9	-30 [53]	1.75	10:10	CcpA (CRE-box)
M42.8	62	14-295	17	0	0.2	-86 [21]	0.61	2:4	loose
M31.4	39	16-81	23	8	18.8	-40 [61]	1.88	2:2	Rex
M68.1	31	20-47	20	5	15	-46 [25]	3.00	10:10	LiaR
M58.7	27	14-40	19	6	9.9	-47 [56]	3.23	9:10	Ribosomal protein genes
M62.3	26	18-34	20	13	16.2	-16 [67]	1.89	10:10	Fur
M38.10	22	18-38	15	8	12.1	-19 [28]	2.36	9:10	LexA
M53.9	20	8-50	20	8	15.1	-47 [60]	1.52	10:10	VirR
M2.6	19	12-53	9	4	1.1	-49 [2]	1.12	3:5	Spx
M13.7	19	7-48	21	5	1.4	-58 [69]	0.89	2:3	-
M61.6	14	9-16	25	14	16.3	-29 [3]	0.81	9:10	BglR2
M54.1	14	6-40	23	8	1	-56 [61]	1.04	2:5	-
M61.4	13	7-25	25	6	0.9	-37 [62]	1.20	2:2	-
M50.10	13	7-43	21	5	9.6	-77 [28]	0.93	2:2	-
M70.6	11	7-19	24	22	19.2	-44 [63]	1.28	10:10	-
M3.1	9	4-20	21	9	15.5	-43 [62]	1.52	10:10	-
M33.8	8	7-13	22	11	4.2	-24 [23]	2.78	9:10	SigL
M49.4	7	6-11	23	12	16.8	-35 [19]	10.55	10:10	PrfA
M20.1	7	5-7	25	16	5.4	-55 [3]	6.64	10:10	-
M17.4	7	5-14	24	9	14.8	-39 [61]	1.14	3:4	CcpB
M73.4	6	4-7	20	5	4.2	-44 [8]	5.03	6:6	-
M61.3	6	5-10	25	22	6.4	-33 [4]	3.14	10:10	-
M2.1	6	4-8	22	9	13	-33 [56]	1.05	2:5	-
M18.6	4	3-8	25	9	17.2	-51 [55]	1.25	6:7	-
M29.1	3	2-6	25	1	3	-39 [62]	1.64	9:10	loose
M59.2	2	0-13	23	2	5.1	-52 [63]	2.43	4:5	loose

Legend for Table 5.1:

^a unique motif identifier build as Mxx.yy where yy is the identifier of the run and xx is the identifier of the motif in the run;

^b number of TSSs with an occurrence of the motif of estimated posterior probability at least 0.5, as computed across the last 10,000 MCMC sweeps; this number was used to order the motifs;

^c number of TSSs when the posterior probability cut-off is set to 0.8 (very likely) or 0.2 (possible);

^d width of the PWM corresponding to the number of columns included with posterior probability above 0.5;

^e number of columns in the PWM with highly informative nucleotide composition, i.e. information content expressed in bits, computed as $2 + \sum_{r \in \{A,C,G,T\}} \theta_{m,w,r} \log_2(\theta_{m,w,r})$, above 1;

^f estimated number of Watson-Crick paired columns in the PWM reflecting the degree of palindromness (highlighted in boldface when strong);

^g median position for the middle of the motif with respect to the TSS, the number between brackets corresponds to the inter-quartile range, both numbers are derived from the estimated probability function for the position of the motif m described by the variables K_m , $\lambda_{m,.}$ and $s_{m,.}$;

^h maximum across the 165 pairs of conditions for the median of the expression values associated with the TSSs counted in the first column;

ⁱ number of parallel runs (out of 10) in which this motif was found as obtained by the motif clustering procedure based on overlaps between motif occurrences, this information is given in the format xx:yy where xx and yy are the numbers obtained with cut-offs of 0.25 and 0.75, respectively.

^j a comment indicating the link to known transcription factors if identified or other observations that were made.

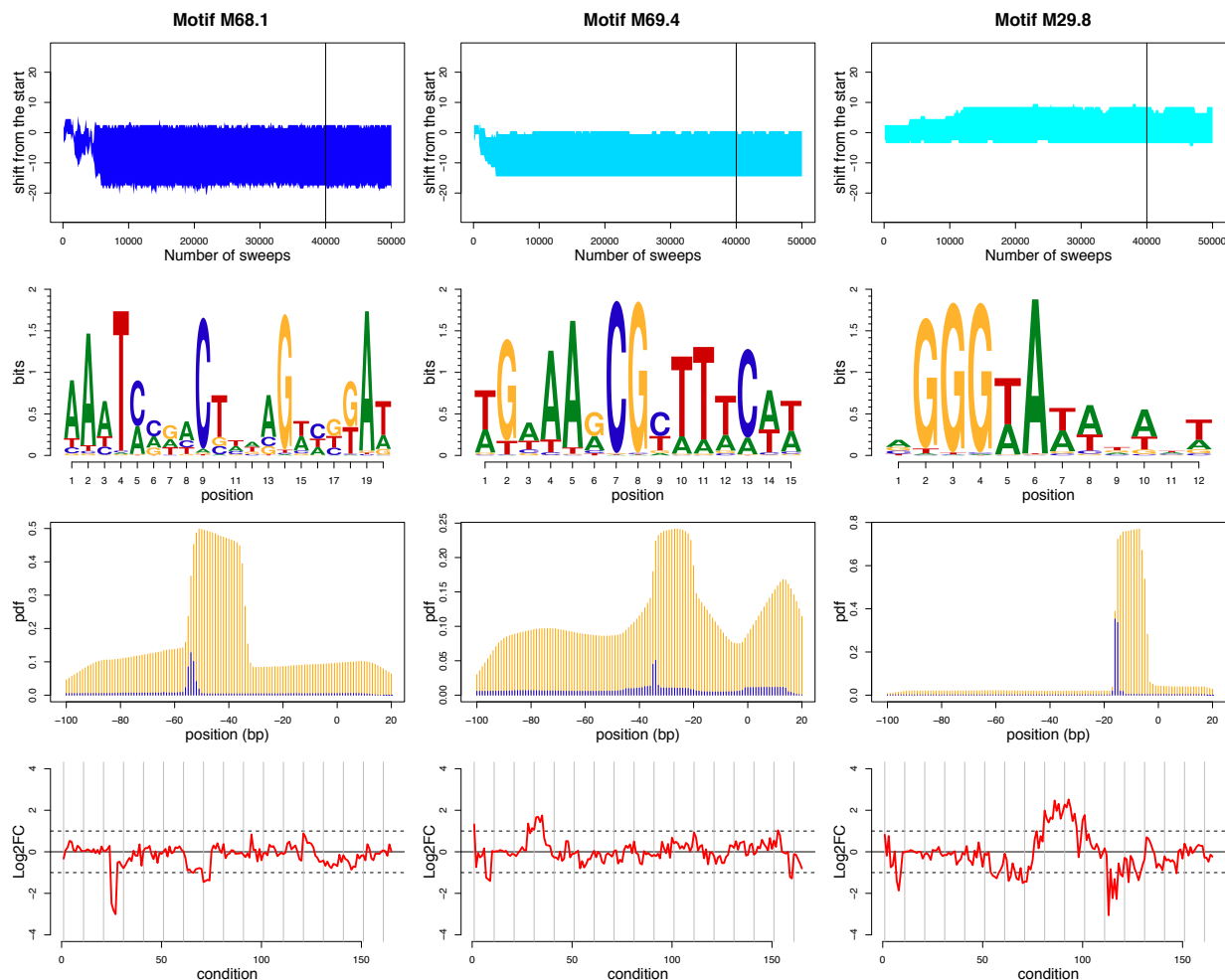


Figure 5.3: Illustration of three motifs found with our algorithm for *de-novo* motif discovery. From left to right, plots are organized in three columns corresponding respectively to motifs M68.1, M69.4, and M29.8 identified as the binding sites of LiaR, CcpA (CRE-box), and SigB (-10 box). First row: convergence plots that show the changes that occurred in the width of the PWM across the 50,000 sweeps of the algorithm, detailed explanation is available in Figure 5.1. Second row: sequence logo where the relative heights of the letters in one column represent the nucleotide composition and the total height represents the information content (in bits). Third row: estimated probability distribution function for the 5'-end of the occurrence (in blue) and probability of having the position covered by the occurrence (in orange). These probabilities are conditional on the presence of the motif in the promoter sequences; position is reported as the distance to the TSS (negative when upstream of TSS). Fourth row: log2 fold-changes across the 165 pairs of conditions that were used to define the expression covariates.

5.3 Comparisons of different models

This section is dedicated to the analysis performed in order to assess the impact of incorporating auxiliary data in the process of *de novo* motif discovery. In practice, we examined the importance of accounting for gene expression profiles (expression data), and modeling the distance from the TSS (position data). To that end, we ran our algorithm with four different settings incorporating: both expression data and position data (results presented in section 5.2), only expression data, only position data, and neither expression data nor position data. These four settings will be referred to as T1E1, T0E1, T1E0, and T0E0, respectively (where T stands for TSS and E for expression).

The analysis described in section 5.1 was performed separately on the output of 10 runs of 50,000 sweeps for each setting, and a total of 145 stable motifs were identified (40 for T1E1, 38 for T0E1, 37 for T1E0, and 30 for T0E0). We used Equation (5.2) to build a distance matrix of dimensions (145 * 145) for the total set of motifs. Figure 5.4 shows the tree resulted from applying hierarchical clustering (average-linkage) on this distance matrix. Table 5.2 shows a global view of the motifs obtained by the full setting, used as reference, and their closest matches that was obtained by each of the other settings.

In order to identify the reference motifs whose discovery depended on incorporating one type of auxiliary data, we accounted for distances in Table 5.2 to a threshold value. We set this threshold value at 0.25, since it corresponds to a high level of correlation and, as well, it is the value that we used in Section 5.1 to define two motifs from different runs as matches. Table 5.3 shows a summary of these matches, their widths, number of columns paired in palindromic structure, and numbers of occurrences. There were 11 out of the 40 motifs have matches in the T0E1 setting, 12 have matches in the T1E0 setting, and 6 have matches in the T0E0 setting.

Somewhat unsurprisingly, motifs that were detected only when accounting for position data (12 motifs), indeed have conserved distance from the TSS. Table 5.2 (column "pos") highlights this fact for some of these motifs, such as M36.4_T1E1 (SigA -35 (TTG)), M59.9_T1E1 (T-rich element in a reserved distance to the TSS), M27.6_T1E1 (SigA -35 (TTGAC)), M61.6_T1E1 (BglR2), and M17.4_T1E1 (CcpB). Similarly, there are motifs that were detected only when accounting for expression data, this set of motifs includes M69.4_T1E1 (CcpA (CRE-box)), M68.1_T1E1 (LiaR), M33.8_T1E1 (SigL), and M49.4_T1E1 (PrfA). These motifs tended to be linked with high variation in expression levels, summarized as \log_2 of the fold change value (see column "FC" in Table 5.2).

When examining the matches found by the setting which do not incorporate any auxiliary data, we found that 4 out of the 6 matches had a high number of paired columns. For instance, M62.3_T1E1 (Fur) was detected in the T0E0 setting, most likely for its palindromic structure. This finding highlights the relevance of modeling the palindromic structure. Finally, as expected, a number of biologically relevant motifs were detected only with the full setting, such as SigB (the two boxes), Rex, LexA, VirR, and Spx.

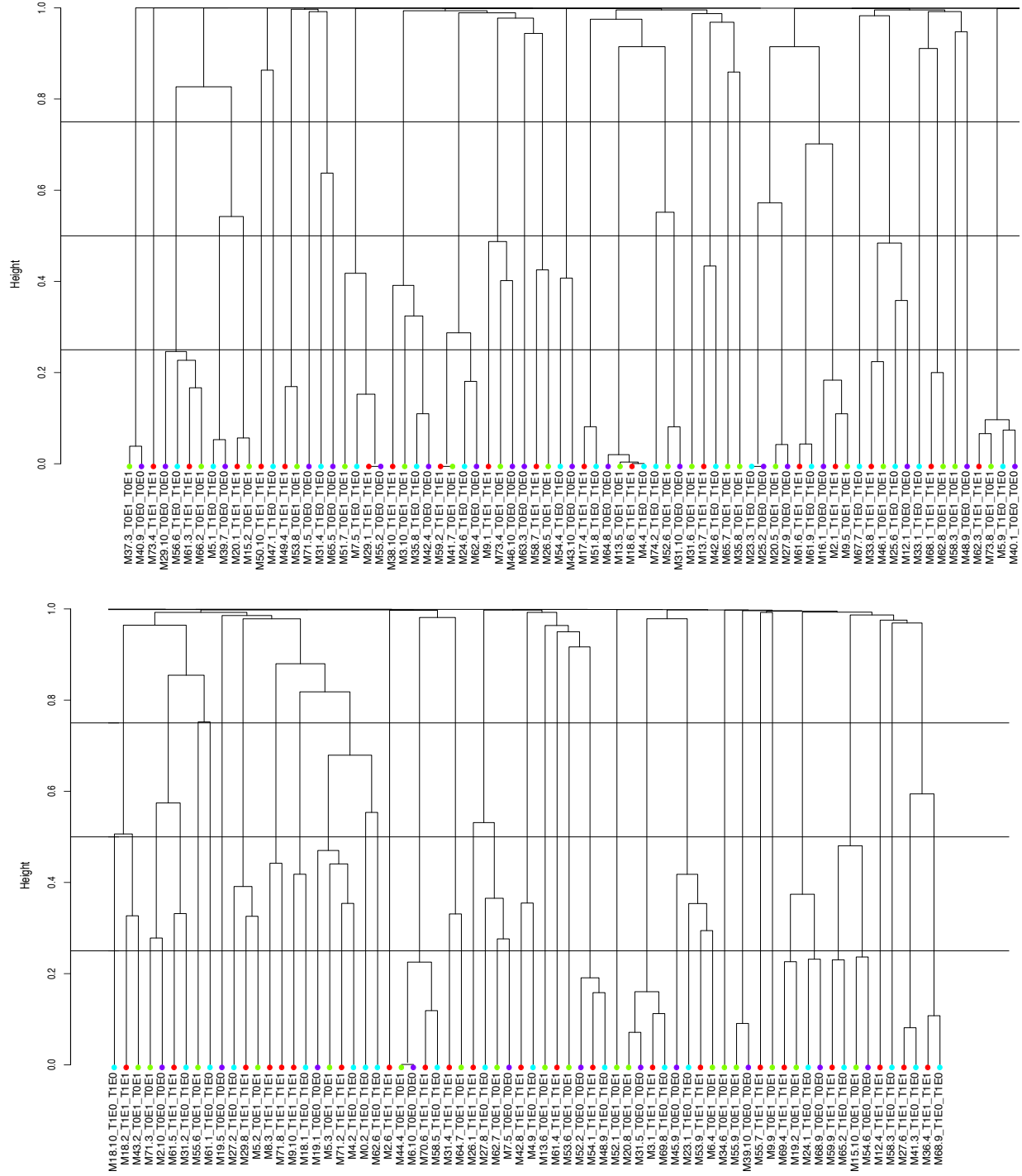


Figure 5.4: Hierarchical clustering of motifs resulted from different models. The tree was divided into two parts for the sake of better presentation.

Table 5.2: The 40 motifs found by the T1E1 setting and their closest matches from other settings.

motif ^a	#TSSs ^b	Pal. ^f	Pos. ^g	FC ^h	d_T0E1 ^k	c_T0E1 ^l	d_T1E0 ^m	c_T1E0 ⁿ	d_T0E0 ^o	c_T0E0 ^p
M71.2	1,325	1.3	-9 [1]	0.44	0.522	M5.3	0.354	M4.2	0.546	M19.1
M71.8	1,308	1.6	1 [1]	0.46	0.857	M5.3	0.448	M0.2	0.770	M19.1
M36.4	1,033	0	-32 [2]	0.48	0.974	M15.10	0.108	M68.9	0.978	M54.6
M55.7	836	1.7	-25 [40]	0.58	0.976	M15.10	0.991	M67.7	0.986	M54.6
M8.3	792	0	4 [1]	0.53	0.939	M5.3	0.688	M0.2	0.871	M19.1
M9.10	735	0.6	-13 [1]	0.50	0.630	M5.3	0.418	M18.1	0.738	M19.1
M61.5	561	0	14 [10]	0.41	0.625	M71.3	0.332	M31.2	0.654	M2.10
M59.9	297	0	-60 [42]	0.36	0.463	M15.10	0.230	M65.2	0.512	M54.6
M26.1	252	2.1	-79 [23]	0.73	0.488	M62.7	0.546	M27.8	0.561	M7.5
M27.6	240	0.4	-33 [2]	0.48	0.931	M35.8	0.081	M41.3	0.974	M45.9
M29.8	122	2.2	-13 [2]	3.06	0.326	M5.2	0.349	M27.2	0.969	M19.1
M9.1	116	5.6	-75 [37]	1.18	0.504	M73.4	0.943	M24.6	0.471	M46.10
M18.2	108	0.3	9 [12]	0.52	0.327	M43.2	0.442	M18.10	0.970	M2.10
M12.4	97	0.6	-32 [3]	3.87	0.988	M9.9	0.946	M68.9	0.983	M63.3
M69.4	87	10.9	-30 [53]	1.75	0.226	M19.2	0.379	M24.1	0.349	M68.9
M42.8	62	0.2	-86 [21]	0.61	0.987	M62.7	0.355	M4.9	0.968	M52.2
M31.4	39	18.8	-40 [61]	1.88	0.331	M64.7	0.987	M58.3	0.988	M7.5
M68.1	31	15	-46 [25]	3.00	0.200	M62.8	0.914	M33.1	0.994	M7.5
M58.7	27	9.9	-47 [56]	3.23	0.425	M26.5	0.979	M7.5	0.966	M46.10
M62.3	26	16.2	-16 [67]	1.89	0.066	M73.8	0.085	M5.9	0.112	M40.1
M38.10	22	12.1	-19 [28]	2.36	0.371	M3.10	0.394	M35.8	0.409	M42.4
M53.9	20	15.1	-47 [60]	1.52	0.294	M6.4	0.297	M23.1	0.454	M45.9
M2.6	19	1.1	-49 [2]	1.12	0.982	M64.7	0.992	M65.2	0.993	M54.6
M13.7	19	1.4	-58 [69]	0.89	0.956	M65.7	0.434	M42.6	0.991	M46.10
M61.6	14	16.3	-29 [3]	0.81	0.560	M20.5	0.043	M61.9	0.572	M27.9
M54.1	14	1	-56 [61]	1.04	0.891	M53.6	0.158	M48.9	0.160	M52.2
M61.4	13	0.9	-37 [62]	1.20	1.000	M3.10	0.923	M48.9	0.963	M52.2
M50.10	13	9.6	-77 [28]	0.93	0.992	M5.2	0.863	M47.1	1.000	M6.10
M70.6	11	19.2	-44 [63]	1.28	0.282	M44.4	0.119	M58.5	0.277	M6.10
M3.1	9	15.5	-43 [62]	1.52	0.122	M20.8	0.112	M69.8	0.194	M31.5
M33.8	8	4.2	-24 [23]	2.78	0.224	M46.1	0.633	M25.6	0.413	M12.1
M49.4	7	16.8	-35 [19]	10.55	0.169	M53.8	0.992	M41.3	1.000	M6.10
M20.1	7	5.4	-55 [3]	6.64	0.057	M15.2	0.524	M5.1	0.554	M39.7
M17.4	7	14.8	-39 [61]	1.14	0.946	M52.6	0.081	M51.8	0.948	M31.10
M73.4	6	4.2	-44 [8]	5.03	0.998	M9.9	0.994	M68.9	0.998	M19.1
M61.3	6	6.4	-33 [4]	3.14	0.167	M66.2	0.252	M56.6	0.254	M29.10
M2.1	6	13	-33 [56]	1.05	0.110	M9.5	0.734	M61.9	0.152	M16.1
M18.6	4	17.2	-51 [55]	1.25	0.010	M13.5	0.005	M4.4	0.025	M64.8
M29.1	3	3	-39 [62]	1.64	0.401	M51.7	0.153	M7.5	0.000	M55.2
M59.2	2	5.1	-52 [63]	2.43	0.000	M41.7	0.246	M24.6	0.329	M62.4

Legend for Table 5.2:

^a unique motif identifier build as Mxx.yy where yy is the identifier of the run and xx is the identifier of the motif in the run;

^b number of TSSs with an occurrence of the motif of estimated posterior probability at least 0.5, as computed across the last 10,000 MCMC sweeps; this number was used to order the motifs;

^f estimated number of Watson-Crick paired columns in the PWM reflecting the degree of palindromness;

^g median position for the middle of the motif with respect to the TSS, the number between brackets corresponds to the inter-quartile range, both numbers are derived from the estimated probability function for the position of the motif m described by the variables K_m , $\lambda_{m,.}$ and $s_{m,.}$;

^h maximum across the 165 pairs of conditions for the median of the expression values associated with the TSSs counted in the first column;

^k smallest distance to a motif from the T0E1 setting;

^l corresponding motif from the T0E1 setting, the one with the closest distance according to Equation (5.2);

^m smallest distance to a motif from the T1E0 setting;

ⁿ corresponding motif from the T1E0 setting;

^o smallest distance to a motif from the T0E0 setting;

^p corresponding motif from the T0E0 setting.

