Spatial and Temporal Coupling Models for the Discovery of Binding Events in ChIP-Seq Data

by

Georgios Papachristoudis

Bachelor of Science, Electrical and Computer Engineering, Aristotle University of Thessaloniki (2007)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2010

Author .....
Department of Electrical Engineering and Computer Science
January 26, 2010

Certified by .....
David K. Gifford
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by .....
Terry P. Orlando
Professor of Electrical Engineering and Computer Science
Chairman, Department Committee on Graduate Theses

Spatial and Temporal Coupling Models for the Discovery of Binding Events in ChIP-Seq Data

by

Georgios Papachristoudis

Submitted to the Department of Electrical Engineering and Computer Science
on January 26, 2010, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science and Engineering

Abstract

In this thesis, we will present two methods for identifying binding events in ChIP-Seq data. The motivation of this venture is to propose a complete read generating process under a probabilistic graphical model framework which will determine more accurately binding event locations and enforce alignment of events across conditions. More specifically, we will first propose the so-called *Spatial Coupling* method which exploits the relative positions of reads by assuming dependent assignment of events to close reads. Second, we will present the so-called *Temporal Coupling* method, whose goal is to align events across multiple conditions assuming that a transcription factor binds to the same genomic coordinates across conditions. We test the Spatial Coupling using toy and real data comparing it with a Simple Mixture model, where the independence assumption between reads' positions and their assignments is taken into account. We show that the latter is generally superior in terms of locating the events more accurately and more efficient in terms of running time to the proposed method. In addition, we apply Temporal Coupling to synthetic and real data and show that it achieves alignment across conditions unlike the Simple Mixture one. Furthermore, we show by using synthetic data that even if the binding events are not aligned or not present in all conditions, the algorithm still holds its alignment property and avoids calling false positive peaks in places where do not actually exist. Lastly, we demonstrate that when binding events are aligned, the spatial resolution of Temporal Coupling is better than that of the Simple Mixture model and furthermore better sensitivity and specificity are achieved.

Thesis Supervisor: David K. Gifford
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

I would like to thank every member of this research group since I was embraced with homeliness and devoutness by each one individually. Since I do not like to give impersonal acknowledgments, I would like to thank everyone for specific reasons.

First, Amy, for all the interesting discussions we had and for her encouragement and advice especially when writing my thesis. Jeanne, the "soul" of this group, for her friendliness and her helpfulness she showed every time I asked for something.

Bob, especially for helping me out to set up the Java working environment and with every other question I had since I first joined this group. Yuchun, for all the help he gave me and the interesting chats we had other than just research related matters. Robin, for her availability and enthusiasm when having discussed research related problems with me and providing me every time with a good biological intuition. Alex, for his sincere willingness to respond to all my questions (even the most naive ones) always instantly, being particularly clear and neat. Tim, for all the time he devoted in introducing me to concepts and methods with his own unique style and passion especially in the period where we were working on a joint project.

Shaun, being one of the leading forces of this group, providing me always with good feedback, suggestions, advice and being exactly as specific and clear in our discussions as a good advising requires. Sometimes though, our good collaboration has been stretched due to his low opinion on the Greek national soccer team ;-). To Chris, not only would I like to thank him for his numerous offers for help, but also for considering him as one of my friends and for all the countless discussions we had in many 'gray' late evenings at the lab.

Closing the acknowledgments circle for the people of my research group, I would like to thank my advisor David Gifford for his willingness to help with and advise me on matters of any possible nature and especially the freedom he gives to his students in research ensuring their eager engagement in any project they are dealing with.

Lastly, there are two other groups of people that I am infinitely grateful to. First, Mr. and Mrs. Kanellakis who supported me with the Paris Kanellakis fellowship during my first

year at MIT. Those two wonderful people, Mr. Leuteris and Mrs. Roula, with all the terrible misfortunes that occurred in their lives found the courage to help young people pursue their dreams and make them feel members of the Kanellakis family.

Last but profoundly not least, I am eternally grateful to my parents and sister, to all their continuing and true love and support without any expectations of taking back.

I would like to close this Acknowledgments section by quoting a few lines from the poem *"Patience"* of Rabindranath Tagore (an Indian Nobel Literature Prize winner who was a poet, novelist and educator among others), whose meaning accompanied me to some discouraging moments throughout the course of this "trip".

> *"I will keep still and wait like the night with starry vigil and its head bent low with patience.*
>
> *The morning will surely come, the darkness will vanish, and thy voice pour down in golden streams breaking through the sky."*

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The identification of DNA–protein binding sites provides key information about the mechanisms that underly gene regulation and chromatin organization. Protein–DNA interactions provide insights into transcription initiation and structural modifications (e.g. histone modification) that take place in order to promote or repress transcription. Binding site discovery can also assist in learning how different molecules (e.g. transcription factors) act synergistically or even auto-regulatory to achieve a certain result.

Techniques for quantifying the amount and determining the regions bound by transcription factors (or other binding molecules) establish an appropriate framework for identifying binding sites of interest. The most widely used methods are ChIP-Chip and ChIP-Seq, which measure where a transcription factor of interest binds based on the DNA that is co-immunoprecipitated with the protein. More details on these two methods will be given later on.

However, even with these techniques, it is difficult to determine a binding site with high resolution since the ChIP signal at a given locus is affected by several factors: the fraction of cells with the binding event occurring at the time of cross-linking, the binding affinity at that location, the efficiency of cross-linking or immunoprecipitation of the protein-DNA complex and the efficiency of the amplification and hybridization of the DNA fragments containing the site [6].

As it becomes apparent ChIP-Seq data is noisy and thus make the exact determination of binding events challenging. Previous tools for binding site discovery on ChIP-Seq data generally either discover binding sites conforming to a general peak profile or being significantly enriched relative to some background distribution or control sample. However, no method has thus far proposed a potential process of generating binding events through a complete probabilistic model. Furthermore, to the best of our knowledge, there is no previous method which has addressed the need for binding event alignment across conditions.

The motivation of this thesis is therefore to propose a read generating process under a complete probabilistic framework and to address the problem of event alignment across conditions. Since previous methods relied only on what is observed (reads) rather than considering as well what is hidden (events), we anticipate that by providing a thorough generating process framework we are going to improve spatial resolution of events and furthermore sensitivity and specificity. In the case of multiple conditions, we additionally seek to provide alignment of events since the locations where transcription factors are bound are not expected to change across conditions.

In this thesis, we present probabilistic based methods for discovering events utilizing the relative positions of reads and data from multiple conditions. The first method, Spatial Coupling, will test the hypothesis that better spatial resolution is accomplished given the assumption that proximal reads are likely to be the result of the same binding event. That is, reads that are close to each other are expected to be generated from the same binding event. So, we will model this constraint by considering that the assignment of a read to an event will affect the assignment of its closest read.

The second method, Temporal Coupling, will assess the hypothesis that better spatial resolution, sensitivity and specificity are achieved by assuming that events occurring in different conditions are expected to be located at the same position (since a protein is likely to bind to the same DNA target across conditions, unless data suggest evidence to the contrary). Therefore, we will introduce a dependency between the calling of events at different conditions.

1.1 Biological Background

In this section, we are going to describe the biology behind our problem beginning from very essential concepts and definitions. We will start by first presenting the process of Transcription Factor (TF) binding in Subsection 1.1.1, then briefly describe the immunoprecipitation process (Subsection 1.1.2), and finally outline the two most popular methods of identifying binding sites on the genome with the aid of immunoprecipitation; ChIP-Chip (Subsection 1.1.3) and ChIP-Seq (Subsection 1.1.4).

1.1.1 TF Binding

Binding sites are regions of protein, DNA or RNA where specific molecules are bound. DNA binding sites are usually short DNA sequences (typically, 4–30 bp long, but up to 200 bp for recombination sites). When categorized according to their biological function, they are split into restriction, recombination and Transcription Factor (TF) binding sites. We are going to focus solely on the last category.

TFs are proteins that bind to specific DNA sequences and thereby regulate transcription. They can either act as activators (by promoting) or repressors (by blocking) the recruitment of RNA polymerase, the enzyme which catalyzes the transcription of DNA to RNA. They bind either to distal enhancers or proximal promoter regions.

In more detail, TFs may contain the following domains: a *DNA-Binding Domain (DBD)* which physically contacts the DNA, a *Trans-Activation Domain (TAD)* which may interact with other proteins such as transcription co-regulators and an optional *Signal Sensing Domain (SSD)* which senses external signals and in response transmits these signals to the rest of the transcription complex thus coordinating the regulation of a gene.

A defining feature of a TF (unlike other binding molecules such as co-activators, histone acetylases, deacetylases, kinases and methylases) is that their DNA Binding Domains (DBDs) attach to specific sequences of DNA.

A typical schematic of a TF is depicted below:

Lastly, it is worth mentioning that TF binding preferences are typically represented

Figure 1-1: **Schematic diagram of the amino acid sequence of a Transcription Factor (TF).**
A TF contains (1) a DNA-Binding Domain (DBD), a (2) Trans-Activation Domain (TAD) and a (3) Signal Sensing Domain (SSD). The order presented here may differ in different types of TFs. Furthermore, the trans-activation and signal sensing functions are usually contained within the same domain.

using Position Specific Weight Matrices (PSWM) and are often depicted using sequence logos rather than consensus sequences, since DNA binding sites for a given TF are usually all different and have varying degrees of affinity.

1.1.2 Immunoprecipitation

Immunoprecipitation (IP) is a technique of precipitating a protein using an antibody to bind to that particular protein. By this process, a protein of interest can be isolated from a sample containing thousands of different proteins. There are different types of immunoprecipitation including *individual protein ImmunoPrecipitation (IP)*, *protein complex ImmunoPrecipitation (co-IP)*, *Chromatin ImmunoPrecipitation (ChIP)* and *RNA ImmunoPrecipitation (RIP)*.

i. The first one (IP) is the simplest where an antibody is used to isolate a protein of interest out of a solution.

ii. In the second one (co-IP), an antibody targeting a member of a protein complex is used with the intention of pulling down the whole complex.

iii. In Chromatin ImmunoPrecipitation (ChIP), the same procedure is applied for the purpose of identifying the locations of binding sites on the genome for a particular protein. The principle governing this technique is that DNA-binding proteins are

cross-linked to DNA, thus allowing for the protein-DNA complex to be pulled out in the presence of the proper antibody. In more detail, the cross-linking is achieved by applying formaldehyde to the cells. The cells are then lysed and the DNA is broken into fragments of $0.2 - 1$ Kb by sonication. At this point, protein-DNA complexes are purified by immunoprecipitation. The purified protein-DNA complexes are then heated to reverse the formaldehyde cross-linking of the protein-DNA complexes, inducing the DNA to be separated from the proteins. Because it can be difficult to find a specific antibody to a single TF, constructs containing a tagged version of the TF, called *tagged proteins* can be generated. In more detail, tags are attached onto the C- or N- terminal end of the protein of interest. In that way, the same tag can be used on many different proteins, many different times.

iv. Lastly, RIP is very similar to ChIP with the only difference that it targets RNA binding proteins.

1.1.3 ChIP-Chip

ChIP-Chip is a technique combining Chromatin ImmunoPrecipitation ("ChIP") and microarray technology ("Chip"). As the name suggests, it provides a framework for the identification of binding sites of DNA-binding proteins on a genome-wide basis.
In brief, ChIP-Chip is comprised of the following steps:

i. Initially, the protein of interest is cross-linked with the DNA it binds to *in vivo*.

ii. The cells are lysed and the DNA is sheared by sonication (or digested by micrococcal nuclease) generating ∼300–500 bp double stranded DNA fragments. Obviously, this mixture contains both "bare" DNA strands, strands attached to molecules of the protein of interest and strands attached to other proteins.

iii. An antibody specific to the protein forces all the "bare" segments to be filtered out since it holds all the protein-DNA complexes, thus leaving the segments not binding to the protein unprotected.

iv. Now, the protein-DNA complexes are reverse cross-linked and purified and the protein of interest may at this step be removed.

v. After that, double-stranded DNA fragments are amplified, denatured and fluorescented.

vi. Furthermore, the fragments are hybridized to the surface of a DNA microarray (DNA chip) comprising of single-stranded sequences of the genomic regions of interest which hybridize to their complementary fragments (created by the aforementioned process).

vii. Lastly, after allowing hybridization for sufficient time, the array is illuminated with fluorescent light, then the image is converted to intensity values and after proper normalization and statistical testing, it is ready to be utilized.

The above are shown better on the figure below:

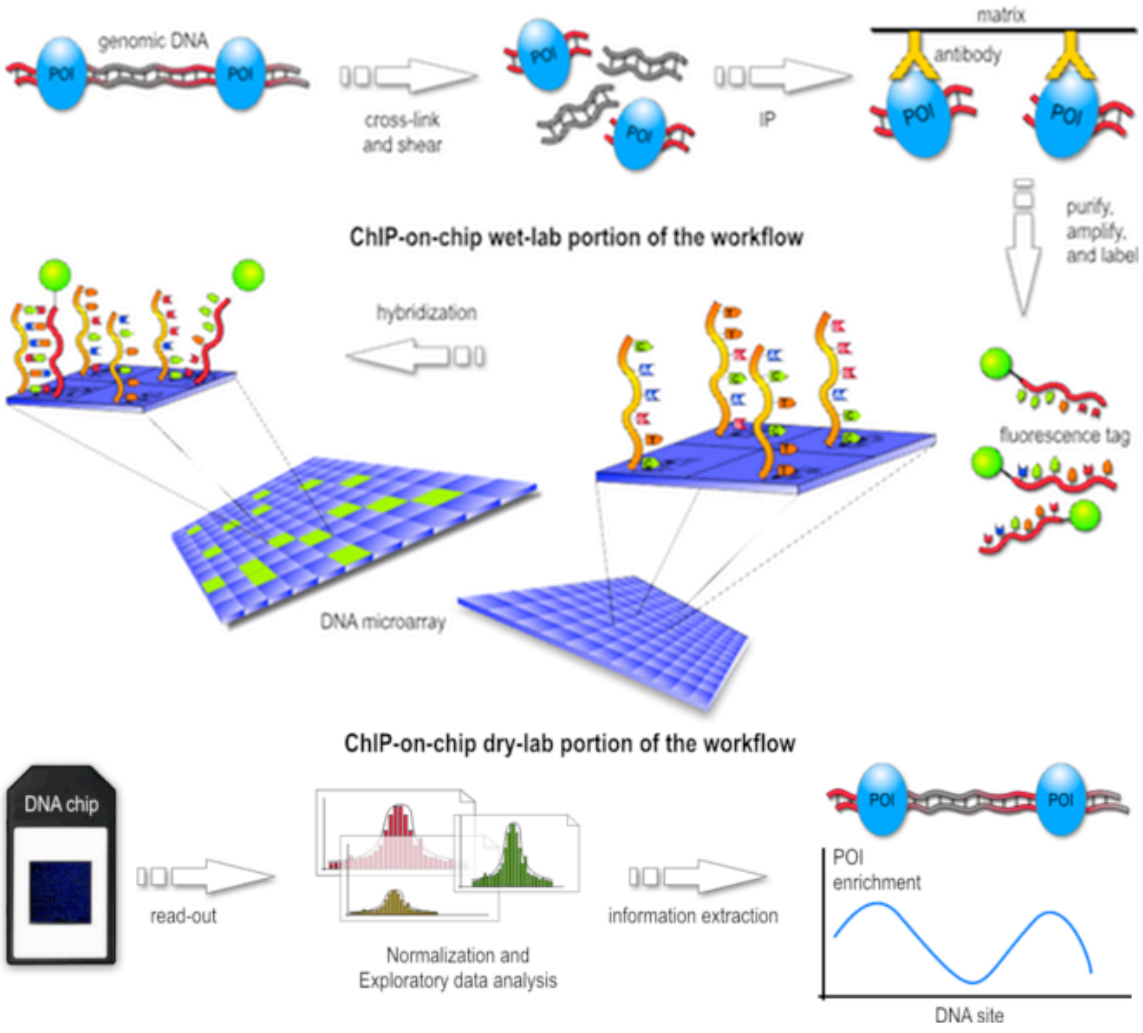


Figure 1-2: **Schematic diagram of a ChIP-Chip experiment.**
The figure illustrates the ChIP-Chip experimental procedure. The protein of interest is cross-linked with the DNA it binds to, then it is sheared by sonication following cell lysis and DNA fragments of length ∼300–500 bp are obtained, filtering out those which are not bound by the protein of interest. Lastly, the remaining segments are poured on the surface of a DNA microarray and are hybridized with their complementary probes generating intensities analogous to the hybridization levels.