Table 5.3: Motifs from other settings with distance < 0.25 to motifs obtained by the full setting.

motif ^a	#TSSs ^b	w ^d	pal. ^f	match ^q	m_#TSSs ^r	m_w ^s	m_pal. ^t
M2.1	6	22	13	M9.5_T0E1	7	25	14.2
M3.1	9	21	15.5	M20.8_T0E1	9	20	16.3
M33.8	8	22	4.2	M46.1_T0E1	8	20	3.4
M20.1	7	25	5.4	M15.2_T0E1	7	24	5.3
M69.4	87	15	10.9	M19.2_T0E1	83	18	11.7
M61.3	6	25	6.4	M66.2_T0E1	7	25	5.3
M49.4	7	23	16.8	M53.8_T0E1	8	21	14.6
M68.1	31	20	15	M62.8_T0E1	28	21	15.6
M62.3	26	20	16.2	M73.8_T0E1	22	21	16.1
M18.6	4	25	17.2	M13.5_T0E1	4	25	18.6
M59.2	2	23	5.1	M41.7_T0E1	2	24	4
M70.6	11	24	19.2	M58.5_T1E0	9	25	20.6
M3.1	9	21	15.5	M69.8_T1E0	9	21	16
M59.9	297	21	0	M65.2_T1E0	365	21	0.1
M36.4	1033	6	0	M68.9_T1E0	1102	6	0
M27.6	240	6	0.4	M41.3_T1E0	246	6	0.4
M29.1	3	25	3	M7.5_T1E0	4	24	2.6
M54.1	14	23	1	M48.9_T1E0	19	23	0.7
M61.6	14	25	16.3	M61.9_T1E0	15	25	16.3
M62.3	26	20	16.2	M5.9_T1E0	23	21	16
M17.4	7	24	14.8	M51.8_T1E0	8	25	15.2
M18.6	4	25	17.2	M4.4_T1E0	4	25	17
M59.2	2	23	5.1	M24.6_T1E0	3	24	4.2
M2.1	6	22	13	M16.1_T0E0	8	22	13
M3.1	9	21	15.5	M31.5_T0E0	8	20	16.6
M29.1	3	25	3	M55.2_T0E0	3	25	2.5
M54.1	14	23	1	M52.2_T0E0	13	23	0.9
M62.3	26	20	16.2	M40.1_T0E0	22	21	15.9
M18.6	4	25	17.2	M64.8_T0E0	4	25	17.4

Legend for Table 5.3:

^a unique motif identifier build as Mxx.yy where yy is the identifier of the run and xx is the identifier of the motif in the run;

^b number of TSSs with an occurrence of the motif of estimated posterior probability at least 0.5, as computed across the last 10,000 MCMC sweeps; this number was used to order the motifs;

^d width of the PWM corresponding to the number of columns included with posterior probability above 0.5;

^f estimated number of Watson-Crick paired columns in the PWM reflecting the degree of palindromness;

^q matched motifs from other settings at a cutoff value of 0.25;

^r number of TSSs (explained in ^b) for the corresponding motif;

^s width of the PWM (explained in ^d) for the corresponding motif;

^t estimated number of Watson-Crick paired columns in the PWM (explained in ^f) for the corresponding motif.

5.4 Comparisons with other tools

In this section, I discuss the results obtained by applying some of the popular tools for *de novo* motif discovery on our dataset. We wanted to assess the relative performance of our developed method in comparison to these tools. Namely, we examined the following tools: MEME, RPMCMC, FIRE, MatrixREDUCE, and RED2 (see Chapter 2 for a presentation of these approaches). MEME and RPMCMC do not incorporate expression data in their searches and they require as input only the set of sequences in which we are aiming to detect motifs. On the contrary, FIRE, MatrixREDUCE, and RED2 are equipped to handle expression data in their searches. Both MatrixREDUCE and RED2 accept two dimensional expression data as an input. FIRE requires that the user cluster the expression data first, since it accepts only one dimensional expression data.

5.4.1 Comparison with MEME

The first tool that was chosen for comparison was MEME. Here, I review the parameters of interest to us, and those that needed to be changed for the sake of fair comparison with our tool. After that, I show the final results for three different sets of settings.

MEME parameters

We used the default values for most of the parameters. Two parameters were specifically of interest for us, "-bfile" and "-pal". The "-bfile" option incorporates a Markov background model file in order to account for biased distribution of nucleotides and groups of nucleotides in the sequences. A 0-order (by default) model adjusts for single letter biases, a 1-order model adjusts for dimer biases (e.g., GC content in DNA sequences), etc. We tested MEME for both 0-order and 3-order background models. The 0-order was selected in order to assess the results of the tool under default parameters values, while the 3-order was selected because the final results of our tool were obtained with a Markov background model of order three. The "-pal" option forces MEME to search only for palindromic motifs in the input DNA data. This parameter was of interest for us since we noticed that many of the motifs discovered by our tool, when applied on data from *L. monocytogenes*, were palindromic.

There were other set of parameters that needed to be changed for the sake of comparison. For instance, "-maxw" is a parameter that set the maximum width for the motifs

under search, by default this parameter is equal to 50, we adjusted it to 25 (similar to our tool). The option "nmotifs" indicates the number of motifs under search, this parameter was set to 75, since it is the number used by our tool to obtain the final results (justification of this choice is available in section 5.1).

Results of MEME

We ran MEME three times with three different set of parameters: in the first run we used the default parameters, in the second run we changed the background model to a 3rd order Markov model, while in the third run we were searching for palindromic motifs. In the three searches, "nmotifs" was set to 75, "maxw" was set to 25, and the rest of parameters were kept at their default values.

The outputs of the three runs are shown in Figure 5.5, in sub-figures A,B, and C, respectively. The motifs are ordered from top to bottom according to their E-value, we presented only motifs above a cutoff E-value of 1.

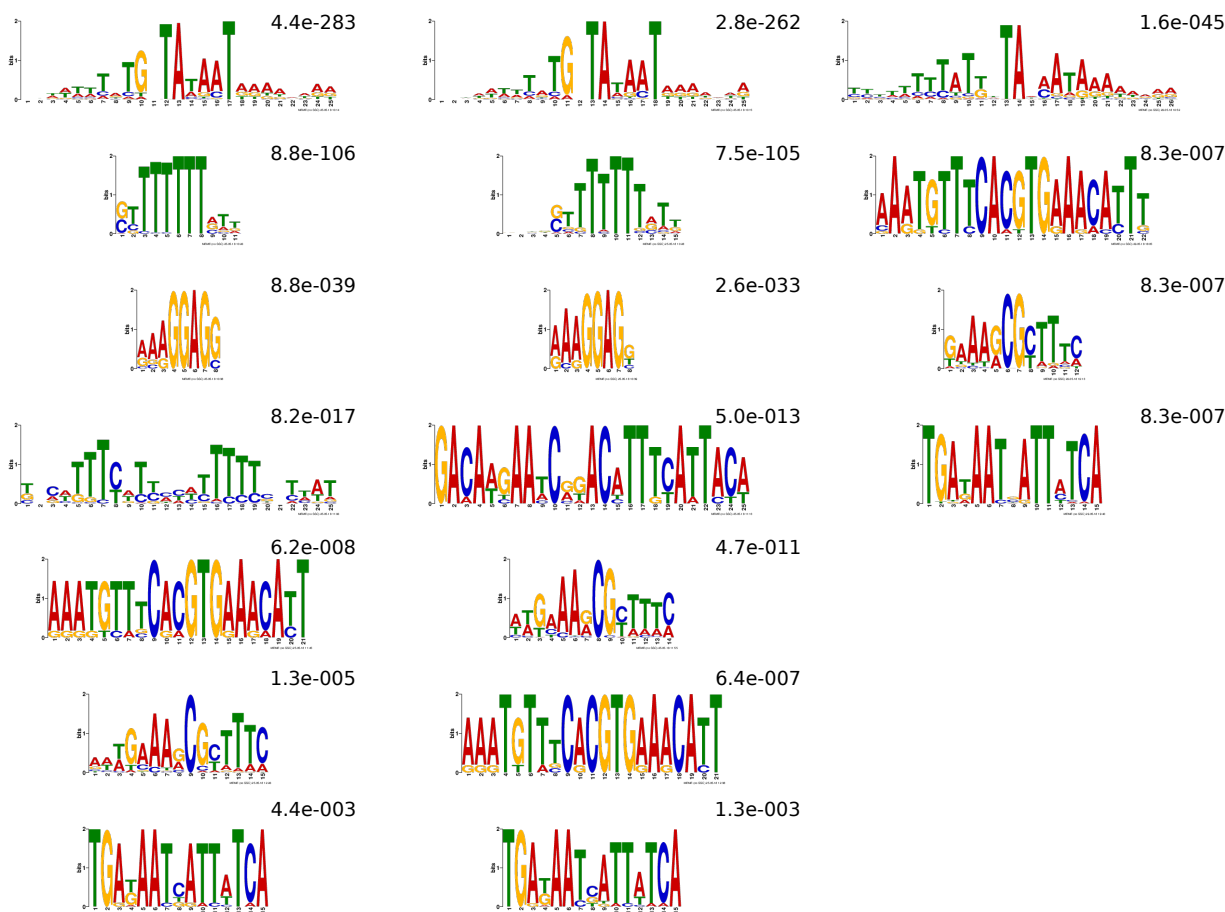
5.4.2 Comparison with Rpmcmc

Rpmcmc takes as an input the sequence data and it produces two output files. One for the PWMs of the discovered motifs, and the second is for the positions of these motifs inside the sequences.

We performed three runs on our data using RPMCMC. The first one was applied using the default set of parameters (number of replicas = 50, number of MCMC iterations = 520, and number of burn-in iterations = 20), the tool returned 158 motifs. On the second run, we changed the number of replicas to 75 while keeping the rest of parameters to their default values, the tool returned 56 motifs. In the third run, we changed the order of the background model to 3 (similar to the Markov order estimated by a dedicated parameter in our developed approach), while the rest of parameters were kept at their default values, the tool returned 249 motifs. In the three runs, the maximum width allowed for motifs was set to 25. In most of the cases, the discovered motifs were of short length.

The relation between the number of discovered motifs and the selected values for parameters was hard to understand. The results were not reproducible, in the sense that running the tool twice using the same set of parameters produced different results. We also noticed that Rpmcmc returned too many motifs and in a very short time relative to any other tool. These observations have given us the impression that the results of

C) 0-order model and searching for palindromic motifs



138

Rpmmc are not reliable, accordingly we did not investigate the results any further.

5.4.3 Comparison with FIRE

We have downloaded the source code of FIRE to run it locally on our data. Since FIRE makes use of expression data (but only one dimensional) in searching for statistically significant motifs, we prepared our expression data file before running the tool. The input files were: sequence data file and another file containing cluster membership for each sequence. The clustering was done using the average-linkage clustering method on the pairwise distances of the expression matrix. The tree was cut at a height value of 0.6 (corresponding to Pearson correlation of 0.4), then all genes that belonged to a cluster containing at least 20 members kept the cluster number as their identifier (a total of 14 clusters satisfied this criteria), and the rest of genes were assigned the same identifier, resulting in a total of 15 clusters.

Since applying this tool on our data using the default values did not result in discovering any motifs, we tried to modify the set of parameters. The two parameters that were changed from the default, after some analysis, were "k" and "jn_t". Where, "k" defines the length of the k-mer seeds (default is 7), and it was set by us to 5, while jn_t takes values between 0 and 10. This parameter defines the robustness index threshold (default is 6), it was set by us to 0 (thereby discarding the robustness test). Using this set of parameters, the tool returned 5 short motifs, shown in Figure 5.6, that did not appear biologically relevant which may not be surprising given that we discarded the robustness test.

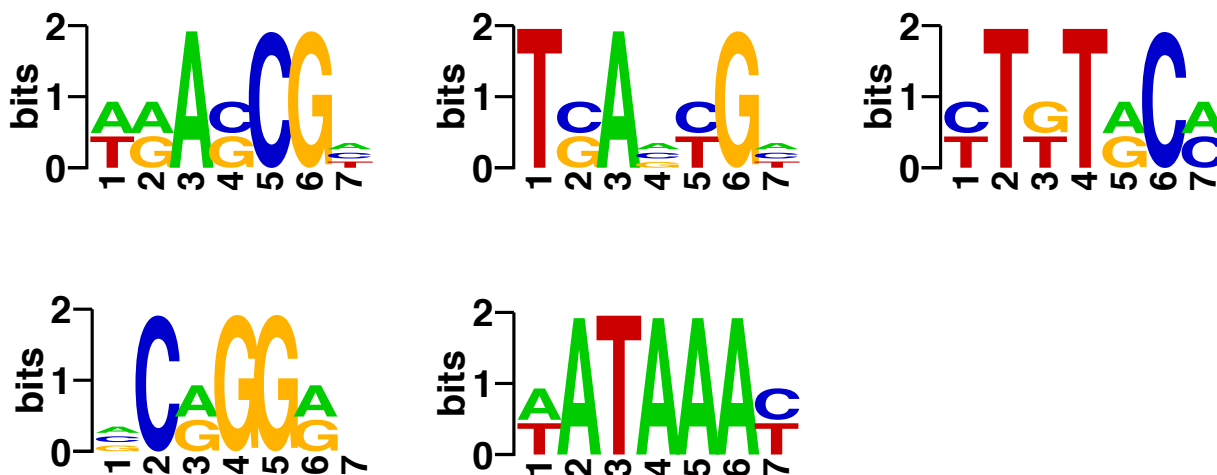


Figure 5.6: The motifs discovered by applying FIRE

5.4.4 Comparison with REDUCE-Suite-v2.2

Out of the tested tools, MatrixREDUCE with RED2 are the most related to our method, since they can handle two dimensional expression data. This makes their results more relevant to compare with our developed method. When applying MatrixREDUCE, we decided not to apply the tool on the original expression matrix composed of 165 columns (log 2 expression ratio between pairs of conditions), but to use the output of the PCA analysis (explained in subsection 4.2.2) as our expression data. The tool takes as input beside this expression data file, the sequence data.

The results obtained by applying MatrixREDUCE, using the default set of parameters, on our data are illustrated in figure 5.7. The tool returned only four motifs of maximum width 8, out of which only SigB (-10 box) was in common with our discovered motifs.

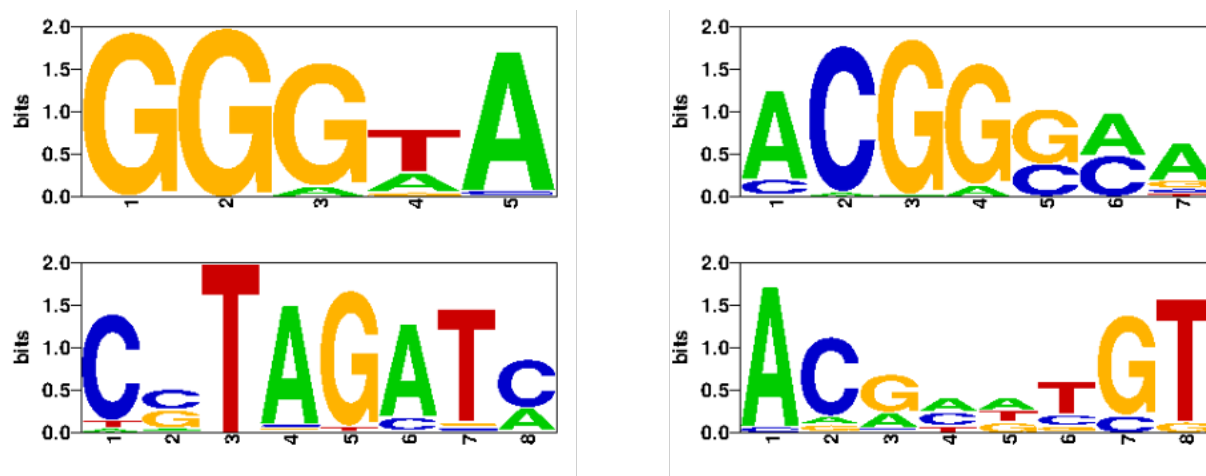


Figure 5.7: The motifs discovered by applying MatrixREDUCE with the default set of parameters. SigB (-10 box) is shown on the upper left of the figure.

5.4.5 Comparison with RED2

RED2 handles two dimensional expression data like MatrixREDUCE, but with as few assumptions as possible about how the presence/absence of a specific motif is linked to the expression profile (see subsection 2.4.2). These two facts, make RED2 the closest tool to ours from this point of view. The input used for RED2 was similar to MatrixREDUCE, the sequence data file and expression covariates resulted from applying PCA on the original expression matrix.


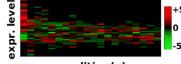


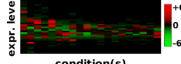
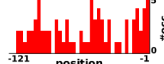

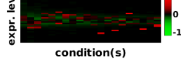
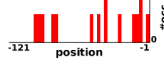

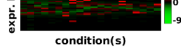

RED2 offers a user-friendly web interface, but with limited freedom on the choice of parameters, for this reason it was necessary to download the software and run it locally. When we applied RED2 on our sequence and expression data using the default parameters we ended up finding no motifs. We had then to do analysis on the different parameters individually in order to detect those that needed to be adjusted.

I briefly recall the procedure of RED2 because it will be needed for understating the tuning of the parameters. RED2 starts by computing the score of every possible q-mer ($q = 7$ by default) according to one of the scoring function (mutual information by default, also called global criteria) and then proceeds with a local optimization of the highest scoring q-mers (those with an FDR below a given threshold), called the seeds, transforming them into better scoring motifs.

Two parameters were specifically essential to adjust to our data size, "-F" which corresponds to the False Discovery Rate (FDR) and "-K" which represent the neighborhood size. The FDR is set by default to 0.001 which is very low for our data set, this value was set by us to 0.1. The neighborhood size is by default set to 200, which may not be the most relevant value in the case of bacteria where the average number of occurrences is usually less than that.

Figure 5.8 shows the results obtained when we applied RED2 on our data set. Using the default values for parameters but changing only the false discovery rate (FDR) to be 0.1 instead of 0.001, the tool returned four short motifs, only SigB (-10 box) was common with our discovered 40 motifs. When we set "-K" to 10, while keeping the FDR value at 0.1, the tool returned the 5 motifs, we could not link any of them with our results.

A) Default parameters with FDR=0.1

id	logo	score	# genes	expression	distances	strand	match	GO terms
#1		21.639	324			→		
#2		11.662	79					
#3		9.657	15					
#4		8.185	32					

B) Default parameters with FDR=0.1 and neighbourhood size=10


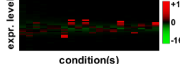
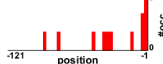

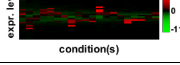
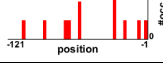
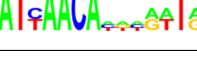
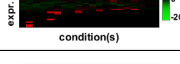
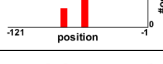




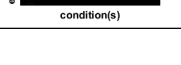

id	logo	score	# genes	expression	distances	strand	match	GO terms
#1		20.306	11					
#2		13.574	12					
#3		13.049	7					
#4		8.556	136					
#5		8.124	14					

Figure 5.8: The results of RED2 in two different settings of parameters. Sub-figure A shows the results when only FDR is changed, while sub-figure B shows the results when both the FDR and neighborhood size are changed. We noticed that accounting for a bigger neighborhood size has resulted in discovering motifs that are highly associated with expression data (like SigB), while in the case of a smaller neighborhood, the motif had more freedom to extend.

Chapter 6

Discussion

This chapter aims to discuss the directions of future studies that may extend the work realized during my PhD, including the possible extensions for our developed model and the challenges that will accompany these extensions. The chapter is divided into three parts. The first part discusses several choices that were made in the thesis. The second part reviews possible extensions on the model. The third concerns the application of the approach, including data preparation and analysis of other data-sets.

6.1 Looking backward: comparison with TreeMM and choices made for the validation of the approach

One of the aims of this PhD project was to develop a coherent statistical framework to address the task of discovering the main regulons of a bacteria by making use of two types of increasingly available transcriptome data : precise information of TSS and a wealth of information on condition-dependent expression profiles. From the beginning, our goal was to build this approach on PWM-based statistical models of the DNA sequences and to incorporate transcriptome data as an additional source of information on the sequences. As already discussed in Section 2.4.2, this point of view contrasts with another, more mechanistic, point of view that consider the transcription factor binding motifs as sequence features that should be able to explain the transcriptome data. Our rationale was to preserve the benefits of the use of simple well-established probabilistic sequence models, making it also possible to find motifs that are not connected to the transcriptome data. By adopting this modeling point of view, our work can be seen as an extension of the TreeMM model

developed by Nicolas et al. (2012).

The generality of the previous model (TreeMM) was however limited since it was specifically tailored for the discovery of Sigma factor binding sites whose specificity is to define and partition the promoter space (Gruber and Gross, 2003). Indeed, for Sigma factors, there is (or at least we expect) only one motif occurrence per sequence. This fact, *per se*, make it unnecessary to model motifs overlaps which is a huge complication in the algorithm and makes that the C++ code for the work described in this thesis had to be developed from scratch.

Besides motif occurrences overlap, the other most significant difference between the two models is in the specific framework that serves to incorporate the expression data. The TreeMM model takes as input a single hierarchical clustering tree which seems particularly relevant when the goal is to find Sigma factor binding sites since Sigma factors do partition the promoter space and implement a very first and strong level of regulation of the expression. Furthermore since only one occurrence of Sigma factor binding motifs is modeled per sequence, the distribution of all the motifs in the sequence set given the tree needs to be modelled jointly by a single probabilistic model. The model proposed by Nicolas et al. (2012) adds only two parameters compared to a simple mixture model. In contrast, our approach adopted a different modeling framework whose goal is to accommodate more subtle effects that could result from the regulation of a same gene by multiple factors. Such cases might be better represented by a position in a multidimensional space than by the position in a tree structure. The extended Probit model that we propose is able to incorporate information on the position of the genes in the expression space encoded both in the form of a tree and in the form of coordinates in a multidimensional space. It allows the simultaneous use of different representations of the expression space and automatic selection of those that are the most relevant for each motif.

When listing the differences between our model and TreeMM, one should also note that Sigma factors are composite (two-element) motifs that are found at a very specific distance from the TSS and this distance is common to all Sigma factors. In Nicolas et al. (2012), the modeling of the positions of motif occurrences in the sequence relied therefore on a single set of parameters instead of one set of parameters k_m , d_m , λ_m per motif in our model. Furthermore, most of the randomness in the positions of occurrences of the sigma factor binding motifs stemmed from uncertainty on the exact position of the TSS which did not have the 1 bp resolution allowed by the dedicated RNA-Seq protocol for genome-wide TSS mapping that we used here (Wurtzel et al., 2012). We can also note that Sigma

factor binding motifs do not exhibit palindromic structures and this aspect of our model is therefore completely new with respect to TreeMM (while the modeling of two-element motifs is specific to TreeMM).

As part of the List_Maps project (see section 1.3), one goal of the PhD work was to refine our knowledge of the main regulons of the bacterium *L. monocytogenes*. While the transcriptional regulatory network of this bacterium is not known as well as those of the two main model bacteria (the gram-negative *E. coli* and the gram-positive *B. subtilis*), a substantial body of knowledge is nevertheless available due to experimental work on this bacterium and its proximity to other well-known gram-positive bacteria including *B. subtilis*. We decided to use this knowledge as a reference point to validate the practical relevance of our approach after carefully checking the mathematics and implementation of the algorithm by the simulation approach briefly described in section 3.1.4. In this process, we bypassed a study of synthetic data sets that would have been particularly tedious to simulate given the considerable diversity of (and uncertainty on) the characteristics of the motifs by which we are interested. These are expected to differ in terms of characteristics of the PWM (width, information content, palindromic structure), but also number of occurrences and randomness of the distance to the TSS, and type of link to the expression covariates. In practice, we did several back-and-forth between the development and tuning of the methodology and the analysis of the results on this real data set until finding the settings described in this work that give results that are interesting in terms of biology.

6.2 Extensions of the model and improvements of the algorithm

In the biological literature (see chapters 1 and 2), it has been shown that multiple occurrences of the same motif per sequence is a common phenomenon in the bacterial promoter regions. Accordingly, accounting for multiple occurrences when developing a *de novo* motif discovery tool is indeed a desirable element. Our developed tool, in its current version, assumes one or zero motif occurrences per sequence. A workaround to this issue without having to change the current model is to consider motifs that were already discovered in one run of the algorithm, and search for further occurrences of them in the post processing of the results. Alternatively, a model extension that would account for multiple occurrence is possible but would be faced with many challenges. Out of these challenges, there are:

- Modeling overlaps between motifs of the same kind, since the global mechanism of the algorithm overlapping between motifs should be allowed (reasons explained in Section 2.5), but when two motifs of the same kind overlap, updating some parameters (ex. motif width) will be more complicated compared to the case of overlaps between two different motifs.
- Extending the regression model used to incorporate expression data is another modeling obstacle, since the used model gives a probability that a specific sequence contains the motif, it would not be straightforward to extend this model to a more complex one that as well gives an indication on the number of motifs contained by a sequence. One option could be to introduce a new independent variable taking values in $\{1, \dots\}$ that would serve, when the motif is predicted present based on the probit model, to account for the number of occurrences found in the sequence (e.g. via a Geometric distribution). In this case, no link is established between expression data and the presence of more than one occurrence.

Modeling the variable B , which is responsible on assigning membership for nucleotides when motif occurrences overlap, can be either θ -dependent weight mixture model (accounting for information content), or equal-weight mixture model. This can easily be extended even further by introducing a tuning parameter that indicates to which extent should the assignment of B be dependent on the information content. In our current framework, updating B according to the information content is not applicable in the case of searching for palindromic motifs. Accounting simultaneously for θ -dependent weight mixture model and palindromic structures would require to develop a new MCMC step to update the palindromic structure. This is challenging, and would probably come at a cost in terms of mixing of the MCMC algorithm, since direct sampling from the conditional distribution of the parameters that describe the palindromic structure seems impossible with the θ -dependent weight mixture model for motif overlaps. Therefore, a Reversible-Jump Metropolis-Hastings step would have to be used instead of the dimension-changing Gibbs-step.

Updating the reference position of a motif is done without updating the positions of the breakpoints of the piecewise constant probability density function that model the position of occurrence of the motif in the sequence. As a result of this choice, for an undetermined number of iterations the update of the reference position was probably rejected. One extension would be to do a simultaneous adjustment of the positions of the breakpoints with the update of reference positions, which should, in principle, improve the mixing of

the algorithm. Another relatively easy, and probably useful, improvement of the MCMC algorithm would be to implement the simultaneous update of the coefficients of the probit model that link the values of the active expression covariates to the probability of motif occurrence.

Modeling two-component (aka bipartite) motifs with a gap between them is another possible extension. Although, there are other ways to detect such kind of motifs without directly modeling them. For instance, accounting for dependency between occurrences of different motifs may be another, and more general way, to address the problem, since these motifs can be regarded as two motifs that always appear together. Of note, even without explicit modelling dependencies between motif occurrences, we were able to detect bipartite motifs such as the sigma factor binding sites (SigA and SigB), probably helped by the modeling of exact distance to the TSS and of accounting for expression data.

6.3 Additional analyses

In chapter 4, we presented three different approaches that were used by us to summarize the expression matrix (PCA, ICA, and hierarchical clustering). These dimensionality reduction techniques are known to remove the redundancy of the data and reduced the columns of the expression matrix to 50 covariates instead of 165 covariates (number of comparisons in the original expression matrix). We do not know which approaches are the best to summarize expression data, nor even to which extent summarizing the dimension reduction is really useful. Thus, comparing the different ways of incorporation expression data, including using the original expression matrix without summarizing, in terms of the quality of the output results, would be indeed interesting.

One other type of auxiliary data, beside expression and positional data, that can be incorporated to our tool is the available information on known motifs (motifs tend to be more conserved than regulons). This extension would not require important modifications to the model since it can be envisioned to incorporate this information into the priors for the PWMs. It would probably improve the sensitivity of the motif discovery.

The results presented in this thesis consisted of the output of applying our methodology to the bacterium (*Listeria monocytogenes*) which was the main focus of the European project (List_MAPS). An obvious direction for future works would be to apply our method on other data sets. To name few candidates, *Bacillus subtilis*, *Staphylococcus aureus*, and *Flavobacterium psychrophilum*, are bacteria on which a high quality expression data sets

have been collected as a result of collaborations between MaIAGE and wet biology labs.

The algorithm for updating the model parameters is written in the C++ language and it has been uploaded online with an open access, one possible future direction is to build a dedicated website to provide a more friendly interface for the developed tool. Among the challenges, is the running time of the algorithm (a couple of weeks) which may not be expected from a user of a web interface.

The way we extended the probit regression model to binarize the vector of predictors and to take into account trees is, to our knowledge, new and could be generalized to other applications. One benefit, as it has been explained in Chapter 3 is that it is possible to use this model even when the dimension of the data used as predictors is relatively high.

Résumé en français

Les facteurs de transcription jouent un rôle clé dans la médiation de l'adaptation des bactéries aux conditions environnementales. Des algorithmes puissants et des approches sophistiquées ont été développés pour la découverte de leurs sites de liaison à l'ADN, mais l'identification *de novo* automatique des principaux régulateurs d'une bactérie à partir des données du génome et du transcriptome reste un défi. L'approche que nous proposons ici pour traiter cette tâche est fondée sur un modèle probabiliste de la séquence d'ADN qui peut utiliser des informations précises sur la position des sites de départ de la transcription et des profils de transcription mesurés dans une collection de conditions expérimentales. Les principales nouveautés introduites consistent à permettre les chevauchements d'occurrences de motifs et à incorporer des covariables résumant les profils de transcription dans la probabilité d'occurrence dans une région promotrice donnée. Chaque covariable peut correspondre à la coordonnée du gène sur un axe (obtenu par exemple par PCA ou ICA) ou à sa position dans un arbre (obtenue par exemple par un regroupement hiérarchique). Tous les paramètres sont estimés dans un cadre bayésien à l'aide d'un algorithme MCMC transdimensionnel dédié. Cela permet d'ajuster simultanément, pour de nombreux motifs et avec de nombreuses covariables de transcription, la largeur des matrices de poids-position correspondantes, le nombre de paramètres permettant de décrire les positions par rapport au site de début de la transcription, et la sélection des covariables pertinentes.

Le manuscrit de thèse est divisé en six chapitres. Le premier et deuxième chapitres introduisent, respectivement, le contexte biologique et le contexte méthodologique de ce travail. Le troisième chapitre présente le noyau méthodologique de la nouvelle approche développée au cours de cette thèse (modèle probabiliste, inférence bayésienne). Le quatrième chapitre est dédié à la collecte et à la préparation des données (séquences et profils de transcription), qui englobe les techniques de réduction de dimensionnalité ayant servi à résumer la position des promoteurs dans l'espace des profils de transcription. Le cinquième chapitre est consacré à la présentation des résultats obtenus sur la bactérie *Listeria mono-*

cytogenes qui était au centre du projet européen List_MAPS dans lequel ce travail a eu lieu. Dans ce chapitre, les résultats sont également comparés à ceux obtenus avec d'autres méthodes de découverte de motifs. Le dernier chapitre aborde brièvement les orientations futures qui pourraient être envisagées pour poursuivre le travail réalisé dans le cadre de ce projet de thèse.

Le dernier chapitre a pour objectif de discuter des orientations d'études futures susceptibles de prolonger le travail réalisé durant ma thèse, y compris les possibles extensions pour notre modèle développé et les défis qui seront accompagner ces extensions. Le chapitre est divisé en trois pièces. La première partie traite de plusieurs choix qui ont été faits dans le thèse. La deuxième partie examine les extensions possibles du modèle. Le troisième concerne l'application de l'approche, y compris la préparation et analyse d'autres ensembles de données.

Regard en arrière: comparaison avec TreeMM et choix effectués pour la validation de l’approche

L’un des objectifs de ce projet de thèse était de développer une approche cohérente cadre statistique permettant de découvrir les principales régulateurs d’une bactérie en utilisant deux types de plus en plus données de transcriptome disponibles : informations précises sur le TSS et richesse d’informations sur les profils d’expression dépendants de la condition. Du Au départ, notre objectif était de construire cette approche sur la base de PWM. modèles statistiques des séquences d’ADN et à incorporer le transcriptome les données en tant que source d’information supplémentaire sur les séquences. Comme déjà discuté dans la section 2.4.2, ce point de Cette vision contraste avec un autre point de vue, plus mécaniste, qui considérer les motifs de liaison du facteur de transcription comme des caractéristiques de séquence cela devrait pouvoir expliquer les données du transcriptome. Notre rationnel était de préserver les avantages de l’utilisation de techniques simples et bien établies. modèles de séquence probabilistes, permettant également de trouver des motifs qui ne sont pas connectés aux données du transcriptome. En adoptant cette modélisation, notre travail peut être considéré comme une extension de la Modèle TreeMM développé par Nicolas et al. (2012).

La généralité du modèle précédent (TreeMM) était cependant limitée car il a été spécialement conçu pour la découverte du facteur Sigma sites de liaison dont la spécificité est de définir et de partitionner le promoteur space (Gruber and Gross, 2003). En effet, pour les facteurs Sigma, il n’existe (ou du moins on s’y attend) qu’une seule occurrence de motif par séquence. Ce fait, *per se*, rend inutile la modélisation motifs se chevauchent qui est une énorme complication de l’algorithme et fait que le code C++ pour le travail décrit dans cette thèse devait être développé à partir de zéro.

Outre les occurrences de motifs qui se chevauchent, l’autre plus important La différence entre les deux modèles se situe dans le cadre spécifique sert à incorporer les données d’expression. Le modèle TreeMM prend comme entrer un seul arbre de clustering hiérarchique qui semble particulièrement pertinent lorsque l’objectif est de trouver des sites de liaison au facteur Sigma puisque Les facteurs Sigma partitionnent l’espace promoteur et implémentent une premier et fort niveau de régulation de l’expression. en outre dans la mesure où une seule occurrence de motifs de liaison au facteur Sigma est modélisée par séquence, la distribution de tous les motifs de l’ensemble de séquences étant donné que l’arbre doit être modélisé conjointement par un seul probabiliste modèle. Le modèle pro-

posé par Nicolas et al. (2012) n'ajoute que deux paramètres comparés à un modèle de mélange simple. En revanche, notre Cette approche a adopté un cadre de modélisation différent visant à prendre en compte des effets plus subtils pouvant résulter de la réglementation d'un même gène par plusieurs facteurs. Ces cas pourraient être mieux représenté par une position dans un espace multidimensionnel que par le position dans une arborescence. Le modèle Probit étendu que nous proposer est capable d'intégrer des informations sur la position du des gènes dans l'espace d'expression codé à la fois sous la forme d'un arbre et sous la forme de coordonnées dans un espace multidimensionnel. Cela permet au utilisation simultanée de différentes représentations de l'espace d'expression et sélection automatique de ceux qui sont les plus pertinents pour chaque motif.