1.1.4 ChIP-Seq

ChIP-Seq is the "descendant" of ChIP-Chip. It is also used for the analysis of protein-DNA interactions. ChIP-Seq essentially combines Chromatin ImmunoPrecipitation (ChIP) with DNA sequencing to identify binding sites. In more detail, the experimental process of ChIP-Seq is as follows:

i. The first phase of ChIP-Seq up to the point of produced segments is identical to the one described in ChIP-Chip. In addition, oligonucleotide adapters are added to the small stretches of DNA that were bound to the protein of interest to enable massively parallel sequencing.

ii. After size selection, all the remaining fragments are sequenced simultaneously using a genome sequencer so that features can be precisely located on the chromosomes.

The above procedure can be summarized in Figure 1-3 below.

At this point, we should note that although the initial fragments (bound by the protein of interest) range from ∼200–1,000 bp, only the first ∼25–50 bp from the ends ("reads") are sequenced, as shown in Figure 1-4 (*reads* are also known as *tags*). The resulting reads are mapped back to a reference genome, and, usually, only those reads that map to a unique locus are considered [14].

It is also worth mentioning that ChIP-selected sample is typically compared to a background sample collected without antibody addition during the immunoprecipitation step since random protein-DNA cross-linking can occur, thus resulting in the retention of non-specific DNA after the immunoprecipitation step (something that may lead to false inference about the data). Then, the two DNA populations (sample with and without (background) antibody) are differentially labeled and compared [8].

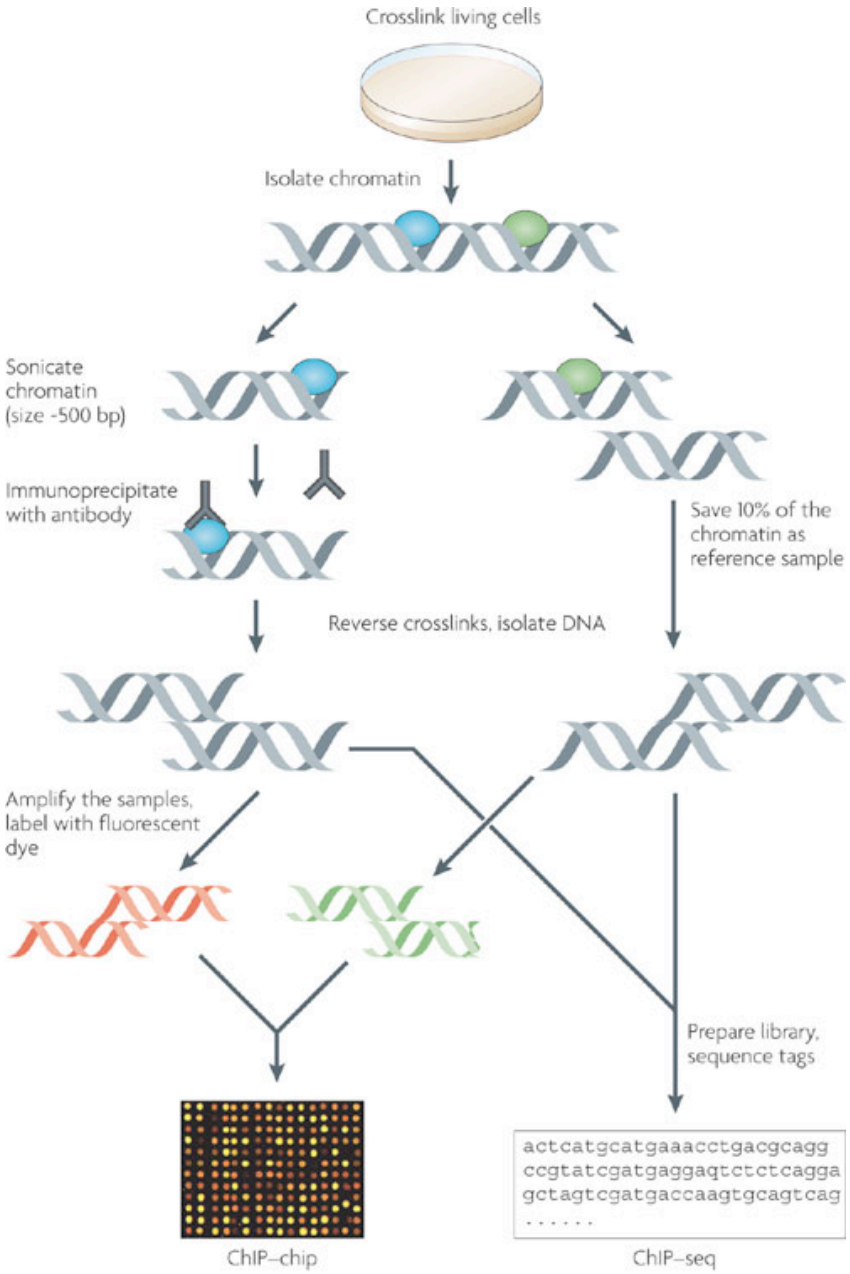


Figure 1-3: **Schematic diagram of a ChIP-Seq experiment.**
The figure illustrates the ChIP-Seq experimental procedure and provides direct comparison with the ChIP-Chip protocol. The protein of interest (here: blue) is cross-linked with DNA by treating cells with formaldehyde or another chemical reagent. Next, an antibody specific to the protein of interest is used to selectively attach to the protein-bound DNA fragments. Finally, the immunoprecipitated protein-DNA links are purified, reversed and then after the molecules of the protein of interest are removed, the recovered DNA is assayed for its sequence determination. From the sequencing process short read length sequences of $\sim$ 25–50 bp are generated providing a far better resolution than the ones created by the ChIP-Chip technology (which are on the order of hundreds of base pairs). (Figure taken from [12])

Figure 1-4: **Read generation from DNA fragments where the protein is bound.** As it is shown, only the ∼25–50 bp ends of the fragments are sequenced (green rectangles) from both directions. As a consequence, the peak signal is not exactly located on the binding site but rather on close stranded regions.
(Figure taken from [16])

1.1.5 Comparison between ChIP-Chip and ChIP-Seq

Closing the section of "Biological Background", we will briefly compare the two different techniques, ChIP-Chip and ChIP-Seq, and thus show why the latter has prevailed lately.

First, ChIP-Seq is not restricted to a specific number of probes while in ChIP-Chip it is necessary to design the microarray specifically for regions of interest in advance. It has also minimal cost compared to ChIP-Chip (especially when it is run on whole genome) and a requirement for fewer replicate experiments [8]. ChIP-Chip also suffers from serious cross-hybridization in high genomic resolution (especially on mammalian genomes) [7].

Furthermore, ChIP-Seq offers direct whole-genome coverage and sensitivity that increases with sequencing depth. In addition, the vast majority of sites are accessible in the genome and there is no need of knowing in advance the identity of the sequence of interest [7].

From the above, it becomes clear that ChIP-Seq is superior to ChIP-Chip. However, ChIP-Seq itself has its own limitations making the identification of binding sites a difficult task and requiring the development of sophisticated tools for achieving higher resolution. In more detail, ChIP-Seq tags represent only the ends of the ChIP fragments (instead of precise protein-DNA binding sites). Typically, genomic regions with high tag densities serve as candidates for binding locations.

Furthermore, the tag to site distance is often unknown to the user (see Figure 1-5 for details). In addition, the background tag distribution is unknown and the depth of sequencing affects the accuracy of the results [13]. Lastly, ChIP-Seq exhibits regional biases along the genome due to sequencing and mapping biases, chromatin structure and genome copy number variations [17]. This is further corroborated by Nix et al. where they underline that the method of DNA fragmentation, the lack of independence in observations, the degree of repetitiveness, and the error in sequencing and alignment are a few of the known sources of systematic bias [5].

1.2 Related Work

There have been previous attempts in determining the binding sites from ChIP data. Initially, Johnson et al. used an *ad hoc* masking method based on control input data and prior qPCR validated regions to set a threshold and assign confidence in binding peaks [7].

Later on, Robertson et al. estimated global Poisson p-values for windowed data using a rate set to 90% the size of the genome [9]. Jothi el al., with their so-called SISSRs tool used the direction and density of reads and the average DNA fragment length to identify binding sites [14]. In more detail, the whole genome is scanned in windows of size w (default: 20 bp) with consecutive windows overlapping by $w/2$. For each window i, the net tag count c_i is estimated as the number of sense tags minus the number of antisense tags mapped to the same window. The net count as we move from left to right over a binding event transitions from positive to negative, thus the transition point coordinate t is defined as the midpoint between the last seen window with positive net tag count and the current window (which has a negative net tag count) (see Figure 1-5 for details). Each one of these transition points serves as a candidate site required to satisfy three constraints to be called as such: (i) number of sense tags p in the region defined by coordinates $[t - F, t]$ is at least E, (ii) number of antisense tags n in the region defined by coordinates $[t, t + F]$ is at least E (E set by the user), and (iii) $p + n$ is at least R, which is estimated based on the user-set false discovery rate (FDR) D.

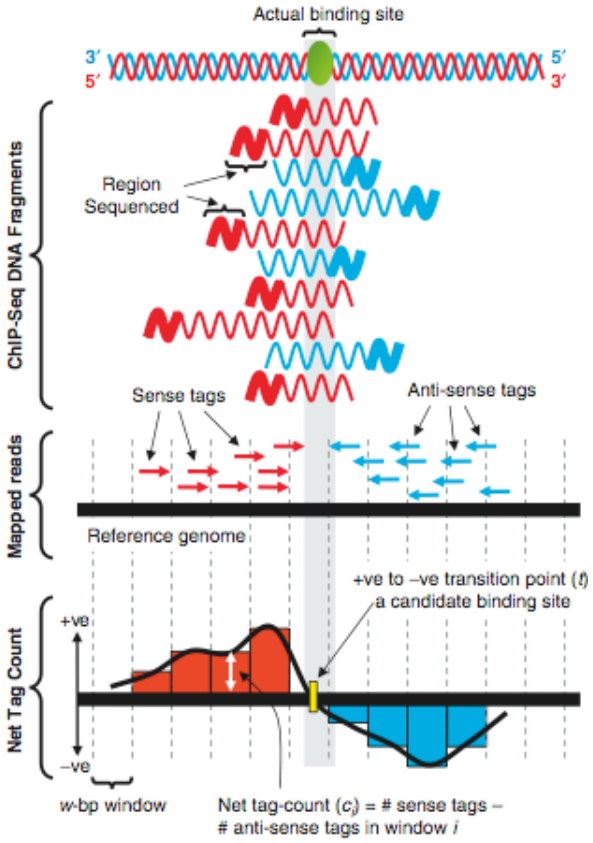


Figure 1-5: **Tag count distribution around a binding site.**
Sense and antisense tags (sense and antisense ∼25–50 bp ends) are sequenced from the larger fragments where the protein has been bound, thus generating the distribution shown on the bottom subfigure. The peaks of tag counts are observed near the true binding site. The region is divided into windows of w bp length and the *net tag count* = $\#$*sense tags* − $\#$*anti-sense tags* is evaluated for each window.
(Figure taken from [14])

The QuEST tool, on the other hand, uses a rather simplified but satisfactory technique [2]. It first constructs two separate profiles (based on kernel density methods) for the forward and reverse strand tags, respectively. Then, it evaluates the distance between the peaks of the two profiles, the so-called *peak shift*. Eventually, it moves the "forward" profile half the shift size to the right and the "reverse" profile half the shift size to the left and sums up those two to acquire the Combined Density Profile (CDP) as shown in Figure 1-6.

Model-based Analysis of ChIP-Seq (MACS) tries as well to empirically model the peak size to determine the binding location [17]. Given a sonication size (*bandwidth*) and a high–confidence fold–enrichment ($mfold$), MACS slides $2 \cdot bandwidth$ windows across the genome to find regions with tags more than $mfold$ enriched relative to a random tag genome distribution. Lastly, the distance between the modes of the forward and reverse peaks is estimated and all the tags are moved toward the 3' end. The mode of this unified function serves as the candidate binding site. MACS was found to be superior to Johnson et al. ([7]), FindPeaks ([3]) and QuEST ([2]) [17].

Kharchenco et al. proposed three different methods: Window Tag Density (WTD), Matching Strand Peaks (MSP) and Mirror Tag Correlation (MTC). In their first method, WTD, each position gets a score based on strand–specific tag counts upstream and downstream of the position. In MSP, local peaks (of strand–specific density functions) are determined and then the candidate positions are identified which are surrounded by positive– and negative– strand peaks of a comparable magnitude at the expected distance. In MTC, the genome is scanned for the detection of non trivial forward– and reverse– strand tag patterns that mirror each other [13]. See Figure 1-7 for details.