En énumérant les différences entre notre modèle et TreeMM, il convient de Notez également que les facteurs Sigma sont des motifs composites (à deux éléments) qui se trouvent à une distance très précise du TSS et cette distance est commun à tous les facteurs Sigma. Dans Nicolas et al. (2012), le modélisation des positions des occurrences de motif dans la séquence utilisée donc sur un seul ensemble de paramètres au lieu d'un ensemble de paramètres k_m , d_m , λ_m par motif dans notre modèle. En outre, la majeure partie du caractère aléatoire des positions de la présence de motifs de liaison au facteur sigma est due à l'incertitude sur la position exacte du TSS qui n'avait pas le 1 bp résolution autorisée par le protocole dédié RNA-Seq pour le génome entier Le mappage TSS que nous avons utilisé ici (Wurtzel et al., 2012). nous pouvons noter également que les motifs de liaison au facteur Sigma ne présentent pas de trouble palindromique. structures et cet aspect de notre modèle est donc complètement nouveau par rapport à TreeMM (alors que la modélisation de motifs à deux éléments est spécifique à TreeMM).

Dans le cadre du projet List_Maps (voir la section 1.3), L'objectif du travail de doctorat était d'affiner nos connaissances des principaux régulateurs de la bactérie *L. monocytogenes*. Tandis que le réseau de régulation de la transcription de cette bactérie n'est pas connu ainsi que ceux des deux principales bactéries modèles (la bactérie gram-négative *E. coli* et le gram-positif *B. subtilis*), une valeur substantielle un corpus de connaissances est néanmoins disponible grâce au travail expérimental sur cette bactérie et sa proximité avec d'autres bactéries à Gram positif bien connues bactéries comprenant *B. subtilis*. Nous avons décidé d'utiliser cette connaissance comme point de référence pour valider la pertinence pratique de notre approche après avoir soigneusement vérifié les mathématiques et la mise en œuvre de l'algorithme par la méthode de simulation brièvement décrite dans section 3.1.4. Dans ce processus, nous avons contourné une étude sur des ensembles de

données synthétiques qui auraient particulièrement fastidieux de simuler étant donné la diversité considérable (et incertitude sur) les caractéristiques des motifs par lesquels nous sommes intéressés. Ceux-ci devraient différer en termes de caractéristiques du PWM (largeur, contenu informationnel, palindrome structure), mais aussi le nombre d'occurrences et le caractère aléatoire des distances au TSS et type de lien aux covariables d'expression. Dans la pratique, nous avons fait plusieurs allers-retours entre le développement et mise au point de la méthodologie et de l'analyse des résultats sur ce réel ensemble de données jusqu'à trouver les paramètres décrits dans ce travail qui donnent des résultats intéressants en biologie.

Extensions du modèle et améliorations de l'algorithme

Dans la littérature biologique (voir les chapitres 1 et 2), il a été démontré que plusieurs occurrences du même motif par séquence sont un phénomène courant dans les régions promotrices bactériennes. En conséquence, la comptabilisation de plusieurs occurrences lors du développement d'un outil de découverte de motif *de novo* est en effet un élément souhaitable. Notre outil développé, dans sa version actuelle, suppose une ou zéro occurrences de motif par séquence.

Une solution de contournement à ce problème sans avoir à changer le modèle actuel consiste à prendre en compte les motifs déjà découverts dans une exécution de l'algorithme et à en rechercher d'autres occurrences dans le post-traitement des résultats. Alternative-ment, une extension de modèle qui prend en compte plusieurs occurrences est possible mais serait confrontée à de nombreux défis.

Parmi ces défis, il y a :

- La modélisation chevauche des motifs du même type, car le mécanisme global de l'algorithme se chevauchant doit être autorisé (raisons expliquées à la section 2.5), mais lorsque deux motifs du même type se chevauchent, actualise certains paramètres. (ex. largeur du motif) sera plus compliqué que dans le cas de chevauchements entre deux motifs différents.
- Extension du modèle de régression utilisé pour incorporer une expression les données sont un autre obstacle à la modélisation, car le modèle utilisé donne une probabilité qu'une séquence spécifique contienne le motif, il serait pas simple d'étendre ce modèle à un modèle plus complexe qui donne également une indication sur le nombre de motifs contenus par une séquence.
- Une option pourrait être d'introduire un nouveau variable prenant des valeurs dans $\{1, \dots\}$ qui serviraient, lorsque le motif est prédit présent sur la base du modèle probit, pour tenir compte de le nombre d'occurrences trouvées dans la séquence (par exemple via un fichier géométrique) Distribution).
- Dans ce cas, aucun lien n'est établi entre données d'expression et la présence de plus d'une occurrence.

Modélisation de la variable B , responsable de l'affectation l'adhésion aux nucléotides lorsque les occurrences de motifs se chevauchent, peut être soit un modèle de mélange

pondéral dépendant de *theta* (représentant contenu de l'information), ou un modèle de mélange à poids égal. Cela peut facilement être étendu encore plus loin en introduisant un paramètre de réglage qui indique dans quelle mesure l'affectation de *B* devrait dépendre de le contenu de l'information.

Dans notre cadre actuel, mettre à jour *B* selon le contenu de l'information n'est pas applicable dans le cas de recherche de motifs palindromiques.

Comptabilité simultanément pour Modèle de mélange pondéral *theta* et structures palindromiques développerait une nouvelle étape MCMC pour mettre à jour le système palindromique structure.

C'est un défi et aurait probablement un coût en termes de mélange de l'algorithme MCMC, car l'échantillonnage direct à partir de la distribution conditionnelle des paramètres décrivant la structure palindromique semble impossible avec le *theta*-dépendent modèle de mélange pondéral pour les recouvrements de motifs. Par conséquent, un saut réversible L'étape Metropolis-Hastings devrait être utilisée à la place du changement de dimension Gibbs-step.

La mise à jour de la position de référence d'un motif se fait sans mise à jour les positions des points d'arrêt de la probabilité constante par morceaux fonction de densité qui modélise la position d'occurrence du motif dans la séquence. A la suite de ce choix, pour un nombre indéterminé itérations, la mise à jour de la position de référence était probablement rejeté.

Une extension serait de faire un ajustement simultané de les positions des points d'arrêt avec la mise à jour de référence positions, qui devraient, en principe, améliorer le mélange des algorithme.

Une autre amélioration relativement facile, et probablement utile, l'algorithme MCMC consisterait à mettre en œuvre la mise à jour simultanée de les coefficients du modèle probit qui relie les valeurs du expression active covarie à la probabilité d'occurrence du motif.

Modélisation de motifs à deux composants (ou bipartites) avec un espace entre eux est une autre extension possible, bien qu'il existe d'autres moyens de détecter ce genre de motifs sans les modeler directement. Pour exemple, la prise en compte de la dépendance entre les occurrences de différentes les motifs peuvent être un autre moyen, plus général, de résoudre le problème, puisque ces motifs peuvent être considérés comme deux motifs qui apparaissent toujours ensemble.

À noter, même sans dépendances de modélisation explicites entre occurrences de motifs, nous avons pu détecter des motifs bipartites tels que comme sites de liaison du facteur sigma (SigA et SigB), probablement aidés par la modélisation de la distance exacte au TSS et de

la comptabilisation des données d'expression.

6.4 Analyses supplémentaires

Dans le chapitre 4, nous avons présenté trois différents les approches que nous avons utilisées pour résumer la matrice d'expression (PCA, ICA et classification hiérarchique). Ces dimensionnalité les techniques de réduction sont connues pour éliminer la redondance des données et réduit les colonnes de la matrice d'expression à 50 covariables au lieu de 165 covariables (nombre de comparaisons dans le rapport initial matrice d'expression). Nous ne savons pas quelles approches sont les meilleures pour résumer les données d'expression, ni même dans quelle mesure résumant les la réduction des dimensions est vraiment utile. Ainsi, en comparant les différents moyens d'incorporer des données d'expression, y compris l'utilisation du matrice d'expression sans résumer, en termes de qualité de la résultats de sortie, serait effectivement intéressant.

Un autre type de données auxiliaires, à côté de l'expression et de la position Les données pouvant être intégrées à notre outil sont les données disponibles. informations sur les motifs connus (les motifs ont tendance à être plus conservés que les régulons). Cette extension ne nécessiterait pas de modifications importantes le modèle puisqu'il est envisageable d'intégrer cette information dans les priors pour les PWM. Cela améliorerait probablement la sensibilité de la découverte du motif.

Dans la plupart des cas, les valeurs précédentes ont été sélectionnées pour être non informatives en raison du manque de connaissances. Une autre approche consisterait à intégrer les informations disponibles sur les motifs connus lors de la sélection des priorités pour les PWM.

Les résultats présentés dans cette thèse consistaient en la sortie de appliquer notre méthodologie à la bactérie (*Listeria monocytogenes*) qui était au centre du projet européen (Liste_MAPS). Une direction évidente pour les travaux futurs serait d'appliquer notre méthode sur d'autres ensembles de données. Pour nommer quelques candidats, textit *Bacillus subtilis*, *Staphylococcus aureus* et textit *Flavobacterium psychrophilum*, sont des bactéries sur lesquelles une grande des ensembles de données d'expression de qualité ont été collectés à la suite de les collaborations entre MaIAGE et les laboratoires de biologie humide.

L'algorithme de mise à jour des paramètres du modèle est écrit en C++ langue et il a été téléchargé en ligne avec un accès ouvert, un L'orientation future possible consiste à créer un site Web dédié offrant une interface plus conviviale pour l'outil développé. Parmi les défis, est la durée d'exécution de l'algorithme (quelques semaines) qui peut ne pas être

être attendu d'un utilisateur d'une interface Web.

La façon dont nous avons étendu le modèle de régression probit pour binariser le vecteur de prédicteurs et de prendre en compte les arbres est, à notre connaissance, une nouvelle et pourrait être généralisé à d'autres applications. Un avantage, comme il a été expliqué au chapitre 3 est qu'il est possible de utiliser ce modèle même lorsque la dimension des données utilisée comme prédicteurs est relativement élevé.

Dans deux dimensions, les cinq paramètres du modèle probit étendu peuvent décrire librement la position des commutateurs sur chaque axe, mais seuls trois paramètres sont utilisés pour décrire la probabilité d'occurrence de motif dans les quatre régions définies par les commutateurs. En effet, lorsque la dimension d augmente le nombre (2^d) de régions délimitées par les commutateurs augmente de façon exponentielle, mais le nombre $(1 + 2d)$ de paramètres n'augmente que de manière linéaire. Cela permet d'utiliser ce modèle même lorsque la dimension est relativement élevée.

Bibliography

- F Abram, E Starr, Kimon-Andreas G Karatzas, Ksenia Matlawska-Wasowska, A Boyd, M Wiedmann, KJ Boor, D Connally, and CP O’Byrne. Identification of components of the sigma b regulon in listeria monocytogenes that contribute to acid and salt tolerance. *Applied and environmental microbiology*, 74(22):6848–6858, 2008.
- James H Albert and Siddhartha Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American statistical Association*, 88(422):669–679, 1993.
- Uri Alon. *An introduction to systems biology: design principles of biological circuits*. Chapman and Hall/CRC, 2006.
- Uri Alon. Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, 8(6):450, 2007.
- Orly Alter, Patrick O Brown, and David Botstein. Singular value decomposition for genome-wide expression data processing and modeling. *Proceedings of the National Academy of Sciences*, 97(18):10101–10106, 2000.
- Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. An introduction to mcmc for machine learning. *Machine learning*, 50(1-2):5–43, 2003.
- Safia Arous, Carmen Buchrieser, Patrice Folio, Philippe Glaser, Abdelkader Namane, Michel Hebraud, and Yann Hechard. Global analysis of gene expression in an rpon mutant of listeria monocytogenes. *Microbiology*, 150(5):1581–1590, 2004.
- M Madan Babu and Sarah A Teichmann. Functional determinants of transcription factors in escherichia coli: protein families and binding sites. *TRENDS in Genetics*, 19(2):75–79, 2003.
- Timothy L Bailey and Charles Elkan. Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Machine learning*, 21(1-2):51–80, 1995a.

- Timothy L Bailey and Charles Elkan. The value of prior knowledge in discovering motifs with meme. In *Ismb*, volume 3, pages 21–29, 1995b.
- Timothy L Bailey, Charles Elkan, et al. Fitting a mixture model by expectation maximization to discover motifs in bipolymers. 1994.
- Enrique Balleza, Lucia N Lopez-Bojorquez, Agustino Martínez-Antonio, Osbaldo Resendis-Antonio, Irma Lozada-Chávez, Yalbi I Balderas-Martínez, Sergio Encarnación, and Julio Collado-Vides. Regulation by transcription factors in bacteria: beyond description. *FEMS microbiology reviews*, 33(1):133–151, 2008.
- Christophe Bécavin, Mikael Koutero, Nicolas Tchitchek, Franck Cerutti, Pierre Lechat, Nicolas Maillet, Claire Hoede, Hélène Chiapello, Christine Gaspin, and Pascale Cossart. Listeriomics: an interactive web platform for systems biology of listeria. *MSystems*, 2(2):e00186–16, 2017.
- I Ben-Gal, Ayala Shani, André Gohr, Jan Grau, S Arviv, Armin Shmilovici, Stefan Posch, and Ivo Grosse. Identification of transcription factor binding sites with variable-order bayesian networks. *Bioinformatics*, 21(11):2657–2666, 2005.
- Mathieu Blanchette and Martin Tompa. Discovery of regulatory elements by a computational method for phylogenetic footprinting. *Genome research*, 12(5):739–748, 2002.
- Monica K Borucki, Jason D Peppin, David White, Frank Loge, and Douglas R Call. Variation in biofilm formation among strains of listeria monocytogenes. *Applied and environmental microbiology*, 69(12):7336–7342, 2003.
- Craig Boutilier, Nir Friedman, Moises Goldszmidt, and Daphne Koller. Context-specific independence in bayesian networks. In *Proceedings of the Twelfth international conference on Uncertainty in artificial intelligence*, pages 115–123. Morgan Kaufmann Publishers Inc., 1996.
- Douglas F Browning and Stephen JW Busby. The regulation of bacterial transcription initiation. *Nature Reviews Microbiology*, 2(1):57, 2004.
- Peter Bühlmann, Abraham J Wyner, et al. Variable length markov chains. *The Annals of Statistics*, 27(2):480–513, 1999.
- Harmen J Bussemaker, Hao Li, and Eric D Siggia. Regulatory element detection using correlation with expression. *Nature genetics*, 27(2):167, 2001.

- C Steven Carmack, Lee Ann McCue, Lee A Newberg, and Charles E Lawrence. Phylscan: identification of transcription factor binding sites using cross-species evidence. *Algorithms for Molecular Biology*, 2(1):1, 2007.
- Anne-Sophie Carpentier, Alessandra Riva, Pierre Tisseur, Gilles Didier, and Alain Hénaut. The operons, a criterion to compare the reliability of transcriptome analysis tools: Ica is more reliable than anova, pls and pca. *Computational biology and chemistry*, 28(1): 3–10, 2004.
- Soraya Chaturongakul, Sarita Raengpradub, M Elizabeth Palmer, Teresa M Bergholz, Renato H Orsi, Yuewei Hu, Juliane Ollinger, Martin Wiedmann, and Kathryn J Boor. Transcriptomic and phenotypic analyses identify coregulated, overlapping regulons among prfa, ctrs, hrca, and the alternative sigma factors σ_b , σ_c , σ_h , and σ_l in listeria monocytogenes. *Applied and environmental microbiology*, 77(1):187–200, 2011.
- Xi Chen, Han Xu, Ping Yuan, Fang Fang, Mikael Huss, Vinsensius B Vega, Eleanor Wong, Yuriy L Orlov, Weiwei Zhang, Jianming Jiang, et al. Integration of external signaling pathways with the core transcriptional network in embryonic stem cells. *Cell*, 133(6): 1106–1117, 2008.
- Paul Cliften, Priya Sudarsanam, Ashwin Desikan, Lucinda Fulton, Bob Fulton, John Majors, Robert Waterston, Barak A Cohen, and Mark Johnston. Finding functional features in saccharomyces genomes by phylogenetic footprinting. *science*, 301(5629):71–76, 2003.
- Paul F Cliften, LaDeana W Hillier, Lucinda Fulton, Tina Graves, Tracie Miner, Warren R Gish, Robert H Waterston, and Mark Johnston. Surveying saccharomyces genomes to identify functional elements by comparative dna sequence analysis. *Genome research*, 11(7):1175–1186, 2001.
- Julio Collado-Vides, Boris Magasanik, and Jay D Gralla. Control site location and transcriptional regulation in escherichia coli. *Microbiological reviews*, 55(3):371–394, 1991.
- Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.
- Modan K Das and Ho-Kwok Dai. A survey of dna motif finding algorithms. In *BMC bioinformatics*, volume 8, page S21. BioMed Central, 2007.

- Aitor de las Heras, Robert J Cain, Magdalena K Bielecka, and Jose A Vazquez-Boland. Regulation of listeria virulence: Prfa master and commander. *Current opinion in microbiology*, 14(2):118–127, 2011.
- Janos Demeter, Catherine Beauheim, Jeremy Gollub, Tina Hernandez-Boussard, Heng Jin, Donald Maier, John C Matese, Michael Nitzberg, Farrell Wymore, Zachariah K Zachariah, et al. The stanford microarray database: implementation of new analysis tools and open source release of software. *Nucleic acids research*, 35(suppl_1):D766–D770, 2006.
- Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.
- Olivier Elemento, Noam Slonim, and Saeed Tavazoie. A universal framework for regulatory element discovery across all genomes and data types. *Molecular cell*, 28(2):337–350, 2007.
- Laurence Ettwiller, John Buswell, Erbay Yigit, and Ira Schildkraut. A novel enrichment strategy reveals unprecedented number of novel transcription start sites at single base resolution in a model prokaryote and the gut microbiome. *Bmc Genomics*, 17(1):199, 2016.
- Daphne Ezer, Nicolae Radu Zabet, and Boris Adryan. Homotypic clusters of transcription factor binding sites: A model system for understanding the physical mechanics of gene expression. *Computational and structural biotechnology journal*, 10(17):63–69, 2014.
- Jeremiah J Faith, Michael E Driscoll, Vincent A Fusaro, Elissa J Cosgrove, Boris Hayete, Frank S Juhn, Stephen J Schneider, and Timothy S Gardner. Many microbe microarrays database: uniformly normalized affymetrix compendia with structured experimental metadata. *Nucleic acids research*, 36(suppl_1):D866–D870, 2007.
- Adriana Ferreira, David Sue, Conor P O’byrne, and Kathryn J Boor. Role of listeria monocytogenes σ^b in survival of lethal acidic conditions and in the acquired acid tolerance response. *Applied and environmental microbiology*, 69(5):2692–2698, 2003.
- Barrett C Foat, Alexandre V Morozov, and Harmen J Bussemaker. Statistical mechanical modeling of genome-wide transcription factor occupancy data by matrixreduce. *Bioinformatics*, 22(14):e141–e149, 2006.