Finally, PeakSeq is a two–pass approach where in the first stage putative binding sites are discovered and during the second stage the ones that are significantly enriched compared to the normalized control are kept. In more detail, in the first pass, peaks are called that are substantially enriched compared to a simulated simple null background model. In this way, the first pass is used as a pre–filtering phase where candidate regions are selected for further comparison against the input–DNA control. In the second phase, the control is normalized

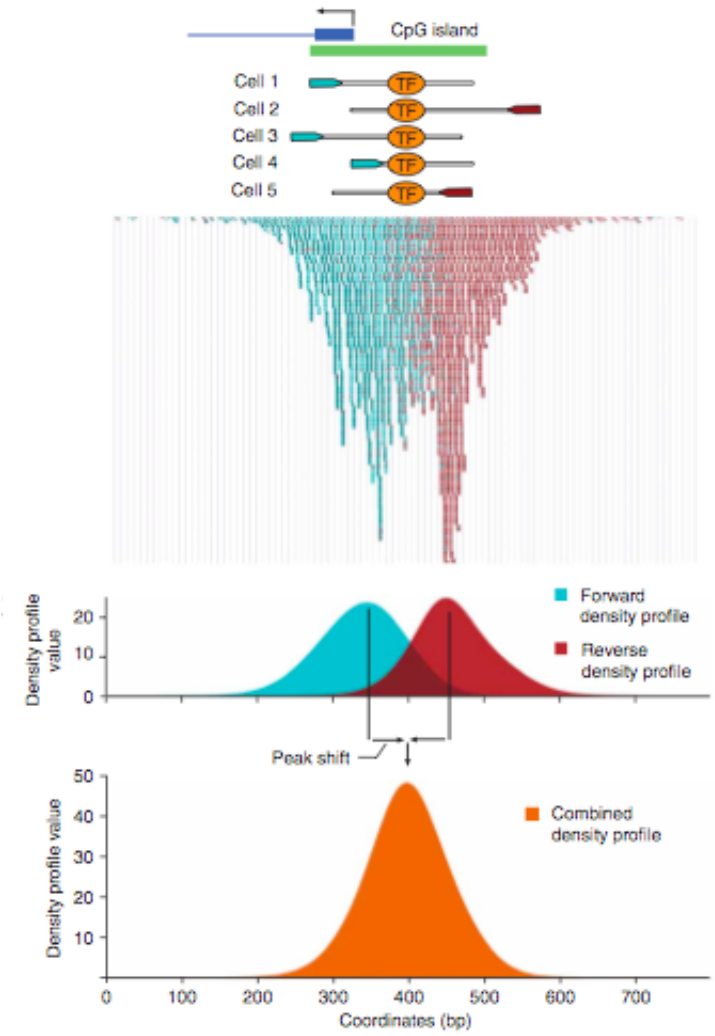


Figure 1-6: **QuEST methodology.**
From the forward and reverse tag counts two density functions are generated (based on kernel methods). Then, the distance between those two peaks is estimated and the two functions ("forward" and "reverse") are shifted half of this distance to the right and left, respectively, to obtain the Combined Density Profile (CDP).
(Figure taken from [2])

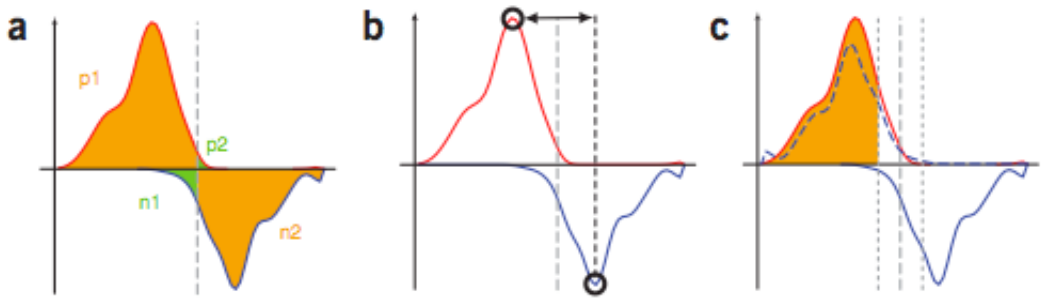


Figure 1-7: **Kharchenco et al. binding detection methods.**
(a) In WTD, the candidate positions are identified based on the difference between the geometric average of the tag counts within the regions marked by orange color ($p1$ and $n2$) and the average tag count within the regions marked by green color ($n1$ and $p2$). (b) In MSP, the local maxima (open circles) are identified on the forward and reverse strands first, and then the positions are determined based on where two such peaks are present in the right order, with the expected separation and comparable magnitude. (c) In MTC, the "mirror" correlation between forward– and reverse– strand tag densities is evaluated. The mirror image of reverse–strand tag density is shown by a broken blue line.
(Figure taken from [13])

against the sample. More explicitly, the background of the sample is normalized by linear regression of the counts of tags from the control against the sample for windows (∼10 Kbp) along each chromosome. Now, only regions that are enriched in the counts of the number of mapped sequence tags in the ChIP-seq samples relative to the control are accepted as binding sites [10].

The above methods, each one with its merits and weaknesses, seek for binding events by applying statistical approaches mainly exploiting the tag count distribution and its shape but no one really proposes a probabilistic model of the read generating process. In this thesis, we will try to simulate this process by designing and exploring probabilistic models taking advantage of the read count distribution as well as the relative distances between reads and the requirement for events being located at the same position across conditions (unless otherwise supported from the data).

Chapter 2

Method Description

In this chapter, we are going to describe our methodologies. We will start by presenting some underlying theory forming the basis of our methods in Section 2.1. First of all, since both our methods are examples of *Probabilistic Graphical Models*, we will briefly outline this group of models. In addition, since the dependency between reads' assignments in the Spatial Coupling method will be modeled by *Hidden Markov Models* (*HMMs*), we will also describe this special type of probabilistic models and, furthermore, refer briefly to *Mixture Models* since they constitute the body of our second method, Temporal Coupling. That is, each read will be generated by one of the (true) events and will have no dependence on the positions of other reads. A few remarks will be made on the sparse prior used in the Maximum A Posteriori (MAP) approach of both problems. Namely, we are going to show that by using this type of prior, we will enforce a parsimonious policy where as fewer events as possible would be recovered to explain the observed read landscape. Later, we are going to outline the notation of our methodology in Section 2.2 and lastly present our methodologies in Sections 2.3 (*Spatial Coupling*) and 2.4 (*Temporal Coupling*).

2.1 Underlying Theory

In this section, we will present the underlying theory related to our methodology. Variables in plain font (e.g. x) will denote scalar random variables, while variables in lowercase bold font (e.g. $\mathbf{x}$) will denote vector random variables. Vector variables can either represent sequences of scalar random variables or a random variable of dimension D. In our case, it will represent a sequence of random variables (e.g. the positions of reads in an experiment: $\mathbf{x} = \{x_1, \ldots, x_N\}$). Lastly, variables in uppercase bold font (e.g. $\mathbf{X}$) will represent vectors of vector random variables. E.g. the positions of reads in all experiments. Three symbols that will also be used throughout the text will be $\mathbb{R}, \mathbb{R}^+, \mathbb{E}$ representing the domain of real numbers ($\mathbb{R} = (-\infty, +\infty)$), the domain of real positive numbers ($\mathbb{R}^+ = (0, +\infty)$) and the expectation of a random variable, respectively. Needless to say, that a variable $\mathbf{x}$ of dimension D taking on any real value, will be denoted as $\mathbf{x} \in \mathbb{R}^D$.

Besides, we assume all probabilities of random variables to be well-posed, that is, to sum (or integrate, if continuous) to one in their support. A last denotation would be the way we represent conditioned probabilities. If a random variable is conditioned on a random variable, we will represent the conditioning with a vertical bar ($|$), but if it is conditioned on a non-random (yet unknown) parameter, we will represent the conditioning with a semicolon ($;$).

Concerning the representation of variables into graphical models, we will represent random variables with big circles. Observed ones will be painted in gray, while hidden (latent) in white. Dependencies between variables will be represented by arrows connecting circles. Also, rounded rectangles will represent the different states of a random variable (and not the random variable itself) and lastly small black filled circles will represent parameters, which are unknown but not random.

2.1.1 Hidden Markov Models

Graphical models are *graphs* comprising of nodes which are connected by *links* (also known as *edges* or *arcs*). Each node, represents a random variable and links between nodes express

probabilistic relationships between these variables. Gray nodes represent observed variables while white hidden ones. *Bayesian Networks* are directed (acyclic) graphical models where the directionality of the arrows indicates a "causality" relationship between the parent and the child node. When the data are sequential, that is, data whose order does matter, the independency assumption does not hold. In this context, successive observations are highly correlated. When the observations are dependent on a countable number of generating states (being represented by discrete hidden random variables), they are usually modeled by the so-called *Hidden Markov Models, HMMs*. On the other hand, when the states causing a specific success of observations is non-countable, thus corresponding to continuous random variables, this problem is usually addressed by the so-called *Linear Dynamical Systems, LDSs* [4].

Here, we will briefly describe HMMs. HMMs are comprised of observed and hidden random variables. Each observed random variable ($\mathbf{x}_n$) is dependent on its corresponding (discrete) hidden variable ($\mathbf{z}_n$), representing the state that generated this specific observation, and, in turn, each hidden variable $\mathbf{z}_n$ is dependent on its most recent one $\mathbf{z}_{n-1}$ forming a first-order Markov chain. In other words, a hidden variable is independent of all variables except the most recent one. This is formalistically posed that, two variables $\mathbf{z}_{n+1}$, $\mathbf{z}_{n-1}$ are conditionally independent given $\mathbf{z}_n$:

$$\mathbf{z}_{n+1} \perp\!\!\!\perp \mathbf{z}_{n-1} | \mathbf{z}_n \tag{2.1}$$

From the figure, it also directly follows that $\mathbf{x}_n$ is dependent only on $\mathbf{z}_n$:

$$\mathbf{x}_n \perp\!\!\!\perp \mathbf{z}_{-n} | \mathbf{z}_n \tag{2.2}$$

where with $\mathbf{z}_{-n}$, we denote every variable except for $\mathbf{z}_n$.

The aforementioned are depicted on the figure below (Figure 2-1).

So, if we represent the observed variables as $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and the hidden as $\mathbf{Z} =$

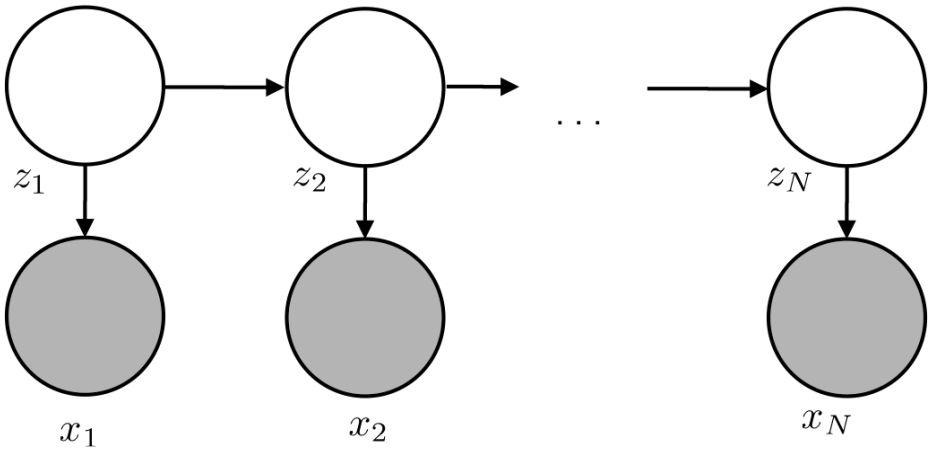


Figure 2-1: **Hidden Markov Model.**
In an HMM, the sequential data are represented with a Markov chain of hidden variables, with each observation conditioned on the state of the corresponding hidden variable. Each hidden variable is dependent on its previous one forming a first-order Markov chain.

$\{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$, then the likelihood of the complete data would be:

$$\begin{aligned} p(\mathbf{X}, \mathbf{Z}) &= p(\mathbf{X}|\mathbf{Z})p(\mathbf{Z}) && \Leftrightarrow \\ &= p(\mathbf{x}_1, \ldots, \mathbf{x}_N|\mathbf{z}_1, \ldots, \mathbf{z}_N)p(\mathbf{z}_1, \ldots, \mathbf{z}_N) && \Leftrightarrow \\ &= \prod_{n=1}^{N} p(\mathbf{x}_n|\{\mathbf{z}_1, \ldots, \mathbf{z}_N\}) \prod_{n=2}^{N} p(\mathbf{z}_n|\{\mathbf{z}_1, \ldots, \mathbf{z}_{n-1}\})p(\mathbf{z}_1) && \overset{2.1,\ 2.2}{\Leftrightarrow} \\ &= p(\mathbf{z}_1) \left[\prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{z}_n)\right] \left[\prod_{n=2}^{N} p(\mathbf{z}_n|\mathbf{z}_{n-1})\right] && (2.3) \end{aligned}$$

We see from 2.3, that the dependence between the hidden variables is expressed through the probability $p(\mathbf{z}_n|\mathbf{z}_{n-1})$. This is called *transition probability* and we actually impose a further constraint by considering a *homogeneous chain*, that is the absolute ordering of two variables does not matter but rather their relative ordering.
Namely:

$$p(\mathbf{z}_n|\mathbf{z}_{n-1}) = p(\mathbf{z}_{n+1}|\mathbf{z}_n)$$

Usually, in the traditional HMM setting, we are interested in evaluating the parameters $\epsilon_j = p(z_1)$, $o_{j,l} = p(x_n = l|z_n = j)$ and $A_{k,j} = p(z_n = j|z_{n-1} = k)$. However, we will provide

a complete derivation of the M step under the context of our method later on.

2.1.1.1 The Forward–Backward Algorithm

As we have just mentioned, we will derive the M step of the EM algorithm particularly for the specifications of our problem. Meanwhile, we will briefly provide the E step of the algorithm, that is, give the formulae for evaluating two key posterior probabilities of hidden variables conditioned on observed ones: $p(z_n = j|\mathbf{x}) \triangleq \zeta(z_n = j)$ and $p(z_{n-1} = k, z_n = j|\mathbf{x}) \triangleq \xi(z_{n-1} = k, z_n = j)$.

We will define two additional probabilities $p(x_1, \ldots, x_n, z_n = j) \triangleq \alpha(z_n = j)$ and $p(x_{n+1}, \ldots, x_N|z_n = j) \triangleq \beta(z_n = j)$ that will help us evaluate the aforementioned probabilities.

It is:

$$
\begin{aligned}
\zeta(z_n = j) &\triangleq p(z_n = j|\mathbf{x}) \propto p(\mathbf{x}, z_n = j) = p(\{x_1, \ldots, x_N\}, z_n = j) \Leftrightarrow \\
&\propto p(\{x_1, \ldots, x_N\}|z_n = j)p(z_n = j) \Leftrightarrow \\
&\propto p(x_1, \ldots, x_n|x_{n+1}, \ldots, x_N, z_n = j)p(x_{n+1}, \ldots, x_N|z_n = j)p(z_n = j) \overset{x_1,\ldots,x_n \perp\!\!\!\perp x_{n+1},\ldots,x_N|z_n}{\Leftrightarrow} \\
&\propto p(x_1, \ldots, x_n|z_n = j)p(x_{n+1}, \ldots, x_N|z_n = j)p(z_n = j) \Leftrightarrow \\
&\propto p(x_1, \ldots, x_n|z_n = j)p(z_n = j)p(x_{n+1}, \ldots, x_N|z_n = j) \\
&\propto \underbrace{p(x_1, \ldots, x_n, z_n = j)}_{\alpha(z_n=j)} \underbrace{p(x_{n+1}, \ldots, x_N|z_n = j)}_{\beta(z_n=j)} \Leftrightarrow \\
&\propto \alpha(z_n = j)\beta(z_n = j)
\end{aligned}
$$

Similarly:

$$
\begin{aligned}
\xi(z_{n-1} = k, z_n = j) &\triangleq p(z_{n-1} = k, z_n = j|\mathbf{x}) \propto p(\mathbf{x}, z_{n-1} = k, z_n = j) \Leftrightarrow \\
&\propto p(x_1, \ldots, x_{n-1}, x_n, x_{n+1}, \ldots, x_N|z_{n-1} = k, z_n = j)p(z_{n-1} = k, z_n = j) \Leftrightarrow \\
&\propto p(x_1, \ldots, x_{n-1}|x_n, x_{n+1}, \ldots, x_N, z_{n-1} = k, z_n = j) \cdot \\
&\quad p(x_n, x_{n+1}, \ldots, x_N|z_{n-1} = k, z_n = j)p(z_n = j|z_{n-1} = k)p(z_{n-1} = k)
\end{aligned}
$$

It is: $x_1, \ldots, x_{n-1} \perp\!\!\!\perp x_n, x_{n+1}, \ldots, x_N, z_n | z_{n-1}$ and $x_n, x_{n+1}, \ldots, x_N \perp\!\!\!\perp z_{n-1} | z_n$.

So:

$$\begin{aligned}
\xi(z_{n-1} = k, z_n = j) &\propto p(x_1, \ldots, x_{n-1} | z_{n-1} = k) p(x_n, x_{n+1}, \ldots, x_N | z_n = j) p(z_n = j | z_{n-1} = k) p(z_{n-1} = k) \\
&\propto p(x_1, \ldots, x_{n-1} | z_{n-1} = k) p(z_{n-1} = k) \cdot \\
&\quad p(x_n | x_{n+1}, \ldots, x_N, z_n = j) p(x_{n+1}, \ldots, x_N | z_n = j) p(z_n = j | z_{n-1} = k)
\end{aligned}$$

Since $x_n \perp\!\!\!\perp x_{n+1}, \ldots, x_N | z_n$:

$$\begin{aligned}
\xi(z_{n-1} = k, z_n = j) &\propto \underbrace{p(x_1, \ldots, x_{n-1}, z_{n-1} = k)}_{\alpha(z_{n-1}=k)} p(x_n | z_n = j) p(z_n = j | z_{n-1} = k) \underbrace{p(x_{n+1}, \ldots, x_N | z_n = j)}_{\beta(z_n=j)} \\
&\propto \alpha(z_{n-1} = k) p(x_n | z_n = j) p(z_n = j | z_{n-1} = k) \beta(z_n = j)
\end{aligned}$$

Therefore, we just have to determine $\alpha(z_n = j)$, $\beta(z_n = j), \forall n$ to evaluate $\zeta(z_n = j)$ and $\xi(z_{n-1} = k, z_n = j)$. Since $\zeta(z_n = j)$ and $\xi(z_{n-1} = k, z_n = j)$ will be present on the nominator and denominator of each estimated parameter (in the M step) as will be shown later, we simply have to calculate these proportional values without normalizing them, thus avoiding further computational cost.

Now, we just have to find a recursive formula for evaluating $\alpha(z_n = j), \beta(z_n = j)$.
It is:

$$\begin{aligned}
\alpha(z_n = j) &= p(x_1, \ldots, x_n, z_n = j) = \sum_k p(x_1, \ldots, x_{n-1}, x_n, z_{n-1} = k, z_n = j) \Leftrightarrow \\
&= \sum_k p(x_1, \ldots, x_{n-1} | x_n, z_{n-1} = k, z_n = j) p(x_n, z_{n-1} = k, z_n = j) \overset{x_1,\ldots,x_{n-1} \perp\!\!\!\perp x_n, z_n | z_{n-1}}{\Leftrightarrow} \\
&= \sum_k p(x_1, \ldots, x_{n-1} | z_{n-1} = k) p(x_n | z_{n-1} = k, z_n = j) p(z_n = j | z_{n-1} = k) p(z_{n-1} = k)
\end{aligned}$$

It is: $x_n \perp\!\!\!\perp z_{n-1} | z_n$.

So:

$$\begin{aligned}\alpha(z_n=j) &= \sum_k \underbrace{p(x_1,\ldots,x_{n-1},z_{n-1}=k)}_{\alpha(z_{n-1}=k)} p(x_n|z_n=j)p(z_n=j|z_{n-1}=k) \Leftrightarrow \\ &= \sum_k \alpha(z_{n-1}=k)p(z_n=j|z_{n-1}=k)p(x_n|z_n=j)\end{aligned}$$

It is:

$$\begin{aligned}\beta(z_n=j) &= p(x_{n+1},x_{n+2},\ldots,x_N|z_n=j) = \sum_k p(x_{n+1},\ldots,x_N,z_{n+1}=k|z_n=j) \Leftrightarrow \\ &= \sum_k p(x_{n+2},\ldots,x_N|x_{n+1},z_n=j,z_{n+1}=k)p(x_{n+1}|z_n=j,z_{n+1}=k)p(z_{n+1}=k|z_n=j)\end{aligned}$$

It is: $x_{n+2},\ldots,x_N \perp\!\!\!\perp x_{n+1},z_n|z_{n+1}$ and $x_{n+1} \perp\!\!\!\perp z_n|z_{n+1}$.

Therefore:

$$\begin{aligned}\beta(z_n=j) &= \sum_k \underbrace{p(x_{n+2},\ldots,x_N|z_{n+1}=k)}_{\beta(z_{n+1}=k)} p(z_{n+1}=k|z_n=j)p(x_{n+1}|z_{n+1}=k) \Leftrightarrow \\ &= \sum_k p(z_{n+1}=k|z_n=j)p(x_{n+1}|z_{n+1}=k)\beta(z_{n+1}=k)\end{aligned}$$

Gathering all the important formulae we have:

$$\zeta(z_n=j) \propto \alpha(z_n=j)\beta(z_n=j) \tag{2.4}$$

$$\xi(z_{n-1}=k,z_n=j) \propto \alpha(z_{n-1}=k)p(x_n|z_n=j)p(z_n=j|z_{n-1}=k)\beta(z_n=j) \tag{2.5}$$

where $\alpha(z_n=j)$ and $\beta(z_n=j)$ are evaluated from:

$$\alpha(z_n=j) = \sum_k \alpha(z_{n-1}=k)p(z_n=j|z_{n-1}=k)p(x_n|z_n=j) \tag{2.6}$$

$$\beta(z_n=j) = \sum_k p(z_{n+1}=k|z_n=j)p(x_{n+1}|z_{n+1}=k)\beta(z_{n+1}=k) \tag{2.7}$$

It follows easily that the values are initialized as:

$$\alpha(z_1 = j) = \pi_j \cdot p(x_1|z_1 = j) \tag{2.8}$$

$$\beta(z_N = j) = 1 \tag{2.9}$$

2.1.1.2 Scaling Factors

Lastly, an important issue that needs to be addressed is that of the number precision. Because these probabilities are often significantly less than unity, as we move forward along the chain, the values of $\alpha(z_n = j)$ and $\beta(z_n = j)$ can go to zero exponentially quickly exceeding the dynamic range of the computer, even if double precision floating point is used.

In this case, we use scaled versions of $\alpha(z_n = j)$ and $\beta(z_n = j)$: $\hat{\alpha}(z_n = j)$ and $\hat{\beta}(z_n = j)$, that is.
More specifically, we define:

$$\hat{\alpha}(z_n = j) \triangleq p(z_n = j|x_1, \ldots, x_n) \tag{2.10}$$

$$\hat{\beta}(z_n = j) \triangleq \frac{p(x_{n+1}, \ldots, x_N|z_n = j)}{p(x_{n+1}, \ldots, x_N|x_1, \ldots, x_n)} \tag{2.11}$$

$$c_n \triangleq p(x_n|x_1, \ldots, x_{n-1}) \tag{2.12}$$

With a little algebraic manipulation, we can see that $\hat{\alpha}(z_n = j)$ and $\hat{\beta}(z_n = j)$ are linked to $\alpha(z_n = j)$ and $\beta(z_n = j)$, respectively, as follows:

$$\hat{\alpha}(z_n = j) = \frac{\alpha(z_n = j)}{p(x_1, \ldots, x_n)}$$

$$\hat{\beta}(z_n = j) = \frac{\beta(z_n = j)}{p(x_{n+1}, \ldots, x_N|x_1, \ldots, x_n)}$$

Since $\hat{\alpha}(z_n = j), \hat{\beta}(z_n = j)$ are ratios of probabilities w.r.t. $\alpha(z_n = j), \beta(z_n = j)$, they are greater than them, thus remaining of order unity.

Now, that we made clear the necessity of the scaled versions of $\alpha(z_n = j)$ and $\beta(z_n = j)$, we can go on giving the recursive formulae for them.

It is:

$$\hat{\alpha}(z_n = j) \propto p(x_n|z_n = j) \sum_k \hat{\alpha}(z_{n-1} = k) p(z_n = j|z_{n-1} = k) \tag{2.13}$$

with

$$c_n = \sum_j p(x_n|z_n = j) \sum_k \hat{\alpha}(z_{n-1} = k) p(z_n = j|z_{n-1} = k) \tag{2.14}$$

In other words, c_n is the normalizing factor of $\hat{\alpha}(z_n = j)$.

Also:

$$\hat{\beta}(z_n = j) = \frac{1}{c_{n+1}} \sum_k \hat{\beta}(z_{n+1} = k) p(x_{n+1}|z_{n+1} = k) p(z_{n+1} = k|z_n = j) \tag{2.15}$$

where we have obtained $c_n s$ from the evaluation of $\hat{\alpha}(z_n = j)s$.

From Equation 2.12, it becomes obvious that the log-likelihood of the data is linked to $c_n s$ as follows:

$$\begin{aligned} \log p(\mathbf{X}) &= \log \prod_{n=1}^{N} p(x_n|x_1, \ldots, x_{n-1}) \\ &= \sum_{n=1}^{N} \log p(x_n|x_1, \ldots, x_{n-1}) \\ &= \sum_{n=1}^{N} \log c_n \end{aligned} \tag{2.16}$$

Eventually, linking $\hat{\alpha}(z_n = j), \hat{\beta}(z_n = j)$ to $\zeta(z_n = j), \xi(z_{n-1} = k, z_n = j)$ we have:

$$\zeta(z_n = j) = \hat{\alpha}(z_n = j)\hat{\beta}(z_n = j) \tag{2.17}$$

$$\xi(z_{n-1} = k, z_n = j) = \frac{1}{c_n} \hat{\alpha}(z_{n-1} = k) p(z_n = j|z_{n-1} = k) p(x_n|z_n = j) \hat{\beta}(z_n = j) \tag{2.18}$$

2.1.2 Gaussian Distribution

The Gaussian distribution will be used under the context of our method. Its probability density function (for the univariate case) as well as its mean and variance are given below:

$$p(x;\boldsymbol{\theta}) = N(x;\boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] \tag{2.19}$$

where $x \in \mathbb{R}, \quad \mu \in \mathbb{R}, \quad \sigma \in \mathbb{R}^+$.

The mean and variance are:

$$\mathbb{E}[x] = \mu \tag{2.20}$$

$$\text{var}[x] = \sigma^2 \tag{2.21}$$

2.1.3 Sparse Prior

As we mentioned in Subsection 2.1.1, we are interested in evaluating the parameters of the model. If we do not have any prior knowledge on them, we apply a Maximum Likelihood (ML) approach. However, many times we have some prior knowledge over the parameters or we would like to apply a special constraint on them that cannot be captured by just assuming uniformity among all the possible parameter configurations. In our context, we are interested in enforcing some sparseness into the parameter space. Good choices for achieving sparseness are the entropic and the negative-Dirichlet type distributions.

The entropic prior is given by the formula:

$$p(\boldsymbol{\theta}) \propto \prod_j \theta_j^{-\theta_j}$$

while the negative Dirichlet-type one is given by:

$$p(\boldsymbol{\theta}) \propto \prod_j \theta_j^{-\alpha}$$

with: $\alpha > 0$, $\theta_j \in [0,1], \forall j$ and $\sum_j \theta_j = 1$.

The larger α is, the more sparseness it enforces. As it is also shown in Figure 2-2, negative Dirichlet-type prior achieves better sparseness than the entropic one [11].

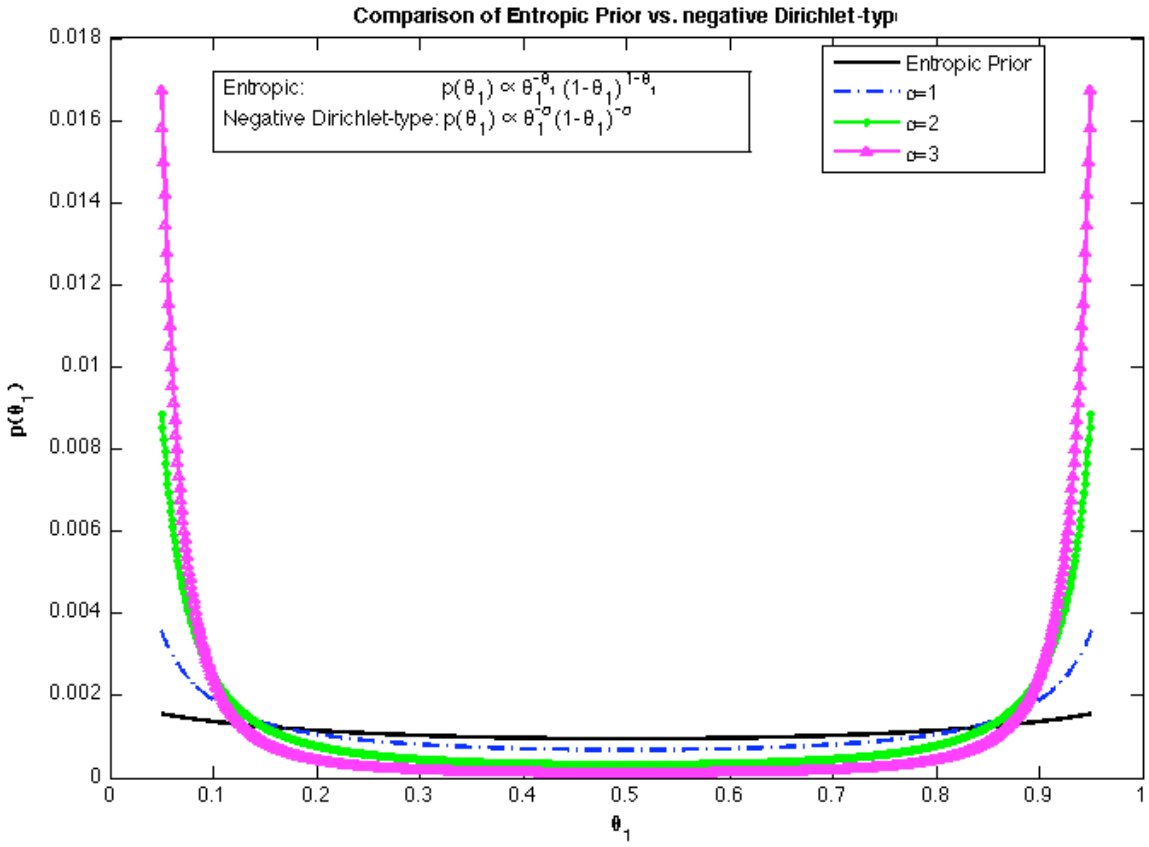


Figure 2-2: **Comparison between entropic and negative Dirichlet-type priors.** This figure illustrates how an entropic and a negative Dirichlet-type prior act on a two-parameter configuration $\boldsymbol{\theta} = [\theta_1, \theta_2]^T$ s.t. $\theta_1 + \theta_2 = 1$. Both the entropic: $p(\boldsymbol{\theta}) \propto \theta_1^{-\theta_1}\theta_2^{-\theta_2} = \theta_1^{-\theta_1}(1-\theta_1)^{-(1-\theta_1)}$ and the negative Dirichlet-type: $p(\boldsymbol{\theta}) \propto \theta_1^{-\alpha}\theta_2^{-\alpha} = \theta_1^{-\alpha}(1-\theta_1)^{-\alpha}$ priors discourage uniform configurations and favor the ones at the extremes, thus imposing sparseness. As shown in figure, the negative Dirichlet priors perform more strongly than the entropic one with greater effects as the value of α grows.