- Nir Friedman and Moises Goldszmidt. Learning bayesian networks with local structure. In *Learning in graphical models*, pages 421–459. Springer, 1998.
- Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe’er. Using bayesian networks to analyze expression data. *Journal of computational biology*, 7(3-4):601–620, 2000.
- Frederike Fritsch, Norman Mauder, Tatjana Williams, Julia Weiser, Markus Oberle, and Dagmar Beier. The cell envelope stress response mediated by the liafsrlm three-component system of listeria monocytogenes is controlled via the phosphatase activity of the bifunctional histidine kinase liaslm. *Microbiology*, 157(2):373–386, 2011.
- Cormac GM Gahan and Colin Hill. Listeria monocytogenes: survival and adaptation in the gastrointestinal tract. *Frontiers in cellular and infection microbiology*, 4:9, 2014.
- Alan E Gelfand and Adrian FM Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410):398–409, 1990.
- Peter J Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732, 1995.
- Tanja M Gruber and Carol A Gross. Multiple sigma subunits and the partitioning of bacterial transcription space. *Annual Reviews in Microbiology*, 57(1):441–466, 2003.
- Rosa Maria Gutierrez-Rios, David A Rosenblueth, Jose Antonio Loza, Araceli M Huerta, Jeremy D Glasner, Fred R Blattner, and Julio Collado-Vides. Regulatory network of escherichia coli: consistency between literature knowledge and microarray profiles. *Genome research*, 13(11):2435–2443, 2003.
- David Heckerman, Dan Geiger, and David M Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243, 1995.
- John D Heimann. The extracytoplasmic function (ecf) sigma factors. 2002.
- Rutger Hermsen, Sander Tans, and Pieter Rein Ten Wolde. Transcriptional regulation by competing transcription factor modules. *PLoS Computational Biology*, 2(12):e164, 2006.
- Christopher F Higgins, Robert S McLaren, and Sarah F Newbury. Repetitive extragenic palindromic sequences, mrna stability and gene expression: evolution by gene conversion?—a review. *Gene*, 72(1-2):3–14, 1988.

- Joshua WK Ho, Eric Bishop, Peter V Karchenko, Nicolas Nègre, Kevin P White, and Peter J Park. Chip-chip versus chip-seq: lessons for experimental design and data analysis. *BMC genomics*, 12(1):134, 2011.
- Oliver Hobert. Gene regulation by transcription factors and micrnas. *Science*, 319(5871):1785–1786, 2008.
- Aapo Hyvarinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE transactions on Neural Networks*, 10(3):626–634, 1999.
- Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000.
- Hisaki Ikebata and Ryo Yoshida. Repulsive parallel mcmc algorithm for discovering diverse motifs from large sequence sets. *Bioinformatics*, 31(10):1561–1568, 2015.
- Irnov Irnov, Cynthia M Sharma, Jörg Vogel, and Wade C Winkler. Identification of regulatory rnas in bacillus subtilis. *Nucleic acids research*, 38(19):6637–6651, 2010.
- Raja Jothi, Suresh Cuddapah, Artem Barski, Kairong Cui, and Keji Zhao. Genome-wide identification of in vivo protein–dna binding sites from chip-seq data. *Nucleic acids research*, 36(16):5221, 2008.
- Ulykbek Kairov, Laura Cantini, Alessandro Greco, Askhat Molkenov, Urszula Czerwinska, Emmanuel Barillot, and Andrei Zinovyev. Determining the optimal number of independent components for reproducible transcriptomic data analysis. *BMC genomics*, 18(1):712, 2017.
- Minoru Kanehisa, Michihiro Araki, Susumu Goto, Masahiro Hattori, Mika Hirakawa, Masumi Itoh, Toshiaki Katayama, Shuichi Kawashima, Shujiro Okuda, Toshiaki Tokimatsu, et al. Kegg for linking genomes to life and the environment. *Nucleic acids research*, 36(suppl_1):D480–D484, 2007.
- Mark J Kazmierczak, Martin Wiedmann, and Kathryn J Boor. Alternative sigma factors and their roles in bacterial virulence. *Microbiology and Molecular Biology Reviews*, 69(4):527–543, 2005.
- Ingrid M Keseler, Julio Collado-Vides, Socorro Gama-Castro, John Ingraham, Suzanne Paley, Ian T Paulsen, Martín Peralta-Gil, and Peter D Karp. Ecocyc: a comprehensive

- database resource for escherichia coli. *Nucleic acids research*, 33(suppl_1):D334–D337, 2005.
- Lawrence J Korn, Cary L Queen, and Mark N Wegman. Computer analysis of nucleic acid regulatory sequences. *Proceedings of the National Academy of Sciences*, 74(10):4401–4405, 1977.
- Sheryl R Krig, Victor X Jin, Mark C Bieda, Henriette O’Geen, Paul Yaswen, Roland Green, and Peggy J Farnham. Identification of genes directly regulated by the oncogene znf217 using chip-chip assays. *Journal of biological chemistry*, 2007.
- Mathieu Lajoie, Olivier Gascuel, Vincent Lefort, and Laurent Bréhélin. Computational discovery of regulatory elements in a continuous expression space. *Genome biology*, 13(11):R109, 2012.
- Charles E Lawrence and Andrew A Reilly. An expectation maximization (em) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins: Structure, Function, and Bioinformatics*, 7(1):41–51, 1990.
- Charles E Lawrence, Stephen F Altschul, John C Wootton, Mark S Boguski, Andrew F Neuwald, and Jun S Liu. A gibbs sampler for the detection of subtle motifs in multiple sequences. In *1994 Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences*, 1994.
- Tong Ihn Lee, Nicola J Rinaldi, François Robert, Duncan T Odom, Ziv Bar-Joseph, Georg K Gerber, Nancy M Hannett, Christopher T Harbison, Craig M Thompson, Itamar Simon, et al. Transcriptional regulatory networks in *saccharomyces cerevisiae*. *science*, 298(5594):799–804, 2002.
- Wolfram Liebermeister. Linear modes of gene expression determined by independent component analysis. *Bioinformatics*, 18(1):51–60, 2002.
- Ann A Lin, Don Walthers, and Peter Zuber. Residue substitutions near the redox center of *bacillus subtilis* spx affect rna polymerase interaction, redox control and spx-dna contact at a conserved cis-acting element. *Journal of bacteriology*, pages JB–00645, 2013.
- Yuin-Han Loh, Qiang Wu, Joon-Lin Chew, Vinsensius B Vega, Weiwei Zhang, Xi Chen, Guillaume Bourque, Joshy George, Bernard Leong, Jun Liu, et al. The oct4 and nanog

- transcription network regulates pluripotency in mouse embryonic stem cells. *Nature genetics*, 38(4):431, 2006.
- Ulrike Mäder, Pierre Nicolas, Maren Depke, Jan Pané-Farré, Michel Debarbouille, Magdalena M van der Kooi-Pol, Cyprien Guérin, Sandra Dérozier, Aurelia Hiron, Hanne Jarmer, et al. Staphylococcus aureus transcriptome architecture: from laboratory to infection-mimicking conditions. *PLoS genetics*, 12(4):e1005962, 2016.
- Hiroto Maeda, Nobuyuki Fujita, and Akira Ishihama. Competition among seven escherichia coli σ subunits: relative binding affinities to the core rna polymerase. *Nucleic acids research*, 28(18):3497–3503, 2000.
- Pierre Mandin, Hafida Fsihi, Olivier Dussurget, Massimo Vergassola, Eliane Milohanic, Alejandro Toledo-Arana, Iñigo Lasa, Jörgen Johansson, and Pascale Cossart. Virr, a response regulator critical for listeria monocytogenes virulence. *Molecular microbiology*, 57(5):1367–1380, 2005.
- Elaine R Mardis. Chip-seq: welcome to the new frontier. *Nature methods*, 4(8):613, 2007.
- Harley H McAdams and Adam Arkin. Simulation of prokaryotic genetic circuits. *Annual review of biophysics and biomolecular structure*, 27(1):199–224, 1998.
- Ugrappa Nagalakshmi, Zhong Wang, Karl Waern, Chong Shou, Debasish Raha, Mark Gerstein, and Michael Snyder. The transcriptional landscape of the yeast genome defined by rna sequencing. *Science*, 320(5881):1344–1349, 2008.
- Andrew F Neuwald, Jun S Liu, and Charles E Lawrence. Gibbs motif sampling: detection of bacterial outer membrane protein repeats. *Protein science*, 4(8):1618–1632, 1995.
- Pierre Nicolas, Ulrike Mäder, Etienne Dervyn, Tatiana Rochat, Aurélie Leduc, Nathalie Pigeonneau, Elena Bidnenko, Elodie Marchadier, Mark Hoebeke, Stéphane Aymerich, et al. Condition-dependent transcriptome reveals high-level regulatory architecture in bacillus subtilis. *Science*, 335(6072):1103–1106, 2012.
- Pavel S Novichkov, Alexey E Kazakov, Dmitry A Ravcheev, Semen A Leyn, Galina Y Kovaleva, Roman A Sutormin, Marat D Kazanov, William Riehl, Adam P Arkin, Inna Dubchak, et al. Regprecise 3.0—a resource for genome-scale exploration of transcriptional regulation in bacteria. *BMC genomics*, 14(1):745, 2013.

- Mark SB Paget and John D Helmann. The σ 70 family of sigma factors. *Genome biology*, 4(1):203, 2003.
- M Elizabeth Palmer, Soraya Chaturongakul, Martin Wiedmann, and Kathryn J Boor. The *listeria monocytogenes* σ b regulon and its virulence-associated functions are inhibited by a small molecule. *MBio*, 2(6):e00241–11, 2011.
- Smitha Pillai and Srikumar P Chellappan. Chip on chip assays: genome-wide analysis of transcription factor binding and histone modifications. In *Chromatin Protocols*, pages 341–366. Springer, 2009.
- John Quackenbush. Microarray data normalization and transformation. *Nature genetics*, 32:496, 2002.
- Nikos B Reppas, Joseph T Wade, George M Church, and Kevin Struhl. The transition between transcriptional initiation and elongation in *e. coli* is highly variable and often rate limiting. *Molecular cell*, 24(5):747–757, 2006.
- Christian Robert. *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer Science & Business Media, 2007.
- Tatiana Rochat, Pierre Nicolas, Olivier Delumeau, Alžbeta Rabatinová, Jana Korelusova, Aurelie Leduc, Philippe Bessieres, Etienne Dervyn, Libor Krásný, and Philippe Noirot. Genome-wide identification of genes directly regulated by the pleiotropic transcription factor *spx* in *bacillus subtilis*. *Nucleic acids research*, 40(19):9571–9583, 2012.
- Dana Ron, Yoram Singer, and Naftali Tishby. The power of amnesia: Learning probabilistic automata with variable memory length. *Machine learning*, 25(2-3):117–149, 1996.
- Geir Kjetil Sandve and Finn Drabløs. A survey of motif discovery methods in an integrated framework. *Biology direct*, 1(1):11, 2006.
- Sophie Schbath. An overview on the distribution of word counts in markov chains. *Journal of Computational Biology*, 7(1-2):193–201, 2000.
- Mark Schena, Dari Shalon, Ronald W Davis, and Patrick O Brown. Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, 270(5235):467–470, 1995.

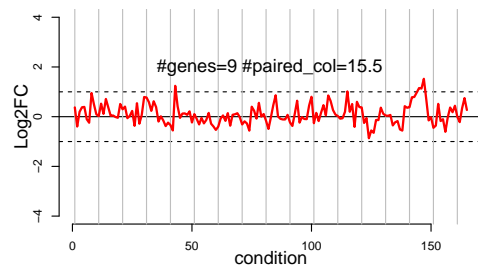
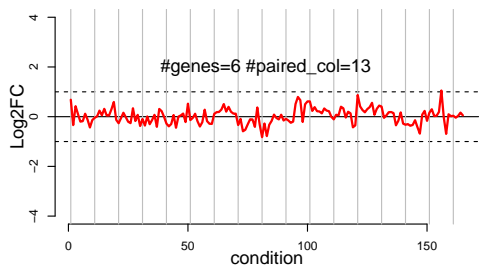
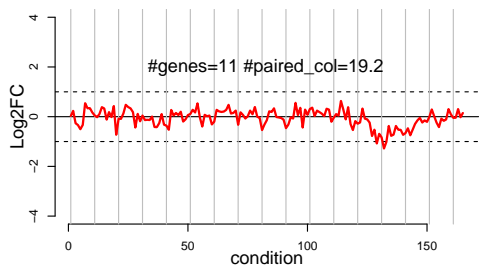
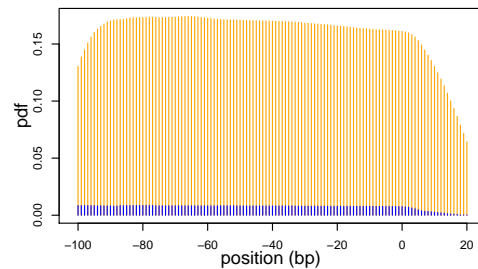
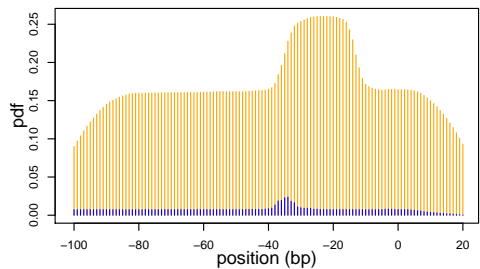
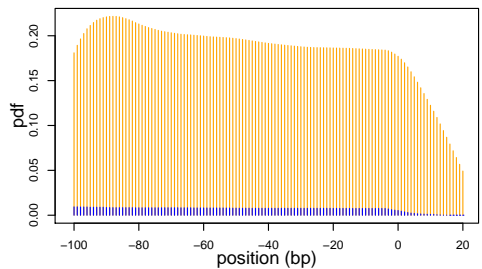
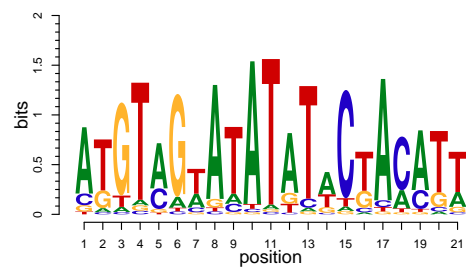
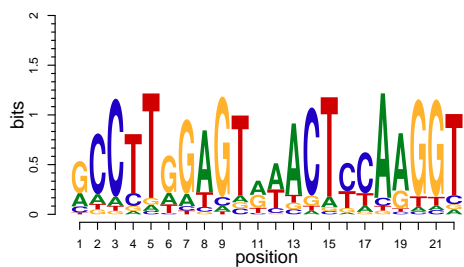
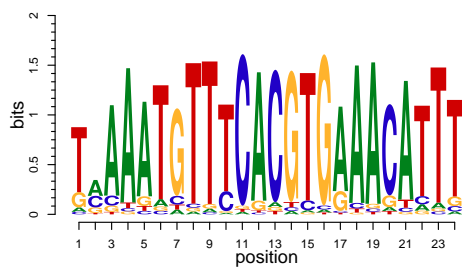
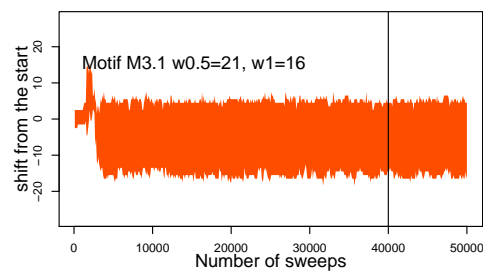
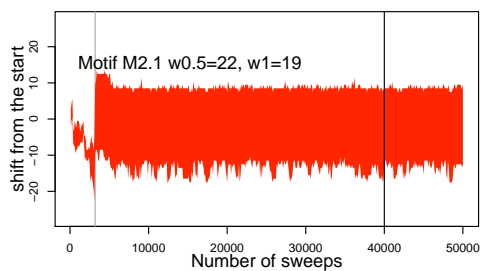
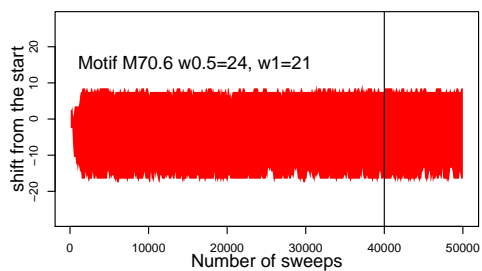
- Mariela Scortti, Héctor J Monzó, Lizeth Lacharme-Lora, Deborah A Lewis, and José A Vázquez-Boland. The prfa virulence regulon. *Microbes and Infection*, 9(10):1196–1207, 2007.
- Rahul Siddharthan. Dinucleotide weight matrices for predicting transcription factor binding sites: generalizing the position weight matrix. *PloS one*, 5(3):e9722, 2010.
- Matthew Slattery, Tianyin Zhou, Lin Yang, Ana Carolina Dantas Machado, Raluca Gordân, and Remo Rohs. Absence of a simple code: how transcription factors read the genome. *Trends in biochemical sciences*, 39(9):381–399, 2014.
- Noam Slonim, Gurinder Singh Atwal, Gašper Tkačik, and William Bialek. Information-based clustering. *Proceedings of the National Academy of Sciences*, 102(51):18297–18302, 2005.
- Robert R Sokal. A statistical method for evaluating systematic relationship. *University of Kansas science bulletin*, 28:1409–1438, 1958.
- Rotem Sorek and Pascale Cossart. Prokaryotic transcriptomics: a new view on regulation, physiology and pathogenicity. *Nature Reviews Genetics*, 11(1):9, 2010.
- D Thieffry and R Thomas. Qualitative analysis of gene networks. In *Biocomputing’98- Proceedings Of The Pacific Symposium*, page 77. World Scientific, 1997.
- Gert Thijs, Magali Lescot, Kathleen Marchal, Stephane Rombauts, Bart De Moor, Pierre Rouze, and Yves Moreau. A higher-order background model improves the detection of promoter regulatory elements by gibbs sampling. *Bioinformatics*, 17(12):1113–1122, 2001.
- Alejandro Toledo-Arana, Olivier Dussurget, Georgios Nikitas, Nina Sesto, Hélène Guet-Revillet, Damien Balestrino, Edmund Loh, Jonas Gripenland, Teresa Tiensuu, Karolis Vaitkevicius, et al. The listeria transcriptional landscape from saprophytism to virulence. *Nature*, 459(7249):950, 2009.
- Andreas Wagner. Genes regulated cooperatively by one or more transcription factors and their identification in whole eukaryotic genomes. *Bioinformatics (Oxford, England)*, 15(10):776–784, 1999.