2.2 Notation

In this subsection, we are going to outline the notation that will later be used in the description of our methods. In more detail, we will denote each read with r_n ($n \in \{1, \ldots, N\}$). If, however, we treat different experiments separately, then reads between experiments will not be aggregated and thus in that case $n \in \{1, \ldots, N_t\}$, where $t = \{0, \ldots, C-1\}$ and C: number of different experiments.

Each read r_n is placed at position x_n ($x_n \in \{0, \ldots, M-1\}$) and is generated by an event (component) at position $z_n = j$, where $j \in \{0, \ldots, M-1\}$. (Here, we implicitly assumed that the indexing of the examined region starts from 0 (0-based). Also, we assume that we will have C experiments (whose indexing starts from 0).

The prior probability (weighting) of an event being at position j will be denoted by π_j. Thus, the probability of an event being somewhere in the examined region will be $\sum_{j=0}^{M-1} \pi_j$ (assuming that *a priori* the place of an event will be independent of the place of the others – independence assumption). Since we assume that the reads' generation is due to the presence of related TF binding events, there must be at least one event responsible for that.
This is formally interpreted as:

$$\sum_{j=0}^{M-1} \pi_j = 1 \tag{2.22}$$

We will also denote the probability of read r_n being at position x_n given that is generated by the event at position $z_n = j$ by $p(x_n|z_n = j)$ ($\triangleq H_{x_n,j}$) and the posterior probability of a read belonging to an event (component) given that is placed at position x_n by $p(z_n = j|x_n)$ ($\triangleq \gamma(z_n = j)$).
Following similar reasoning as above, we expect that once we know that a read is generated by an event, it should be placed somewhere in the examined region and that once we know the position of the read we expect that it will be generated by some event (if not coming from the background (noise) model). The above are neatly interpreted as:

$$\sum_{x_n=0}^{M-1} p(x_n|z_n = j) = 1 \tag{2.23}$$

$$\sum_{j=0}^{M-1} p(z_n = j|x_n) = 1 \tag{2.24}$$

, respectively.

In the following, we gather definitions of variables we are going to use into tables for easier reference.

Table 2.1: **Math Notation. Scalar variables.** Different experiments are treated jointly.

N	number of reads
M	number of components and length of examined region
d	maximum distance between two adjacent reads where the dependency assumption holds
$\theta \triangleq p(m_n = \mathtt{m})$	probability that a read comes from the mixture model
$\theta_1 \triangleq p(m_n = \mathtt{m}\|m_{n-1} = \mathtt{b})$	probability that a read comes from the mixture model if the previous one comes from the background model

$1-\theta_2 \triangleq p(m_n = \mathtt{b}\|m_{n-1} = \mathtt{m})$	probability that a read comes from the background model if the previous one comes from the mixture model
$H_{x_n,j} \triangleq p(x_n\|z_n = j, \mathbf{\Theta}^{(i)})$ $H_{x_n,j} \triangleq p(x_n\|z_n = j)$	probability of n^{th} read being at position x_n given that it belongs to component j (at iteration i)
$\gamma(z_n = j) \triangleq p(z_n = j\|x_n, \mathbf{\Theta}^{(i-1)})$ $\gamma(z_n = j) \triangleq p(z_n = j\|x_n)$	posterior probability of n^{th} read belonging to component j given that it is at position x_n (at iteration $i-1$)
$\zeta(z_n = j) \triangleq p(z_n = j\|\mathbf{x}, \mathbf{\Theta}^{(i-1)})$ $\zeta(z_n = j) \triangleq p(z_n = j\|\mathbf{x})$	posterior probability of n^{th} read belonging to component j given that all the reads are at positions $\mathbf{x}$ (at iteration $i-1$)
$\xi(z_{n-1} = k, z_n = j) \triangleq p(z_{n-1} = k, z_n = j\|\mathbf{x}, \mathbf{\Theta}^{(i-1)})$ $\xi(z_{n-1} = k, z_n = j) \triangleq p(z_{n-1} = k, z_n = j\|\mathbf{x})$	posterior probability of $(n-1)^{th}$ and n^{th} reads belonging to components k and j, respectively, given that all the reads are at positions $\mathbf{x}$ (at iteration $i-1$)
$\alpha(z_n = j) \triangleq p(x_1, \ldots, x_n, z_n = j\|\mathbf{\Theta}^{(i-1)})$ $\alpha(z_n = j) \triangleq p(x_1, \ldots, x_n, z_n = j)$	Probability that the first n reads are at positions $\{x_1, \ldots, x_n\}$ and the n^{th} one belongs to component j
$\beta(z_n = j) \triangleq p(x_{n+1}, \ldots, x_N\|z_n = j, \mathbf{\Theta}^{(i-1)})$ $\beta(z_n = j) \triangleq p(x_{n+1}, \ldots, x_N\|z_n = j)$	Probability that the last $N-n$ reads are at positions $\{x_{n+1}, \ldots, x_N\}$ given that the n^{th} one belongs to component j

$\pi_j \triangleq p(z_n = j\|\mathbf{\Theta}^{(i)})$ $\pi_j \triangleq p(z_n = j)$	prior probability of n^{th} read belonging to component j (n^{th} read can be any read cause we do not know its position (at iteration i)
$A_{k,j} \triangleq p(z_n = j\|z_{n-1} = k, \mathbf{\Theta}^{(i)})$ $A_{k,j} \triangleq p(z_n = j\|z_{n-1} = k)$	transition probability of n^{th} read belonging to component j given that $(n-1)^{th}$ read belongs to component k (at iteration i)

Table 2.2: **Math Notation. Scalar variables.** Different experiments are treated separately.

C	number of different experiments
N_t	number of reads of t^{th} experiment
M	number of components and length of examined region
$H_{x_{t,n},j} \triangleq p(x_{t,n}\|z_{t,n} = j, \mathbf{\Theta}^{(i)})$ $H_{x_{t,n},j} \triangleq p(x_{t,n}\|z_{t,n} = j)$	probability of n^{th} read being at position x_n (at time point t) given that it belongs to component j (at iteration i)
$\gamma(z_{t,n} = j) \triangleq p(z_{t,n} = j\|x_{t,n}, \mathbf{\Theta}^{(i-1)})$ $\gamma(z_{t,n} = j) \triangleq p(z_{t,n} = j\|x_{t,n})$	posterior probability of n^{th} read belonging to component j (at time point t) given that it is at position x_n (at iteration $i-1$)

$\zeta(z_{t,n} = j) \triangleq p(z_{t,n} = j \mid \mathbf{x}_t, \mathbf{\Theta}^{(i-1)})$ $\zeta(z_{t,n} = j) \triangleq p(z_{t,n} = j \mid \mathbf{x}_t)$	posterior probability of n^{th} read belonging to component j (at time point t) given that all the reads are at positions $\mathbf{x}_t$ (at iteration $i-1$)
$\xi(z_{t,n-1} = k, z_{t,n} = j) \triangleq p(z_{t,n-1} = k, z_{t,n} = j \mid \mathbf{x}_t, \mathbf{\Theta}^{(i-1)})$ $\xi(z_{t,n-1} = k, z_{t,n} = j) \triangleq p(z_{t,n-1} = k, z_{t,n} = j \mid \mathbf{x}_t)$	posterior probability of $(n-1)^{th}$ and n^{th} reads belonging to components k and j, respectively, (at time point t) given that all the reads are at positions $\mathbf{x}_t$ (at iter $i-1$)
$\alpha(z_{t,n} = j) \triangleq p(x_{t,1}, \ldots, x_{t,n}, z_{t,n} = j \mid \mathbf{\Theta}^{(i-1)})$ $\alpha(z_{t,n} = j) \triangleq p(x_{t,1}, \ldots, x_{t,n}, z_{t,n} = j))$	Probability that the first n reads (at time point t) are at positions $\{x_1, \ldots, x_n\}$ and the n^{th} one belongs to component j (at iteration $i-1$)
$\beta(z_{t,n} = j) \triangleq p(x_{t,n+1}, \ldots, x_{t,N_t} \mid z_{t,n} = j, \mathbf{\Theta}^{(i-1)})$ $\beta(z_{t,n} = j) \triangleq p(x_{t,n+1}, \ldots, x_{t,N_t} \mid z_{t,n} = j)$	Probability that the last $N_t - n$ reads (at time point t) are at positions $\{x_{n+1}, \ldots, x_N\}$ given that the n^{th} one belongs to component j (at iteration $i-1$)
$\pi_{t,j} \triangleq p(z_{t,n} = j \mid \mathbf{\Theta}^{(i)})$ $\pi_{t,j} \triangleq p(z_{t,n} = j)$	prior probability of n^{th} read belonging to component j at time point t (at iteration i)
$A_{t,k,j} \triangleq p(z_{t,n} = j \mid z_{t,n-1} = k, \mathbf{\Theta}^{(i)})$ $A_{t,k,j} \triangleq p(z_{t,n} = j \mid z_{t,n-1} = k)$	transition probability of n^{th} read belonging to component j (at time point t) given that $(n-1)^{th}$ read belongs to component k (at iteration i)

Table 2.3: **Math Notation. Vector variables.** Different experiments are treated jointly.

$\mathbf{x} =$ $\{x_1, \ldots, x_N\}$	observed variables (vector of scalar variables) Positions of reads
$\mathbf{z} =$ $\{z_1, \ldots, z_N\}$	observed variables (vector of scalar variables) Read memberships
$\boldsymbol{\pi} =$ $\{\pi_0, \ldots, \pi_{M-1}\}$	prior probability of components
$p(\mathbf{x}\|\mathbf{z}, \boldsymbol{\Theta}^{(i)})$ $p(\mathbf{x}\|\mathbf{z})$	probability of reads $\{r_1, \ldots, r_N\}$ being at positions $\mathbf{x} = \{x_1, \ldots, x_N\}$ given that they belong to components $\mathbf{z} = \{z_1, \ldots, z_N\}$
$p(\mathbf{z}\|\mathbf{x}, \boldsymbol{\Theta}^{(i-1)})$ $p(\mathbf{z}\|\mathbf{x})$	posterior probability of reads $\{r_1, \ldots, r_N\}$ belonging to components $\mathbf{z} = \{z_1, \ldots, z_N\}$ given that they are at positions $\mathbf{x} = \{x_1, \ldots, x_N\}$
$p(\mathbf{z}\|\boldsymbol{\Theta}^{(i)})$ $p(\mathbf{z})$	prior probability of reads $\{r_1, \ldots, r_N\}$ belonging to components $\mathbf{z} = \{z_1, \ldots, z_N\}$

Table 2.4: **Math Notation. Vector variables.** Different experiments are treated separately.

$\mathbf{x}_t =$ $\{x_{t,1}, \ldots, x_{t,N_t}\}$	observed variables (vector of scalar variables) at time point t Positions of reads at time point t
$\mathbf{z}_t =$ $\{z_{t,1}, \ldots, z_{t,N_t}\}$	observed variables (vector of scalar variables) at time point t Read memberships at time point t
$\boldsymbol{\pi}_t =$ $\{\pi_{t,0}, \ldots, \pi_{t,M-1}\}$	prior probability of components
$p(\mathbf{x}_t \mid \mathbf{z}_t, \boldsymbol{\Theta}^{(i)})$ $p(\mathbf{x}_t \mid \mathbf{z}_t)$	probability of reads $\{r_1, \ldots, r_{N_t}\}$ being at positions $\mathbf{x}_t = \{x_1, \ldots, x_{N_t}\}$ (at time point t) given that they belong to components $\mathbf{z}_t = \{z_1, \ldots, z_{N_t}\}$
$p(\mathbf{z}_t \mid \mathbf{x}_t, \boldsymbol{\Theta}^{(i-1)})$ $p(\mathbf{z}_t \mid \mathbf{x}_t)$	posterior probability of reads $\{r_1, \ldots, r_{N_t}\}$ belonging to components $\mathbf{z}_t = \{z_1, \ldots, z_{N_t}\}$ (at time point t) given that they are at positions $\mathbf{x}_t = \{x_1, \ldots, x_{N_t}\}$
$p(\mathbf{z}_t \mid \boldsymbol{\Theta}^{(i)})$ $p(\mathbf{z}_t)$	prior probability of reads $\{r_1, \ldots, r_{N_t}\}$ (at time point t) belonging to components $\mathbf{z}_t = \{z_1, \ldots, z_{N_t}\}$

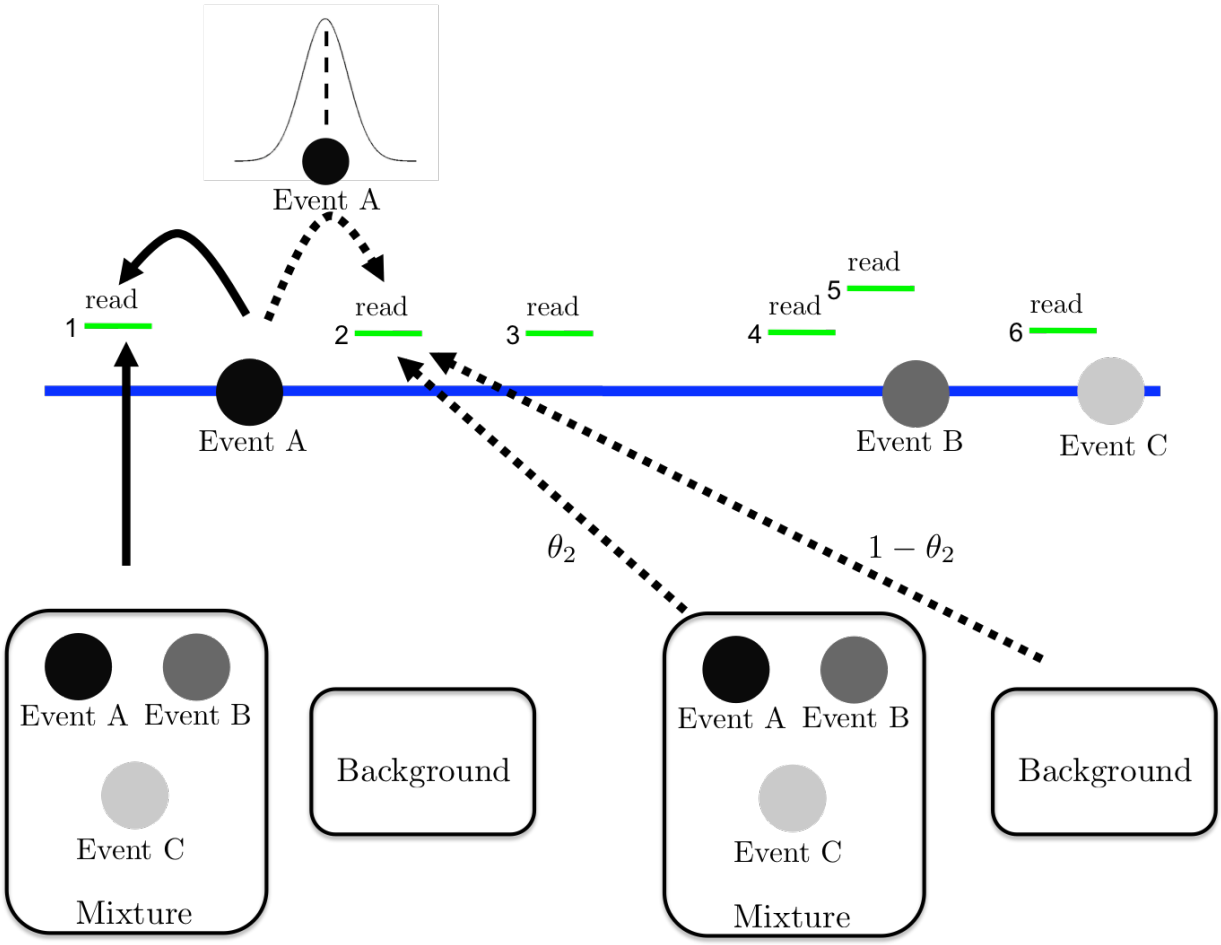
Table 2.5: **Math Notation. Matrix variables.**

$\mathbf{X} =$ $\{\mathbf{x}_0, \ldots, \mathbf{x}_{C-1}\}$	observed variables (vector of vector variables, matrix that is) Positions of reads at time points $t \in \{0, \ldots, C-1\}$
$\mathbf{Z} =$ $\{\mathbf{z}_0, \ldots, \mathbf{z}_{C-1}\}$	hidden variables (vector of vector variables, matrix that is) Read memberships at time points $t \in \{0, \ldots, C-1\}$
$\mathbf{\Pi} =$ $\{\boldsymbol{\pi}_0, \ldots, \boldsymbol{\pi}_{C-1}\}$	Prior probability of components at time points $t \in \{0, \ldots, C-1\}$
$p(\mathbf{X}\|\mathbf{Z}, \mathbf{\Theta}^{(i)})$ $p(\mathbf{X}\|\mathbf{Z})$	probability of reads being at positions $\mathbf{X} = \{\mathbf{x}_0, \ldots, \mathbf{x}_{C-1}\}$ given that they belong to components $\mathbf{Z} = \{\mathbf{z}_0, \ldots, \mathbf{z}_{C-1}\}$
$p(\mathbf{Z}\|\mathbf{X}, \mathbf{\Theta}^{(i-1)})$ $p(\mathbf{Z}\|\mathbf{X})$	posterior probability of reads belonging to components $\mathbf{Z} = \{\mathbf{z}_0, \ldots, \mathbf{z}_{C-1}\}$ given that they are at positions $\mathbf{X} = \{\mathbf{x}_0, \ldots, \mathbf{x}_{C-1}\}$
$p(\mathbf{Z}\|\mathbf{\Theta}^{(i)})$ $p(\mathbf{Z})$	prior probability of reads belonging to components $\mathbf{Z} = \{\mathbf{z}_0, \ldots, \mathbf{z}_{C-1}\}$

2.3 Spatial Coupling

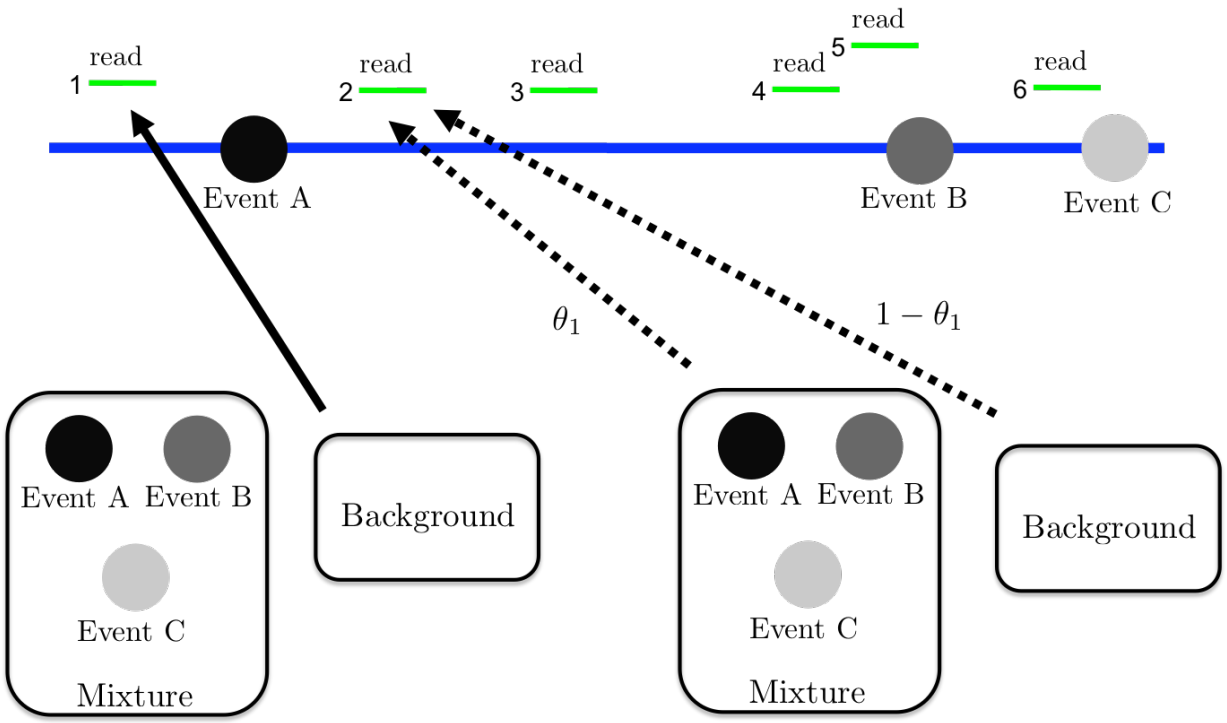
After having provided the necessary notation, we will go on presenting the Spatial Coupling method. We will test the hypothesis that the Spatial Coupling achieves better spatial resolution, sensitivity and specificity by exploiting the fact that proximal reads may be generated by the same binding event. So, we will enforce a dependence of the assignment of a read to a binding event to the assignment of its closest read (to a binding event). We will also assume that the assignment of a read will follow a Gaussian distribution (for reasons that will become apparent later on) with mean the assignment of the previous read, thus favoring it to be derived from the same (or a very close) binding event since more distant events will correspond to very low transition probability values.
In addition, we will further propose an error (background) model, trying to model the fact that a read may not be the outcome of a binding event but the result of non-specific binding. In more detail, when two reads are very close together, then it is likely that the read will be assigned on the same component (as its neighbor). On the other hand, if the previous read comes from the background model, then the current one will be coming from the background with some probability $(1 - \theta_1)$ as well and if not, its probability of belonging to a component will be solely depending on the position of this read (regardless of the position of its neighbor). See Figure 2-3 for a visualization of the workflow of this method.

Before going into mathematical details, we will make a few more assumptions pertaining to the spatial case. We will denote with m_n ($n \in \{1, \ldots, N\}$), the model that a read r_n comes from. In more detail, we assume that a read can either come from the mixture model (comprising of the M components) or the background (noise) model. So, m_n will be a binary random variable taking on the values $m_n \in \{\mathtt{m}, \mathtt{b}\}$ ($\mathtt{m}$: for the mixture, $\mathtt{b}$: for the background (noise) model). We will further assume that the (transition) probability of a read r_n coming from the mixture model while the read r_{n-1} comes from the background model and vice versa will both be Bernoulli distributed.



(a) **Read r_{n-1} comes from the mixture model.**

Figure 2-3



(b) **Read r_{n-1} comes from the background model.**

Figure 2-3: **Spatial Coupling workflow.**
This figure presents the workflow of the Spatial Coupling method. The main idea of this method is that the assignment z_n of a read r_n to a component is dependent on the assignment z_{n-1} of its previous (neighboring) read r_{n-1}. For the purpose of this figure, we assume n to be 2.
In 2-3a, it is assumed that read 1 is coming from the mixture model and more specifically from Event A. Therefore, read 2 will be coming from Event A with a probability dependent on how likely it is to be derived from the mixture model (θ_2) and how close read 2 to Event A is (the distance of a read to a component is assumed to follow a Gaussian distribution). If, however, read 1 comes from the background model (2-3b), then the assignment of read 2 is dependent only on how likely it is for read 2 to be coming from the mixture model (θ_1) provided that the previous was derived from the background.
Blue bold line indicates the genome. Black solid arrows indicate the state (event) that a read comes from. Black dashed arrows indicate the probability that a read comes from a state (event).

That is,

$$p(m_n = \mathtt{m}|m_{n-1} = \mathtt{b}) = \theta_1 \quad (2.25)$$

$$p(m_n = \mathtt{b}|m_{n-1} = \mathtt{m}) = 1 - \theta_2 \quad (2.26)$$

Immediately following that:

$$p(m_n = \mathtt{b}|m_{n-1} = \mathtt{b}) = 1 - \theta_1$$

$$p(m_n = \mathtt{m}|m_{n-1} = \mathtt{m}) = \theta_2$$

The aforementioned can be more clearly depicted on Figure 2-4.

For the sake of convenience, we will represent the background model as $j = M$. So, when we denote that $z_n = M$, we will mean that this read comes from the background model.

Previously, we defined that when a read r_n comes from one of the M components (thus, from the mixture model), it follows a distribution: $p(x_n|z_n = j, m_n = \mathtt{m}) = p(x_n|z_n = j) = H_{x_n,j} \quad , n \in \{1, \ldots, N\}, j \in \{0, \ldots, M-1\}$. Now, we further define that when a read comes from the background model, then it follows a uniform distribution: $p(x_n|z_n = M, m_n = \mathtt{b}) = p(x_n|z_n = M) = \frac{1}{M} \quad , n \in \{1, \ldots, N\}$. That is, a read coming from the background model can be generated with an equal probability regardless of its position on the genome.

This is formally written as:

$$p(x_n|z_n = j) = \begin{cases} H_{x_n,j} & , j \in \{0, \ldots, M-1\} \\ \frac{1}{M} & , j = M \end{cases} \quad (2.27)$$

We silently implied that there is a (spatial) dependency between two adjacent reads. In other words, if two reads are located not too far away from each other ($|x_n - x_{n-1}| \leq d$), then the membership of the previous read affects the membership of the current one, as shown in Figure 2-5.

For each pair of reads, we define the following transition probability: $p(z_n = j|z_{n-1} = k)$: When read r_{n-1} comes from the mixture model: $m_{n-1} = \mathtt{m} \Rightarrow k \in \{0, \ldots, M-1\}$, then we

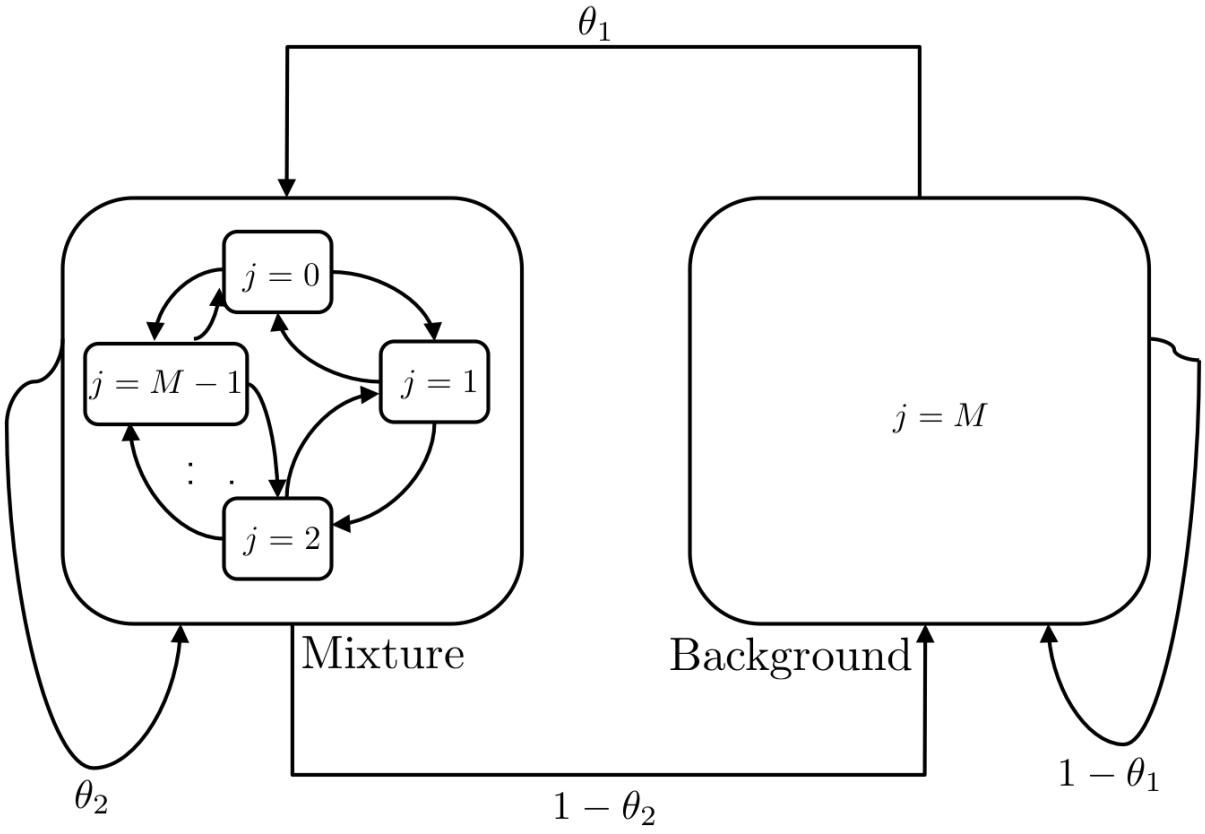


Figure 2-4: **Transition probabilities between the background and the mixture model.**
This figure illustrates the transitions between the background and the mixture model. When a read r_{n-1} comes from a model (mixture or background), it can either transit to another or stay in the same. Once in the background model, the read follows a uniform distribution while once in the mixture model, it follows a distribution dependent on the specific component of the mixture model that it comes from.