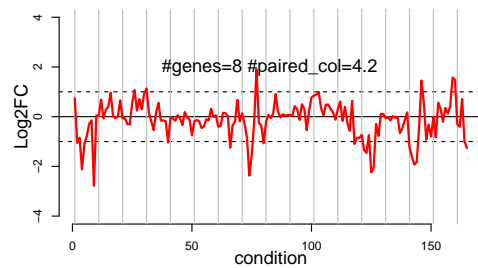
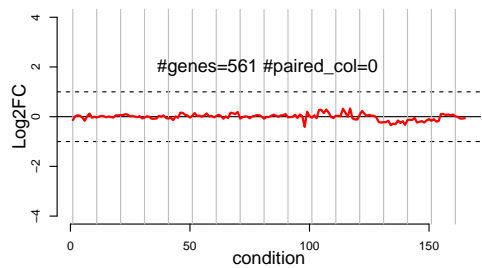
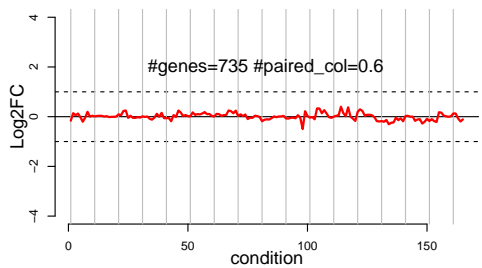
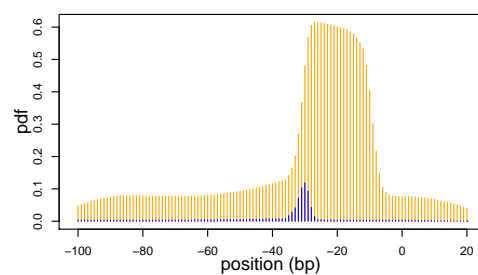
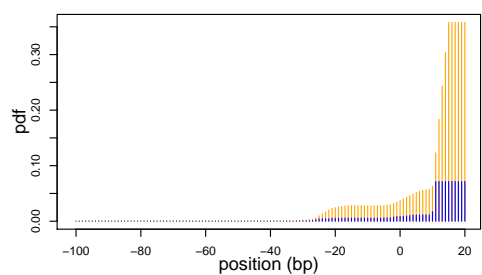
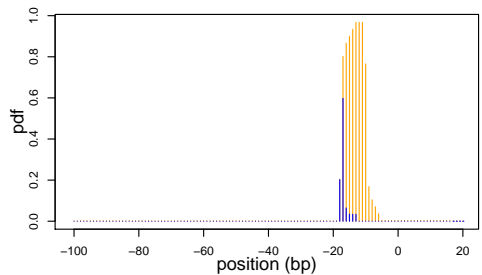
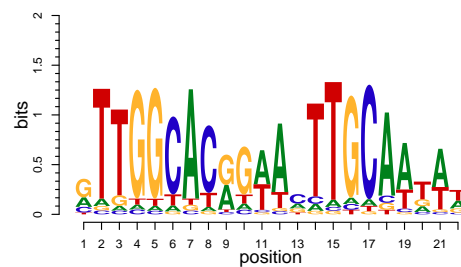
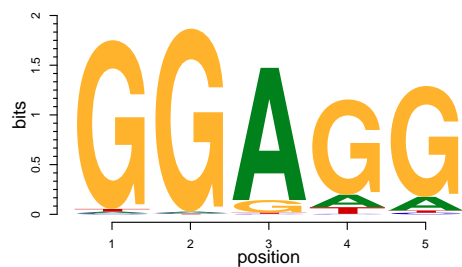
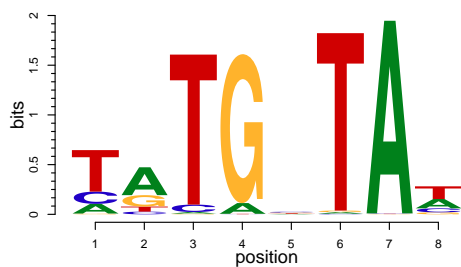
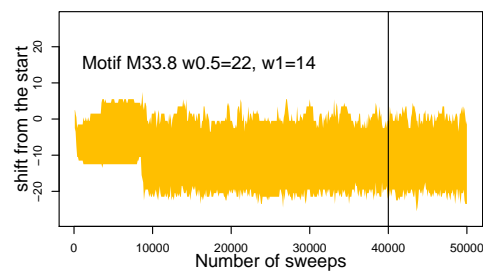
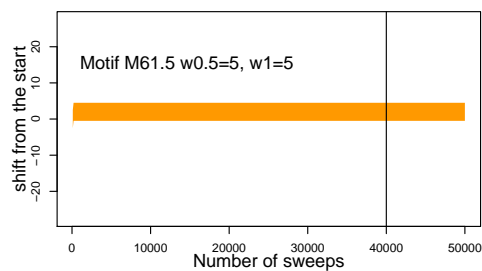
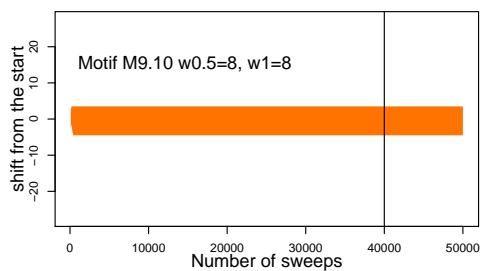
- Ting Wang and Gary D Stormo. Identifying the conserved network of cis-regulatory sites of a eukaryotic genome. *Proceedings of the National Academy of Sciences*, 102(48):17400–17405, 2005.
- Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.
- Matthew T Weirauch, Atina Cote, Raquel Norel, Matti Annala, Yue Zhao, Todd R Riley, Julio Saez-Rodriguez, Thomas Cokelaer, Anastasia Vedenko, Shaheynoor Talukder, et al. Evaluation of methods for modeling transcription factor sequence specificity. *Nature biotechnology*, 31(2):126, 2013.
- HJ Welshimer. Survival of listeria monocytogenes in soil. *Journal of Bacteriology*, 80(3):316, 1960.
- Aaron T Whiteley, Brittany R Ruhland, Mauna B Edrozo, and Michelle L Reniere. A redox-responsive transcription factor is critical for pathogenesis and aerobic growth of listeria monocytogenes. *Infection and immunity*, pages IAI–00978, 2017.
- Omri Wurtzel, Rajat Sapra, Feng Chen, Yiwen Zhu, Blake A Simmons, and Rotem Sorek. A single-base resolution map of an archaeal transcriptome. *Genome research*, 20(1):133–141, 2010.
- Omri Wurtzel, Nina Sesto, Jeff R Mellin, Iris Karunker, Sarit Edelheit, Christophe Bécavin, Cristel Archambaud, Pascale Cossart, and Rotem Sorek. Comparative transcriptomics of pathogenic and non-pathogenic listeria species. *Molecular systems biology*, 8(1):583, 2012.
- Jason A Young, Jeffery R Johnson, Chris Benner, S Frank Yan, Kaisheng Chen, Karine G Le Roch, Yingyao Zhou, and Elizabeth A Winzeler. In silico discovery of transcription regulatory elements in plasmodium falciparum. *BMC genomics*, 9(1):70, 2008.
- Federico Zambelli, Graziano Pesole, and Giulio Pavesi. Motif discovery and transcription factor binding sites before and after the next-generation sequencing era. *Briefings in bioinformatics*, 14(2):225–237, 2012.
- Karen I Zeller, XiaoDong Zhao, Charlie WH Lee, Kuo Ping Chiu, Fei Yao, Jason T Yustein, Hong Sain Ooi, Yuriy L Orlov, Atif Shahab, How Choong Yong, et al. Global mapping

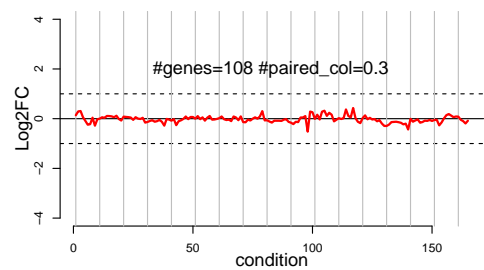
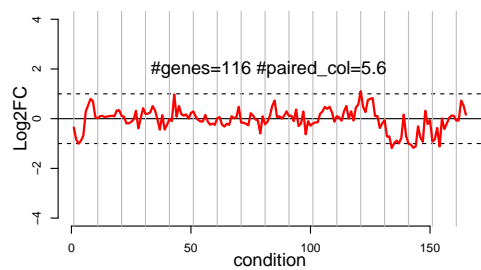
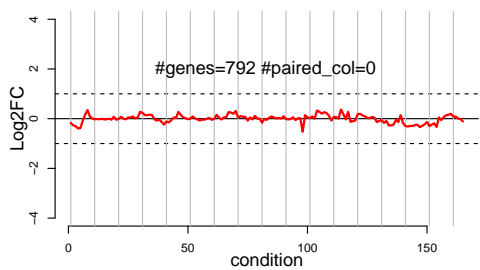
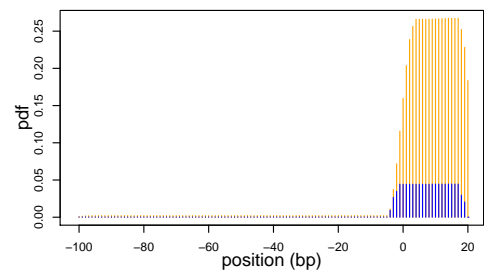
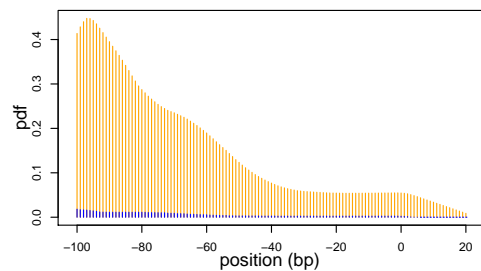
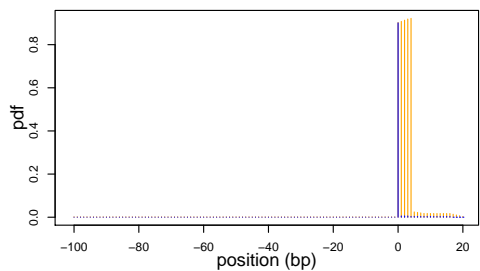
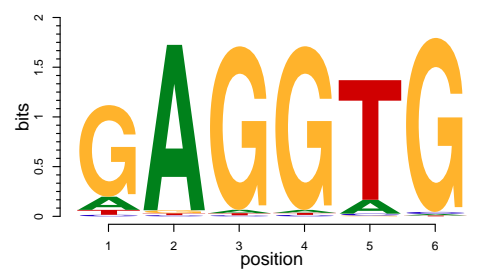
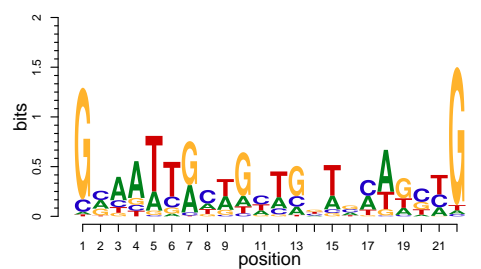
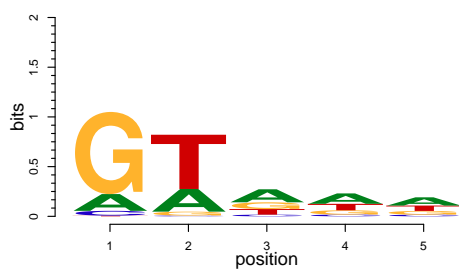
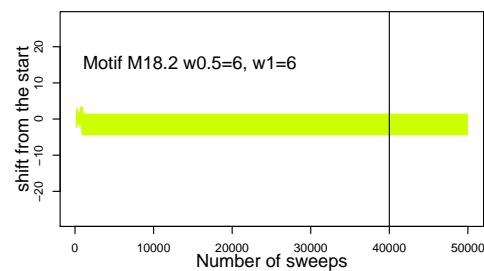
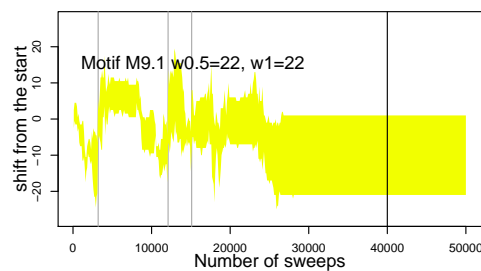
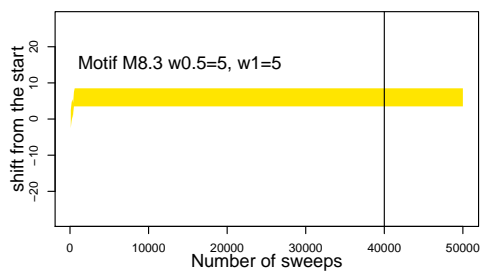
of c-myc binding sites and target gene networks in human b cells. *Proceedings of the National Academy of Sciences*, 103(47):17834–17839, 2006.

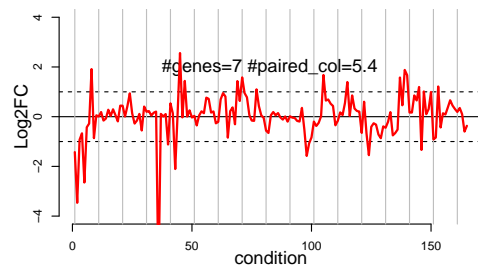
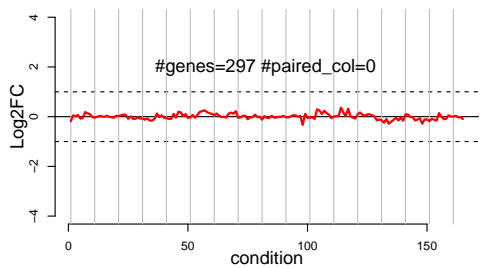
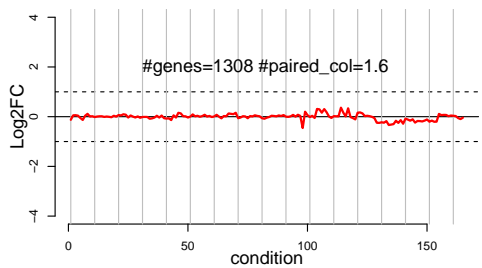
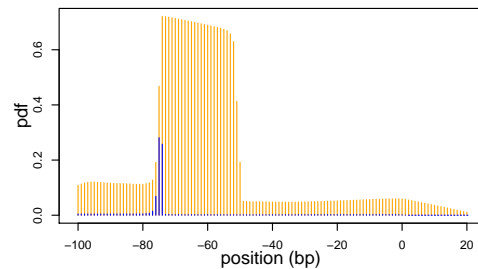
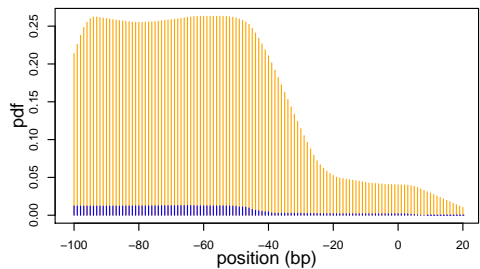
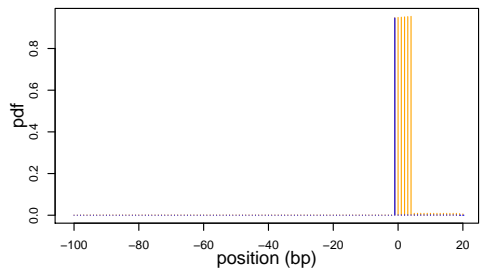
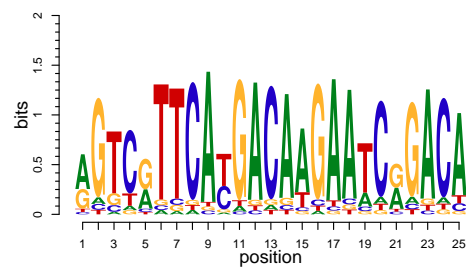
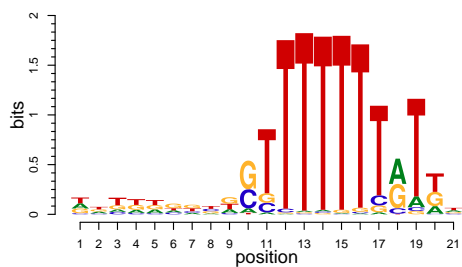
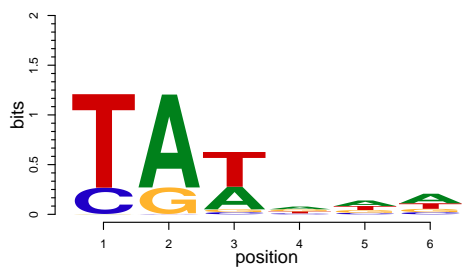
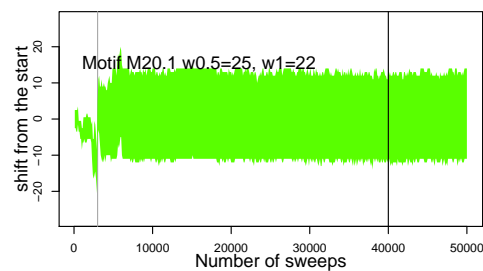
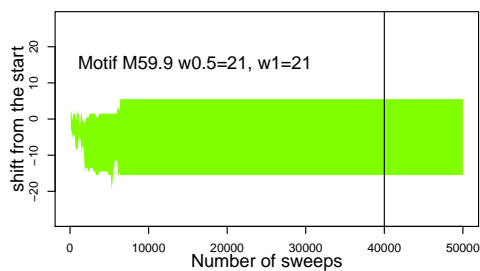
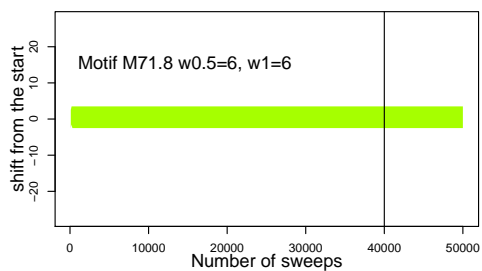
Supplementary materials