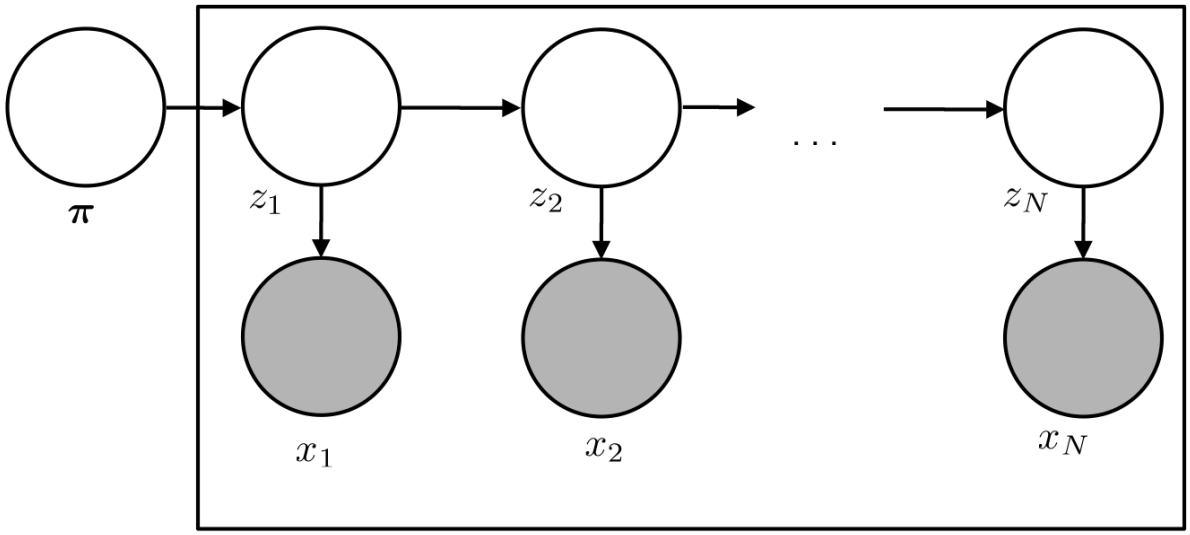


Figure 2-5: **Spatial coupling between reads.**
This figure illustrates the dependency assumption between adjacent reads. In more detail, assuming sorted (w.r.t. position) reads, we expect that the assignment of a read to an event (component) will affect the assignment of the next read to an event (component) unless they are too far away from each other.

have:

If $j \in \{0, \ldots, M-1\}$:

$$p(z_n = j | z_{n-1} = k) = \theta_2 \cdot p(z_n = j | z_{n-1} = k, m_n = \mathtt{m}, m_{n-1} = \mathtt{m})$$

That is, the probability that a read belongs to component j given that the previous one belongs to component k, is the product of the probability that both reads come from the mixture model and some transition probability (defined subsequently).

Similarly, if $j = M$:

$$p(z_n = M | z_{n-1} = k) = 1 - \theta_2$$

which is essentially the probability of moving from the mixture to the background model.

Lastly, if read r_{n-1} comes from the background model: $m_{n-1} = \mathtt{b} \Rightarrow k = M$, then we have:

If $j \in \{0, \ldots, M-1\}$:

$$p(z_n = j | z_{n-1} = k) = \theta_1 \cdot p(z_n = j | z_{n-1} = k, m_n = \mathtt{m}, m_{n-1} = \mathtt{b})$$

which is the product of the probability that we move from the background to the mixture model times some transition probability (defined later),
while if $j = M$:

$$p(z_n = M | z_{n-1} = M) = 1 - \theta_1$$

The above are summarized following:

$$p(z_n = j | z_{n-1} = k) = \begin{cases} \theta_2 \cdot p(z_n = j | z_{n-1} = k, m_n = \mathtt{m}, m_{n-1} = \mathtt{m}) & , k, j \in \{0, \ldots, M-1\} \\ 1 - \theta_2 & , k \in \{0, \ldots, M-1\}, j = M \\ \theta_1 \cdot p(z_n = j | z_{n-1} = M, m_n = \mathtt{m}, m_{n-1} = \mathtt{b}) & , k = M, j \in \{0, \ldots, M-1\} \\ 1 - \theta_1 & , k = M, j = M \end{cases} \tag{2.28}$$

For the first read, since it is not dependent on any read, we define the probability that it comes from a mixture model as: $p(m_1 = \mathtt{m}) = \theta$. In addition, we define the probability $\epsilon_j \triangleq p(z_1 = j | m_1 = \mathtt{m}) \quad , j \in \{0, \ldots, M-1\}$. Also, we have $p(z_1 = M | m_1 = \mathtt{m}) = 0$, while $p(z_1 = M | m_1 = \mathtt{b}) = 1$ and $p(z_1 = j | m_1 = \mathtt{b}) = 0 \quad , j \in \{0, \ldots, M-1\}$.
So, the probability $p(z_1 = j)$ is defined as:

$$p(z_1 = j) = \begin{cases} \theta \cdot \epsilon_j & , j \in \{0, \ldots, M-1\} \\ 1 - \theta & , j = M \end{cases} \tag{2.29}$$

where $\sum_{j=0}^{M-1} \epsilon_j = 1$.

Now for the choice of the distribution for $p(z_n = j | z_{n-1} = k, m_n = \mathtt{m}, m_{n-1} = \mathtt{m})$ and $p(z_n = j | z_{n-1} = M, m_n = \mathtt{m}, m_{n-1} = \mathtt{b})$, we have considered the following alternatives:

i. a Poisson distribution

It is a discrete variable distribution (as in our case) but it is of infinite support (while

here the support is M) and the variance increases linearly with the mean.

ii. a Binomial distribution

It is a discrete variable distribution (as in our case), it is of finite support but the variance depends on the mean. Medium means correspond to bigger variances, while very small or big means correspond to low variances.

iii. a Gaussian distribution

It is a continuous variable distribution. However, it has also been used in the discrete case without much arbitrariness. It is of infinite support, however $\approx 95\%$ of the data will be gathered in the interval $[\mu - 2\sigma, \mu + 2\sigma]$. The advantage of the Gaussian is that the variance can be constant and independent of the mean.

We have decided to adopt the Gaussian distribution since there is no indication that the variance around the component of the previous read will be dependent on its position along the genome.

So, we define (for sufficiently close reads: $|x_n - x_{n-1}| \leq d$):

$$p(z_n = j | z_{n-1} = k, m_n = \mathtt{m}, m_{n-1} = \mathtt{m}) = N(j; k, \sigma^2) \tag{2.30}$$

and

$$p(z_n = j | z_{n-1} = M, m_n = \mathtt{m}, m_{n-1} = \mathtt{b}) = N(j; x_n, \sigma^2) \tag{2.31}$$

where $k, j \in \{0, \ldots, M-1\}$ and σ^2 is to be learned from the data.
Here we implicitly assumed that $|x_n - x_{n-1}| \leq d$.

Equation 2.30 simply says that we expect the component of a read (z_n) to be around the component of the previous read ($z_{n-1} = k$). If the previous read comes from the background model though, then we expect that the component of this read (z_n) will be around its position (x_n) (equation 2.31).

In the case, however, where two adjacent reads are further apart ($|x_n - x_{n-1}| > d$), then

we cannot make any strong dependency assumptions.

Thus, here we assume that:

$$p(z_n = j|z_{n-1} = k, m_n = \mathtt{m}, m_{n-1} = \mathtt{m}) = p(z_n = j|z_{n-1} = M, m_n = \mathtt{m}, m_{n-1} = \mathtt{b}) = \frac{1}{M} \tag{2.32}$$

where $k, j \in \{0, \ldots, M-1\}$.

In other words, the membership of the previous read does not affect the membership of the current one.

After gathering all the aforementioned, we have:

If $|x_n - x_{n-1}| \leq d$:

$$p(z_n = j|z_{n-1} = k) = \begin{cases} \theta_2 \cdot N(j; k, \sigma^2) & , k, j \in \{0, \ldots, M-1\} \\ 1 - \theta_2 & , k \in \{0, \ldots, M-1\}, j = M \\ \theta_1 \cdot N(j; x_n, \sigma^2) & , k = M, j \in \{0, \ldots, M-1\} \\ 1 - \theta_1 & , k = M, j = M \end{cases} \tag{2.33}$$

while, if $|x_n - x_{n-1}| > d$:

$$p(z_n = j|z_{n-1} = k) = \begin{cases} \theta_2 \cdot \frac{1}{M} & , k, j \in \{0, \ldots, M-1\} \\ 1 - \theta_2 & , k \in \{0, \ldots, M-1\}, j = M \\ \theta_1 \cdot \frac{1}{M} & , k = M, j \in \{0, \ldots, M-1\} \\ 1 - \theta_1 & , k = M, j = M \end{cases} \tag{2.34}$$

Now that we have made all the necessary assumptions, we can go on deriving the Expectation Maximization (EM) equations.

It is known, that in the EM setting we are interested in maximizing the expectation of the complete data (both observed and hidden ones) over the posterior distribution of the hidden variables: $\mathbb{E}_{p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\Theta}^{(i-1)})} \left[\log p(\mathbf{X}, \mathbf{Z}; \boldsymbol{\Theta}^{(i)})\right]$

For less cluttering, we will hereby omit the parameter symbol $\boldsymbol{\Theta}^{(i-1)}$ of a distribution and will only indicate it if it is not clear from the context. E.g. $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\Theta}^{(i-1)})$ will be represented

as $p(\mathbf{Z}|\mathbf{X})$, $p(\mathbf{X},\mathbf{Z};\mathbf{\Theta}^{(i)})$ as $p(\mathbf{X},\mathbf{Z})$ and so on. In general, the posterior probabilities of hidden variables given observed ones are conditioned on the parameters of the previous iteration ($\mathbf{\Theta}^{(i-1)}$), while all the others on the parameters of the current iteration ($\mathbf{\Theta}^{(i)}$).

We will deal first with the case of a single condition (time point) set and then the transition to multiple conditions will be straightforward.

It is:

$$\begin{aligned} Q(\mathbf{\Theta}^{(i)};\mathbf{\Theta}^{(i-1)}) &= \sum_{\mathbf{z}} p(\mathbf{z}|\mathbf{x},\mathbf{\Theta}^{(i-1)}) \log p(\mathbf{x},\mathbf{z};\mathbf{\Theta}^{(i)}) \Leftrightarrow \\ &= \sum_{j=0}^{M} \underbrace{p(z_1=j|\mathbf{x})}_{\zeta(z_1=j)} \log p(z_1=j) \\ &+ \sum_{n=1}^{N}\sum_{j=0}^{M} \underbrace{p(z_n=j|\mathbf{x})}_{\zeta(z_n=j)} \log p(x_n|z_n=j) \\ &+ \sum_{n=2}^{N}\sum_{k=0}^{M}\sum_{j=0}^{M} \underbrace{p(z_{n-1}=k,z_n=j|\mathbf{x})}_{\xi(z_{n-1}=k,z_n=j)} \log p(z_n=j|z_{n-1}=k) \end{aligned} \quad (2.35)$$

which due to Equations 2.27, 2.29, 2.33, 2.34 leads to:

$$\begin{aligned} Q(\mathbf{\Theta}^{(i)};\mathbf{\Theta}^{(i-1)}) &= \sum_{j=0}^{M-1} \zeta(z_1=j)\log(\theta\epsilon_j) + \zeta(z_1=M)\log(1-\theta) \\ &+ \sum_{n=1}^{N}\left\{\sum_{j=0}^{M-1}\zeta(z_n=j)\log H_{x_n,j} + \zeta(z_n=M)\log\frac{1}{M}\right\} \\ &+ \sum_{n'\in\{n':|x_{n'}-x_{n'-1}|\le d\}} \Big\{S_1\Big\} \\ &+ \sum_{n''\in\{n'':|x_{n''}-x_{n''-1}|>d\}} \Big\{S_2\Big\} \end{aligned} \quad (2.36)$$

where:

$$S_1 = \sum_{k=0}^{M-1}\left[\sum_{j=0}^{M-1}\xi(z_{n'-1}=k,z_{n'}=j|\mathbf{x})\log\left(N(j;k,\sigma^2)\theta_2\right) + \xi(z_{n'-1}=k,z_{n'}=M|\mathbf{x})\log(1-\theta_2)\right]$$

$$+ \quad \sum_{j=0}^{M-1} \xi(z_{n'-1} = M, z_{n'} = j|\mathbf{x}) \log\left(N(j; x_{n'}, \sigma^2)\theta_1\right) + \xi(z_{n'-1} = M, z_{n'} = M|\mathbf{x}) \log(1-\theta_1)$$

$$\begin{aligned} S_2 &= \sum_{k=0}^{M-1} \left[\sum_{j=0}^{M-1} \xi(z_{n''-1} = k, z_{n''} = j|\mathbf{x}) \log\left(\frac{1}{M}\theta_2\right) + \xi(z_{n''-1} = k, z_{n''} = M|\mathbf{x}) \log(1-\theta_2) \right] \\ &+ \sum_{j=0}^{M-1} \xi(z_{n''-1} = M, z_{n''} = j|\mathbf{x}) \log\left(\frac{1}{M}\theta_1\right) + \xi(z_{n''-1} = M, z_{n''} = M|\mathbf{x}) \log(1-\theta_1) \end{aligned}$$

and $\sum_{j=0}^{M-1} \epsilon_j = 1$ with $\epsilon_j \geq 0, \forall j \in \{0, \ldots, M-1\}$.

The parameters to be learned are: $\boldsymbol{\epsilon} = \{\epsilon_0, \ldots, \epsilon_{M-1}\}, \theta, \theta_1, \theta_2, \sigma^2 \triangleq v$.

All we need to do is differentiate 2.36 with respect to $\boldsymbol{\epsilon}, \theta, \theta_1, \theta_2, v$ in succession.

We have:

$$\begin{aligned} \frac{\partial Q}{\partial \epsilon_j} &= 0 \overset{\sum_{j=0}^{M-1} \epsilon_j = 1}{\Rightarrow} \\ \epsilon_j &= \frac{\zeta(z_1 = j)}{\sum_{k=0}^{M-1} \zeta(z_1 = k)} \Leftrightarrow \\ &= \frac{\zeta(z_1 = j)}{1 - \zeta(z_1 = M)} \quad , j \in \{0, \ldots, M-1\} \end{aligned} \tag{2.37}$$

In other words, the prior weighting of the component in the first read is nothing more than the relative responsibility of this component given all the observed variables.

Similarly:

$$\begin{aligned} \frac{\partial Q}{\partial \theta} &= 0 \Rightarrow \\ \theta &= \frac{\sum_{j=0}^{M-1} \zeta(z_1 = j)}{\underbrace{\sum_{j=0}^{M} \zeta(z_1 = j)}_{=1}} \Leftrightarrow \end{aligned}$$

$$= \sum_{j=0}^{M-1} \zeta(z_1 = j) = 1 - \zeta(z_1 = M)$$

which is simply the relative responsibility of staying in the mixture model.