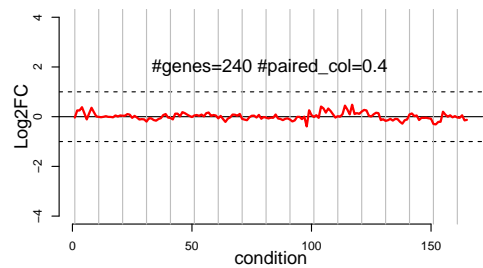
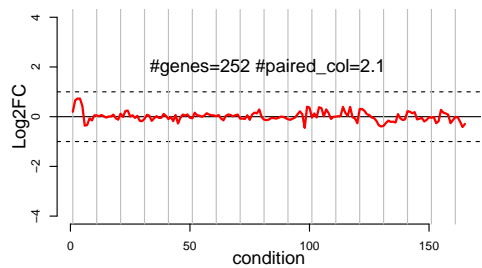
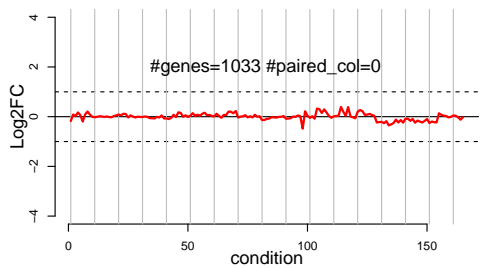
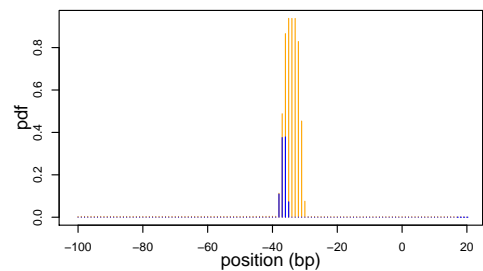
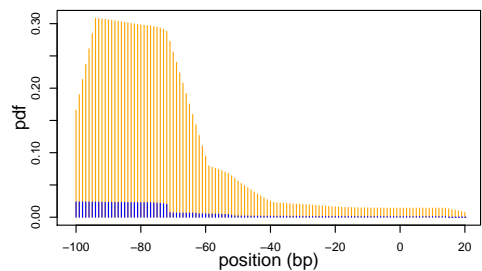
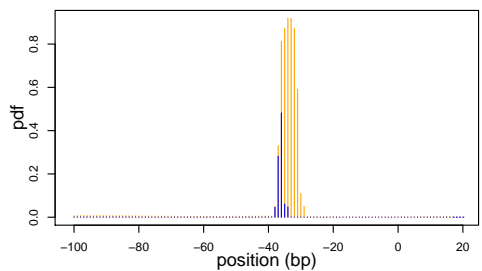
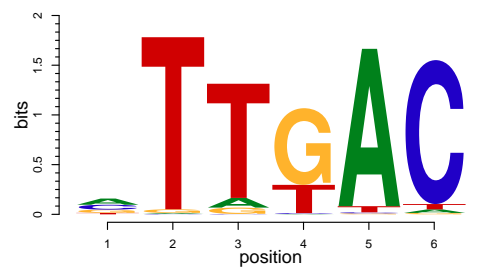
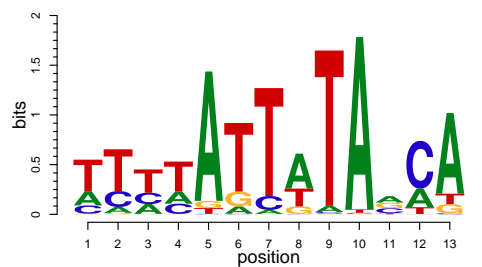
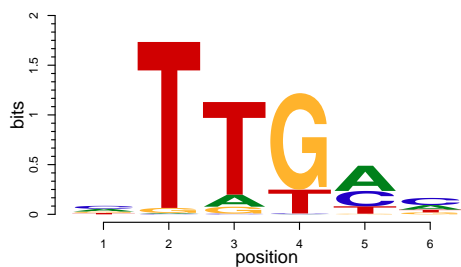
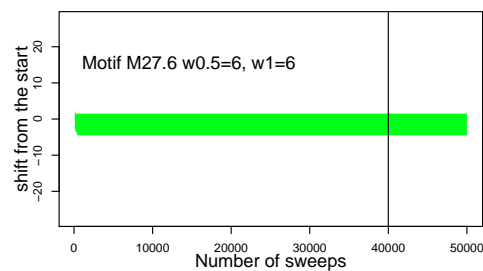
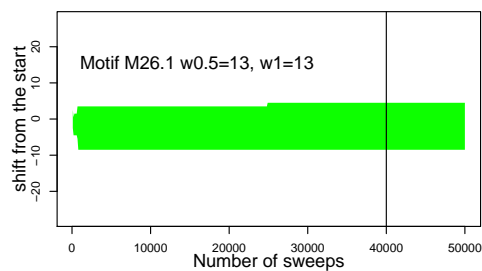
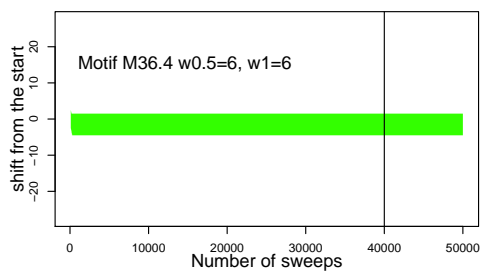
Supplementary file 1 (S1): Convergence plot, sequence logo, distance from TSS, and expression profile for the 40 discovered motifs.

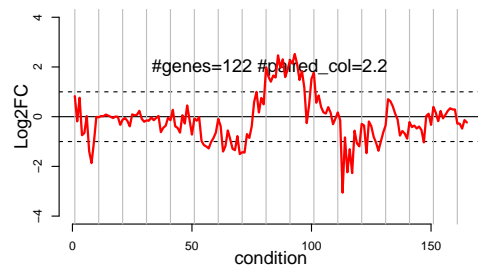
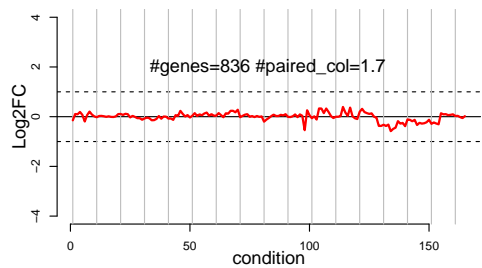
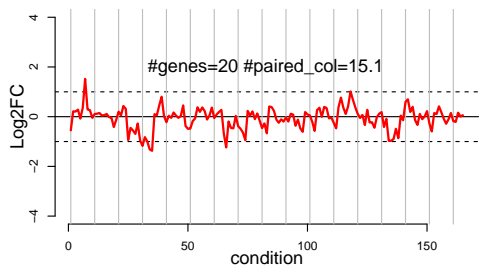
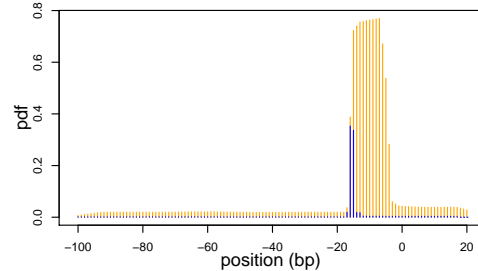
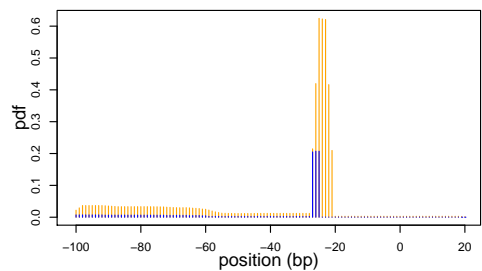
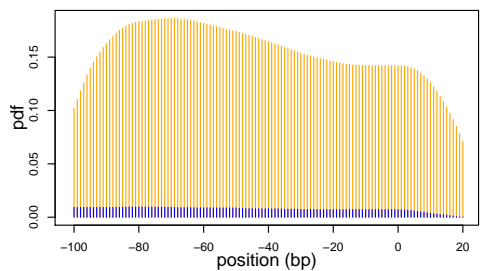
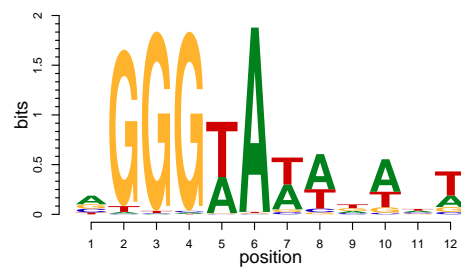
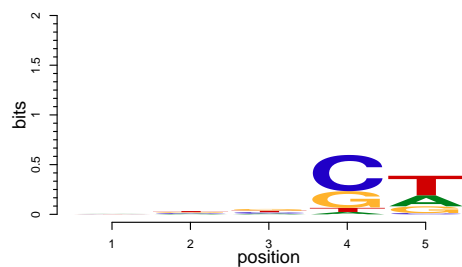
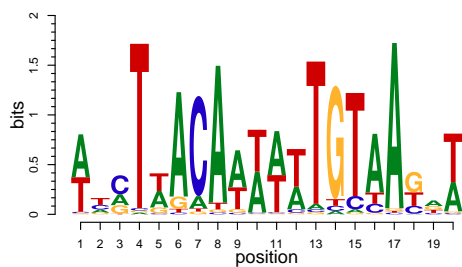
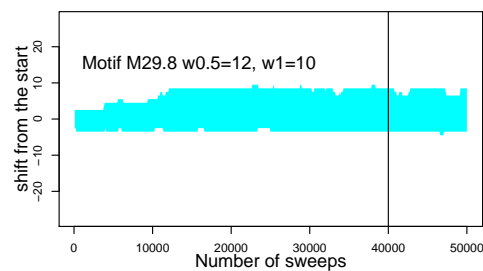
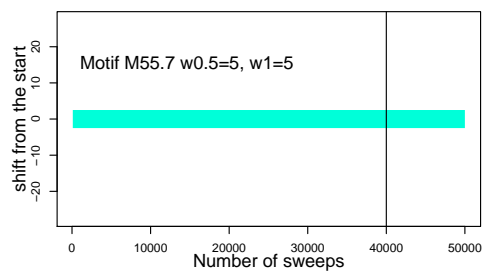
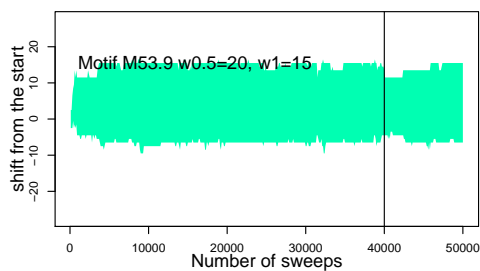


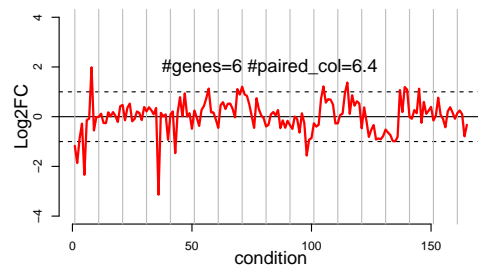
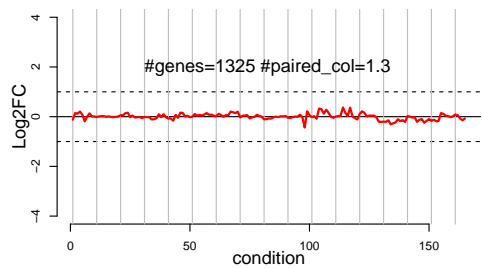
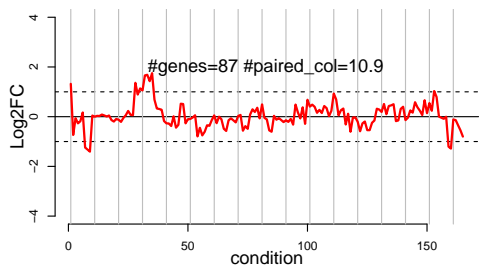
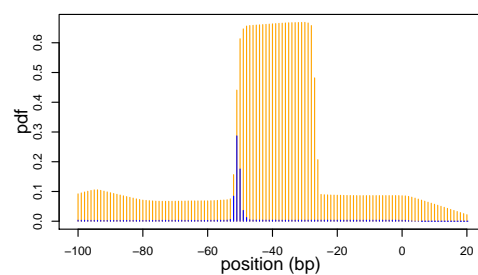
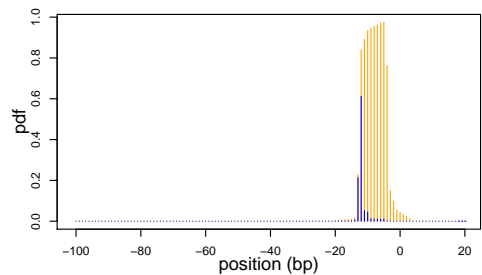
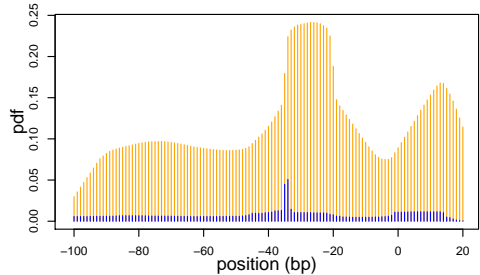
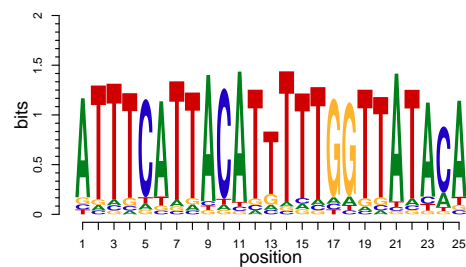
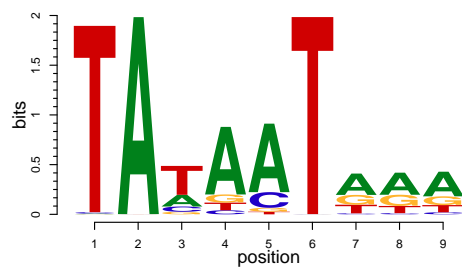
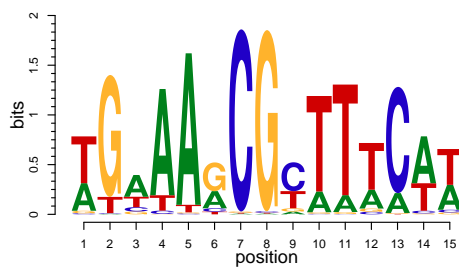
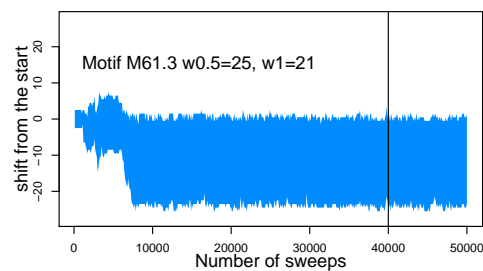
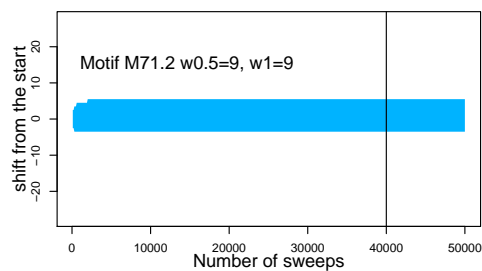
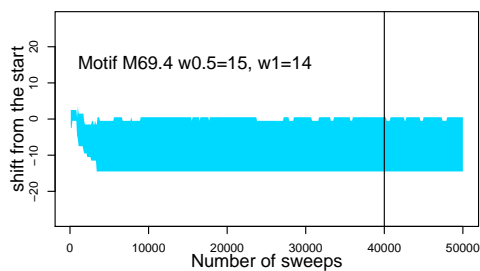


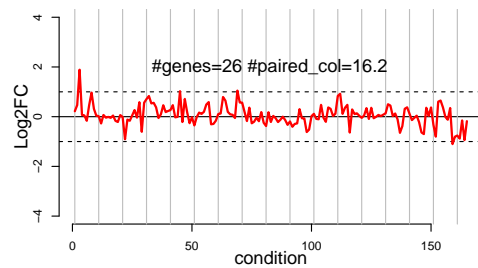
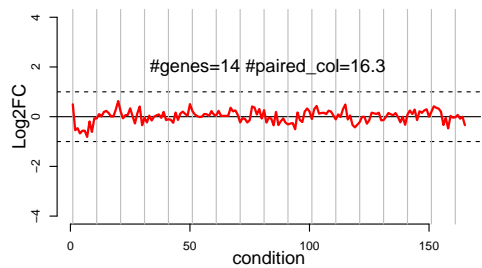
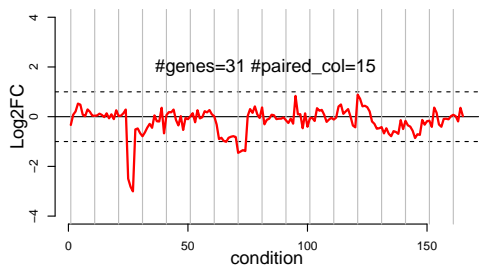
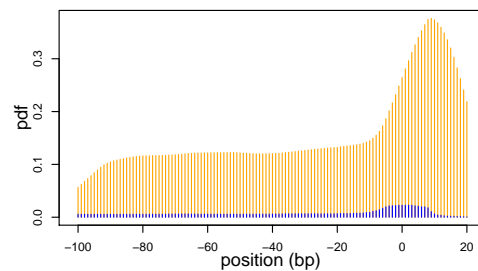
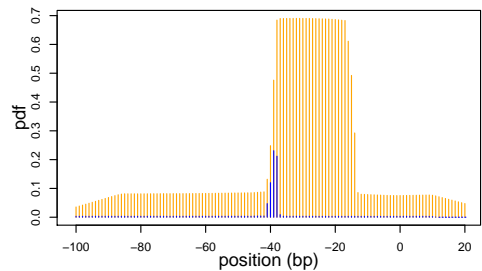
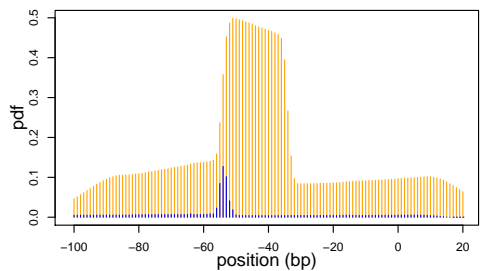
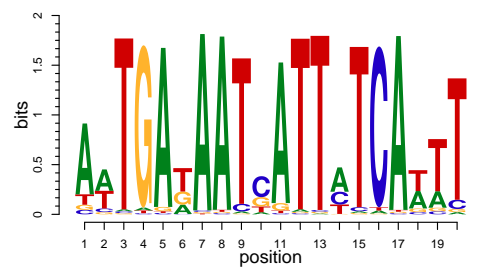
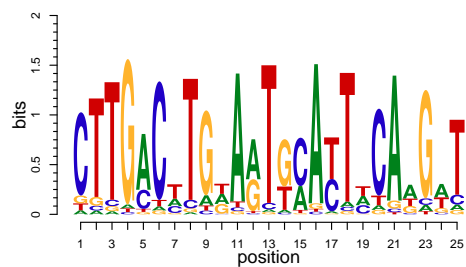
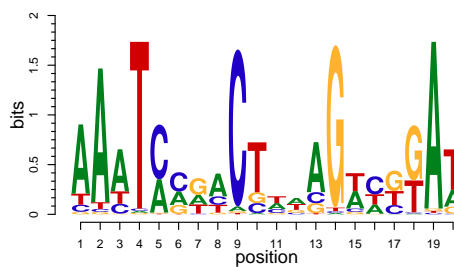
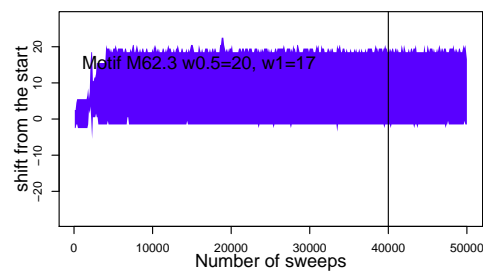
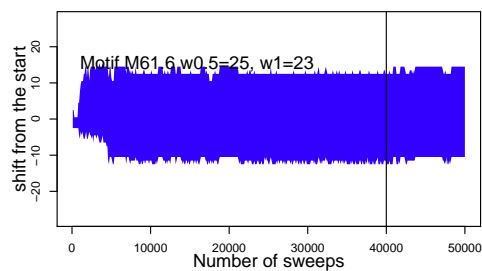
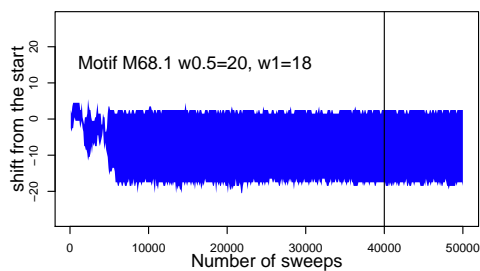


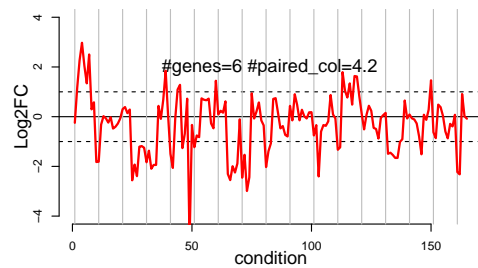
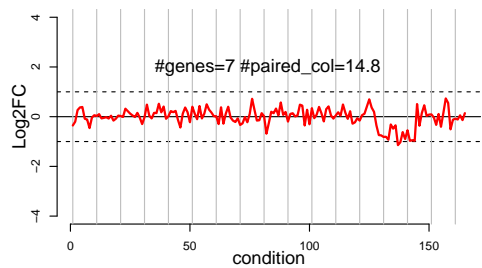
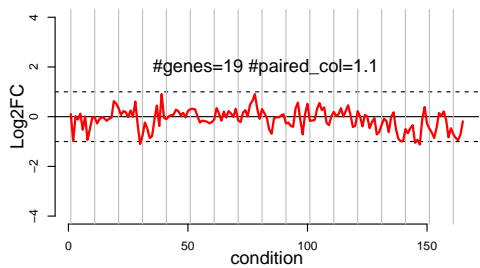
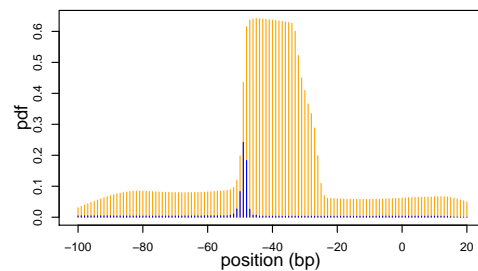
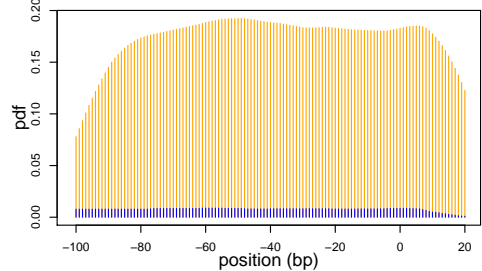
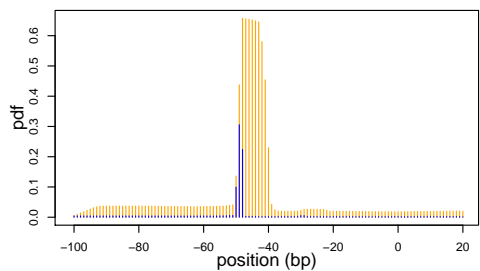
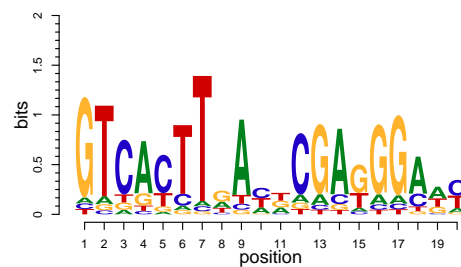
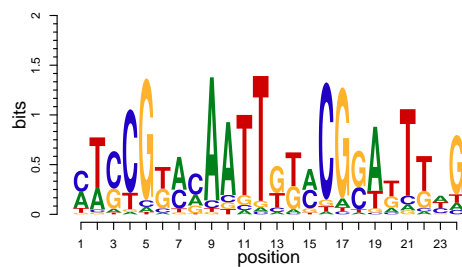
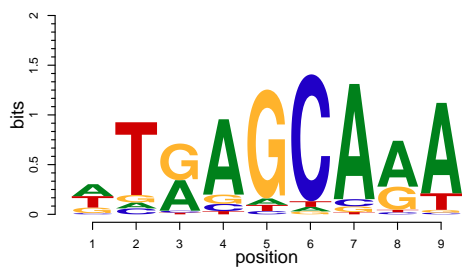
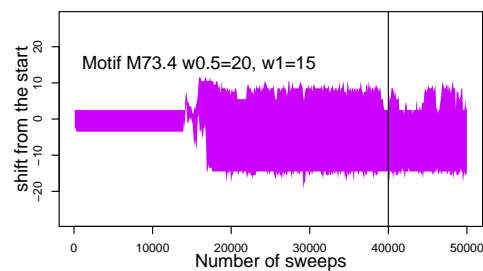
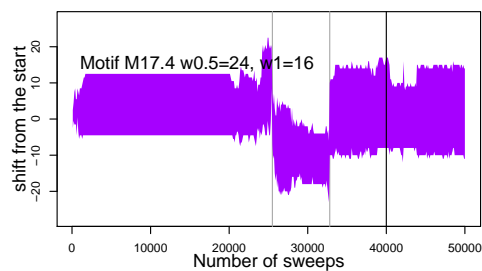
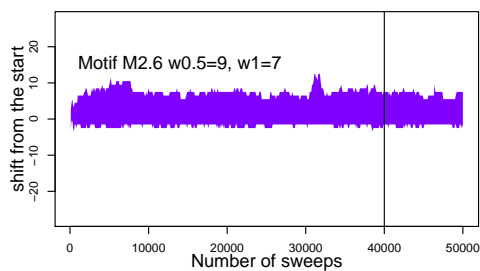


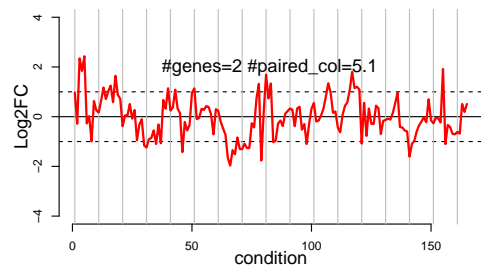
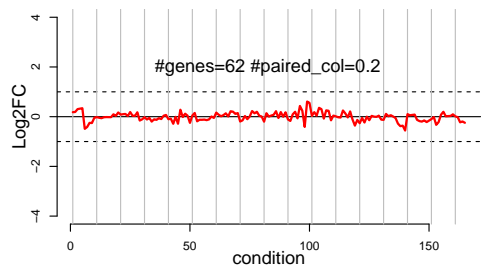
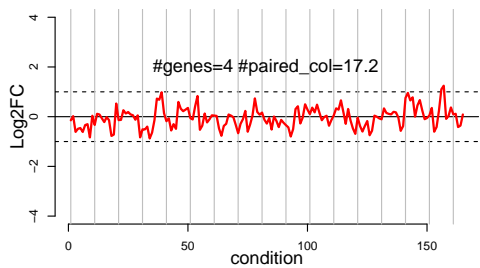
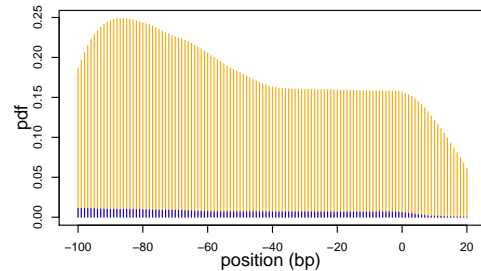
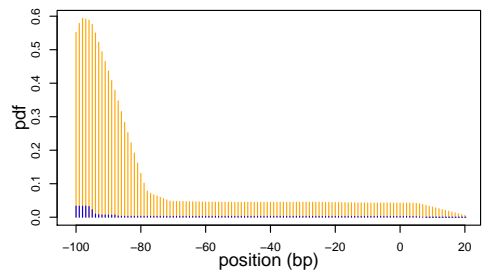
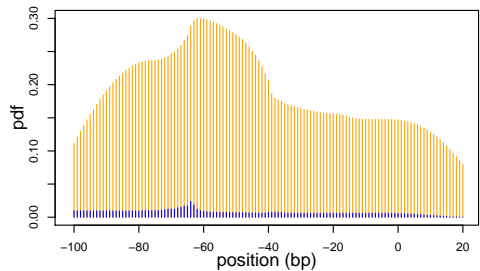
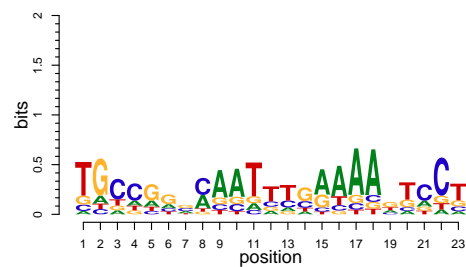
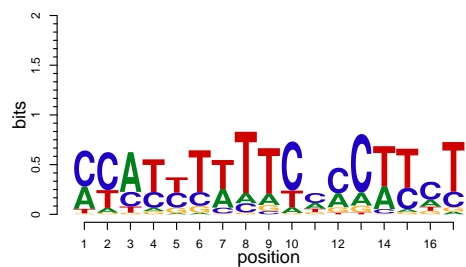
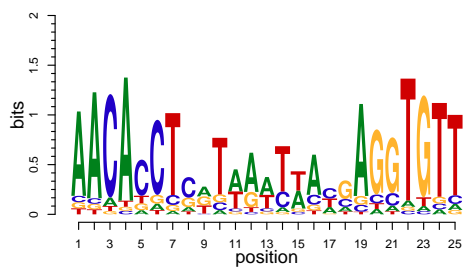
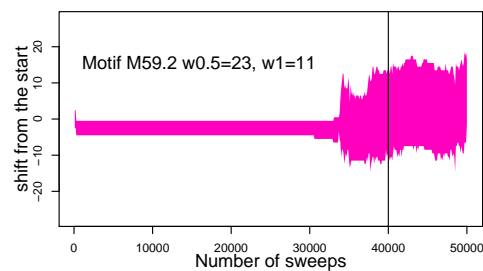
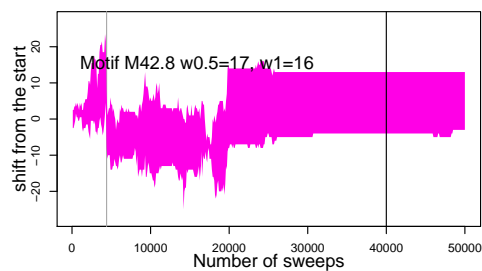
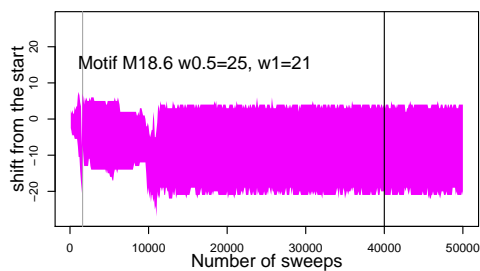


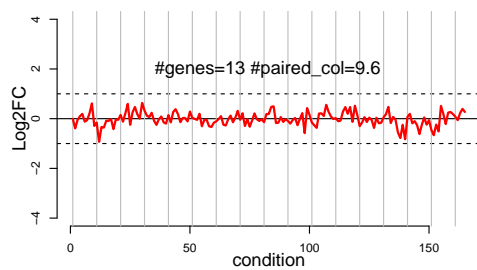
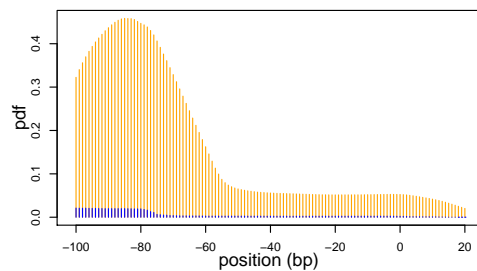
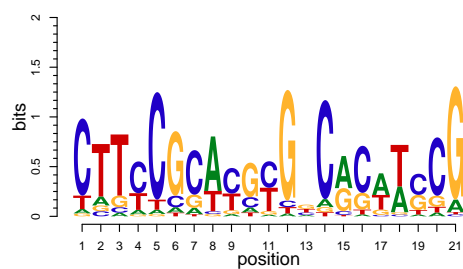
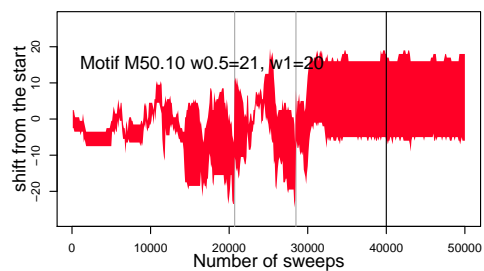












Titre : La construction du réseau de régulation transcriptionnelle

Mots clés : élément régulateur; profil d'expression; Facteur de transcription; construction; Biologie des systèmes

Résumé : Une part prépondérante de la régulation au niveau transcriptionnel passe par la modulation du taux d'initiation de la transcription. Chez les bactéries, l'initiation de la transcription implique la reconnaissance par le facteur sigma de l'ANR polymérase d'un motif de séquence particulier localisé approximativement 10 bp en amont du site d'initiation de la transcription (TSS). Elle est modulée par la fixation de facteurs de transcription qui reconnaissent d'autres motifs à proximité. La technologie RNA-Seq donne accès au répertoire des TSS et des unités de transcriptions et offre donc des perspectives renouvelées pour s'attaquer au problème de l'identification des motifs de fixation des facteurs de transcription. Ce travail de thèse vise à évaluer les outils existants et à développer de nouvelles méthodes pour la prédiction des sites de fixation des facteurs de transcription en combinant l'information des profils d'expression et des positions des TSS. Plusieurs approches fondées sur les modèles de matrices poids-position (PWM) vont être

explorées pour étendre le modèle de mélange classiquement utilisé en relâchant l'hypothèse selon laquelle les motifs correspondants aux différents sites de fixations apparaissent indépendamment dans les différentes régions promotrices. Dans les nouveaux modèles, nous prendrons explicitement en compte une probabilité supérieure d'apparition d'un même motif dans des promoteurs dont les profils d'activité sont similaires. Une attention particulière sera aussi portée à la position du motif par rapport au TSS et au site de fixation du facteur sigma. En parallèle des développements méthodologiques nous travaillerons aussi sur l'utilisation de ces approches pour reconstruire le réseau des régulations transcriptionnelles chez *L. monocytogenes* en s'appuyant sur les données de la littérature et du projet List.MAPS. Enfin, nous envisageons d'utiliser l'information sur le réseau de régulation pour étudier un point particulier qui serait pertinent pour le projet List.MAPS avec un modèle dédié.

Title : Transcriptional regulatory network construction

Keywords : regulatory element; Systems biology; expression profile; Transcription factor; constructions

Abstract : This PhD project takes place in List.MAPS, a Horizon 2020-funded Marie Curie Actions Innovative Training Network (ITN) with the goal of understanding of the ecology of *Listeria monocytogenes* through the combination of high throughput Epigenetics, Deep sequencing of transcripts, Proteomics, Bioinformatics, Mathematics and Microbiology. A central objective of the ITN is to decipher the mechanisms underlying adaptation and virulence of *L. monocytogenes* "from farm to fork". This PhD project (sub-project 9) aims to tackle the task of transcription regulatory network construction. A significant part of regulation at the transcriptional level is achieved by modulation of transcription initiation rate. In bacteria, transcription initiation relies on recognition of particular sequence motif by a Sigma-factor approximately 10 bp upstream of the transcription start site (TSS) and is modulated by the binding of transcription factors recognizing other sequence motifs located nearby. RNA-Seq transcriptomics provides direct information on the repertoire of TSSs and transcription units and thereby offers renewed perspectives to address the problem of transcription factor binding sites identification. The

goal of this PhD project is to assess existing tools and to develop new methods for prediction of TF binding sites by combining expression profiles and precise information on the location of the TSSs. Several approaches based on position weight matrix (PWM) models will be investigated to extend the classical mixture model by relaxing the hypothesis that motifs corresponding to different TF binding sites occur independently between TSS regions. In the new model, we will explicitly account for the increased probability of occurrence of a same motif in two promoters when their profiles of activity across conditions are similar. A particular attention will also be paid to the position of the motif with respect to the TSS and the sigma factor binding site. In parallel to the methodological developments we will also work on the use of these approaches to build the transcription regulatory network of *L. monocytogenes* based on data from the literature and from the List.MAPS project. Finally, we wish to use the information on the regulatory network to tackle a particular point relevant for the List.MAPS project using a dedicated model.