After differentiating with respect to θ_1 and θ_2, we also have:

$$\begin{aligned} \theta_1 &= \frac{\sum_{n=2}^{N}\sum_{j=0}^{M-1}\xi(z_{n-1}=M, z_n=j)}{\sum_{n=2}^{N}\sum_{j=0}^{M}\xi(z_{n-1}=M, z_n=j)} \Leftrightarrow \\ &= 1 - \frac{\sum_{n=2}^{N}\xi(z_{n-1}=M, z_n=M)}{\sum_{j=0}^{M}\sum_{n=2}^{N}\xi(z_{n-1}=M, z_n=j)} \end{aligned}$$

and

$$\theta_2 = \frac{\sum_{n=2}^{N}\sum_{k=0}^{M-1}\sum_{j=0}^{M-1}\xi(z_{n-1}=k, z_n=j)}{\sum_{n=2}^{N}\sum_{k=0}^{M-1}\sum_{j=0}^{M}\xi(z_{n-1}=k, z_n=j)}$$

It is: $\sum_{k=0}^{M-1}\sum_{j=0}^{M-1}\xi(z_{n-1}=k, z_n=j) = 1 + \xi(z_{n-1}=M, z_n=M) - \sum_{j=0}^{M}\xi(z_{n-1}=M, z_n=j) - \sum_{k=0}^{M}\xi(z_{n-1}=k, z_n=M)$ and $\sum_{k=0}^{M-1}\sum_{j=0}^{M}\xi(z_{n-1}=k, z_n=j) = 1 - \sum_{j=0}^{M}\xi(z_{n-1}=M, z_n=j)$.
So:

$$\begin{aligned} \theta_2 &= \frac{\sum_{n=2}^{N}\left[1 + \xi(z_{n-1}=M, z_n=M) - \sum_{j=0}^{M}\xi(z_{n-1}=M, z_n=j) - \sum_{k=0}^{M}\xi(z_{n-1}=k, z_n=M)\right]}{\sum_{n=2}^{N}\left[1 - \sum_{j=0}^{M}\xi(z_{n-1}=M, z_n=j)\right]} \Leftrightarrow \\ &= 1 - \frac{\sum_{k=0}^{M-1}\sum_{n=2}^{N}\xi(z_{n-1}=k, z_n=M)}{N - \sum_{j=0}^{M}\sum_{n=2}^{N}\xi(z_{n-1}=M, z_n=j)} \end{aligned}$$

Lastly, differentiating w.r.t. v $\left(\text{keeping in mind that } \frac{\partial N(x;\mu,v)}{\partial v} = \frac{1}{2}\cdot\frac{1}{v^2}\cdot[(x-\mu)^2 - v]\, N(x;\mu,v)\right)$

gives us:

$$
\begin{aligned}
v &= \frac{\sum_{n'\in\mathcal{S}}\sum_{j=0}^{M-1}\left[\sum_{k=0}^{M-1}\xi(z_{n'-1}=k,z_{n'}=j)(j-k)^2+\xi(z_{n'-1}=M,z_{n'}=j)(j-x_{n'})^2\right]}{\sum_{n'\in\mathcal{S}}\underbrace{\sum_{j=0}^{M-1}\sum_{k=0}^{M}\xi(z_{n'-1}=k,z_{n'}=j)}_{1-\sum_{k=0}^{M}\xi(z_{n'-1}=k,z_{n'}=M)}} \Leftrightarrow \\
&= \frac{\sum_{n'\in\mathcal{S}}\left[\sum_{j=0}^{M-1}\sum_{k=0}^{M-1}\xi(z_{n'-1}=k,z_{n'}=j)(j-k)^2+\sum_{j=0}^{M-1}\xi(z_{n'-1}=M,z_{n'}=j)(j-x_{n'})^2\right]}{\sum_{n'\in\mathcal{S}}\left[1-\sum_{k=0}^{M}\xi(z_{n'-1}=k,z_{n'}=M)\right]} \Leftrightarrow \\
&= \frac{\sum_{k=0}^{M-1}\sum_{j=0}^{M-1}\sum_{n'\in\mathcal{S}}\xi(z_{n'-1}=k,z_{n'}=j)(j-k)^2+\sum_{j=0}^{M-1}\sum_{n'\in\mathcal{S}}\xi(z_{n'-1}=M,z_{n'}=j)(j-x_{n'})^2}{|\mathcal{S}|-\sum_{k=0}^{M}\sum_{n'\in\mathcal{S}}\xi(z_{n'-1}=k,z_{n'}=M)}
\end{aligned}
$$

where $\mathcal{S}=\{n':|x_{n'}-x_{n'-1}|\leq d\}$ and $|\mathcal{S}|$ is the *cardinality* of the set, that is, the number of its members.

Gathering all the equations for convenience, we have:

$$\epsilon_j = \frac{\zeta(z_1=j)}{1-\zeta(z_1=M)} \quad ,j\in\{0,\ldots,M-1\} \tag{2.38}$$

$$\theta = 1-\zeta(z_1=M) \tag{2.39}$$

$$\theta_1 = 1-\frac{\sum_{n=2}^{N}\xi(z_{n-1}=M,z_n=M)}{\sum_{j=0}^{M}\sum_{n=2}^{N}\xi(z_{n-1}=M,z_n=j)} \tag{2.40}$$

$$\theta_2 = 1-\frac{\sum_{k=0}^{M-1}\sum_{n=2}^{N}\xi(z_{n-1}=k,z_n=M)}{N-\sum_{j=0}^{M}\sum_{n=2}^{N}\xi(z_{n-1}=M,z_n=j)} \tag{2.41}$$

$$v=\frac{\sum_{k=0}^{M-1}\sum_{j=0}^{M-1}\sum_{n'\in\mathcal{S}}\xi(z_{n'-1}=k,z_{n'}=j)(j-k)^2+\sum_{j=0}^{M-1}\sum_{n'\in\mathcal{S}}\xi(z_{n'-1}=M,z_{n'}=j)(j-x_{n'})^2}{|\mathcal{S}|-\sum_{k=0}^{M}\sum_{n'\in\mathcal{S}}\xi(z_{n'-1}=k,z_{n'}=M)} \tag{2.42}$$

Lastly, we find an estimate of π_j by:

$$\pi_j=\frac{1}{N}\sum_{n=1}^{N}p(z_n=j|\mathbf{x})=\frac{1}{N}\sum_{n=1}^{N}\zeta(z_n=j) \tag{2.43}$$

It seems intuitive to take the averaged posterior of z given the observed data ($\mathbf{x}$) to estimate the prior weighting ($\boldsymbol{\pi}$). However, there is a reason for doing that. If we take the

expectation of the statistic $\frac{1}{N}\sum_{n=1}^{N} p(z_n = j|\mathbf{x})$ over the distribution of the hidden variables ($\mathbf{z}$) then we see that it equals the expectation of the (true underlying) prior weighting over the posterior distribution of the hidden variables given the observed ones:

$$\begin{aligned}\mathbb{E}_{p(\mathbf{z})}\left[\frac{1}{N}\sum_{n=1}^{N} p(z_n|\mathbf{x})\right] &= \frac{1}{N}\sum_{z_1=0}^{M}\sum_{z_2=0}^{M}\cdots\sum_{z_N=0}^{M}\left[p(z_1|\mathbf{x}) + \ldots + p(z_N|\mathbf{x})\right] p(z_1,\ldots,z_N) \\ &= \frac{1}{N}\sum_{n=1}^{N}\sum_{z_n=0}^{M} p(z_n|\mathbf{x})p(z_n)\end{aligned}$$

Assuming that the $z_n s$ are identically distributed ($p(z_n|\mathbf{x})p(z_n) = p(z|\mathbf{x})p(z)$), we have that:

$$\mathbb{E}_{p(\mathbf{z})}\left[\frac{1}{N}\sum_{n=1}^{N} p(z_n|\mathbf{x})\right] = \sum_{z=0}^{M} p(z|\mathbf{x})p(z) = \mathbb{E}_{p(z|\mathbf{x})}\left[p(z)\right] = \mathbb{E}_{p(z|\mathbf{x})}\left[\pi_z\right]$$

So, by obtaining more and more data ($N \uparrow$), we expect that the statistic $\frac{1}{N}\sum_{n=1}^{N} p(z_n = j|\mathbf{x})$ will be closer to the true value of π_j.

We can easily generalize the previous results in the multi-condition case (where we have C different conditions (experiments)) Remember from previously that an experiment is indexed with t spanning from 0 to $C-1$ ($t \in \{0,\ldots,C-1\}$.

Now, we have:

$$\begin{aligned}\epsilon_{t,j} &= \frac{\zeta(z_{t,1}=j)}{\sum_{k=0}^{M-1}\zeta(z_{t,1}=k)} \qquad ,t \in \{0,\ldots,C-1\}, j \in \{0,\ldots,M-1\} \\ \theta &= \frac{\sum_{t=0}^{C-1}\sum_{j=0}^{M-1}\zeta(z_{t,1}=j)}{\sum_{t=0}^{C-1}\sum_{j=0}^{M}\zeta(z_{t,1}=j)} \\ \theta_1 &= \frac{\sum_{t=0}^{C-1}\sum_{n=2}^{N_t}\sum_{j=0}^{M-1}\xi(z_{t,n-1}=M, z_{t,n}=j)}{\sum_{t=0}^{C-1}\sum_{n=2}^{N_t}\sum_{j=0}^{M}\xi(z_{t,n-1}=M, z_{t,n}=j)} \\ \theta_2 &= \frac{\sum_{t=0}^{C-1}\sum_{n=2}^{N_t}\sum_{k=0}^{M-1}\sum_{j=0}^{M-1}\xi(z_{t,n-1}=k, z_{t,n}=j)}{\sum_{t=0}^{C-1}\sum_{n=2}^{N_t}\sum_{k=0}^{M-1}\sum_{j=0}^{M}\xi(z_{t,n-1}=k, z_{t,n}=j)} \\ v &= \frac{\sum_{t=0}^{C-1}\sum_{n'\in\mathcal{S}_t}\sum_{j=0}^{M-1}\left[\sum_{k=0}^{M-1}\xi(z_{t,n'-1}=k, z_{t,n'}=j)(j-k)^2 + \xi(z_{t,n'-1}=M, z_{t,n'}=j)(j-x_{t,n'})^2\right]}{\sum_{t=0}^{C-1}\sum_{n'\in\mathcal{S}_t}\sum_{j=0}^{M-1}\sum_{k=0}^{M}\xi(z_{t,n'-1}=k, z_{t,n'}=j)}\end{aligned}$$

where $\mathcal{S}_t = \{n' : |x_{t,n'} - x_{t,n'-1}| \leq d\}, \forall t \in \{0, \ldots, C-1\}$.

After some algebraic manipulation, we end up in the following equations, which are very similar to the single condition case:

$$\epsilon_{t,j} = \frac{\zeta(z_{t,1} = j)}{1 - \zeta(z_{t,1} = M)} \quad , t \in \{0, \ldots, C-1\}, j \in \{0, \ldots, M-1\} \tag{2.44}$$

$$\theta = 1 - \frac{1}{C} \sum_{t=0}^{C-1} \zeta(z_{t,1} = M) \tag{2.45}$$

$$\theta_1 = 1 - \frac{\sum_{t=0}^{C-1} \sum_{n=2}^{N_t} \xi(z_{t,n-1} = M, z_{t,n} = M)}{\sum_{j=0}^{M} \sum_{t=0}^{C-1} \sum_{n=2}^{N_t} \xi(z_{t,n-1} = M, z_{t,n} = j)} \tag{2.46}$$

$$\theta_2 = 1 - \frac{\sum_{k=0}^{M-1} \sum_{t=0}^{C-1} \sum_{n=2}^{N_t} \xi(z_{t,n-1} = k, z_{t,n} = M)}{N - \sum_{j=0}^{M} \sum_{t=0}^{C-1} \sum_{n=2}^{N_t} \xi(z_{t,n-1} = M, z_{t,n} = j)} \tag{2.47}$$

$$\begin{aligned} v = & \frac{\sum_{k=0}^{M-1} \sum_{j=0}^{M-1} \sum_{t=0}^{C-1} \sum_{n' \in \mathcal{S}_t} \xi(z_{t,n'-1} = k, z_{t,n'} = j)(j-k)^2}{\sum_{t=0}^{C-1} |\mathcal{S}_t| - \sum_{k=0}^{M} \sum_{t=0}^{C-1} \sum_{n' \in \mathcal{S}_t} \xi(z_{t,n'-1} = k, z_{t,n'} = M)} \\ & + \frac{\sum_{j=0}^{M-1} \sum_{t=0}^{C-1} \sum_{n' \in \mathcal{S}_t} \xi(z_{t,n'-1} = M, z_{t,n'} = j)(j - x_{t,n'})^2}{\sum_{t=0}^{C-1} |\mathcal{S}_t| - \sum_{k=0}^{M} \sum_{t=0}^{C-1} \sum_{n' \in \mathcal{S}_t} \xi(z_{t,n'-1} = k, z_{t,n'} = M)} \end{aligned} \tag{2.48}$$

where $N = \sum_{t=0}^{C-1} N_t$.

Similarly with Equation 2.43, we have:

$$\pi_{t,j} = \frac{1}{N_t} \sum_{n=1}^{N_t} p(z_{t,n} = j | \mathbf{x}_t) = \frac{1}{N} \sum_{n=1}^{N} \zeta(z_{t,n} = j) \tag{2.49}$$

The above equations (either considering the single or multi– condition case) essentially form the M step of the EM algorithm.

In the E step, we will have to evaluate $\zeta(z_n = j), \xi(z_{n-1} = k, z_n = j)$ based on the $\boldsymbol{\pi}, \theta, \theta_1, \theta_2, v$ of the previous iteration. After obtaining these values, we will be able to evaluate the parameters $\boldsymbol{\pi}, \theta, \theta_1, \theta_2, v$, thus continuing in a recursive approach between E and M steps until some convergence criterion is satisfied.

2.3.1 Complexity Analysis

In the single condition case, the computational cost of the method is as follows:
From Equation 2.38, we easily see that it suffices to store $\zeta(z_1 = j), \forall j \in \{0, \ldots, M\}$ to determine $\boldsymbol{\pi}$, that is, we need to store $M + 1$ values. Similarly, from Equation 2.40, we need only to store the sums $\sum_{n=1}^{N} \xi(z_{n-1} = M, z_n = j), \forall j \in \{0, \ldots, M\}$, that is, $M + 1$ values. Also, from Equation 2.41, we have to do the same for the sums $\sum_{n=1}^{N} \xi(z_{n-1} = k, z_n = M), \forall k \in \{0, \ldots, M\}$, that is, another $M+1$ values. Of course, in order to be able to calculate the above sums, we need to know $\hat{\alpha}(z_n = j),\ c_n,\ \hat{\beta}(z_n = j) \quad , \forall n \in \{1, \ldots, N\},\ j \in \{0, \ldots, M\}$, which requires storage of $O((M + 1)N) + O(N) = O(NM)$.
In the latter one, we will also need the sums $\sum_{n' \in \mathcal{S}} \xi(z_{n'-1} = k, z_{n'} = M), \forall k \in \{0, \ldots, M\}$ which are a subset of the sums $\sum_{n=1}^{N} \xi(z_{n-1} = k, z_n = M)$. Lastly, we will need the sums $\sum_{n' \in \mathcal{S}} \xi(z_{n'-1} = k, z_{n'} = j)(j - k)^2, \forall k, j \in \{0, \ldots, M - 1\}$ and $\sum_{n' \in \mathcal{S}} \xi(z_{n'-1} = M, z_{n'} = j)(j - x_{n'})^2, \forall j \in \{0, \ldots, M - 1\}$ which require storage of M^2 and M values, respectively.
Summing up, instead of storing all the values of ζs and ξs, which would require a space of $N(M + 1) + N(M + 1)^2 = O(NM^2)$, we would only need $O(NM) + 4(M + 1) + M^2 + M = M^2 + 5M + 4 = O(NM) + O(M^2)$. Furthermore, we will save space for another $M + 4$ values for the storage of variables $\boldsymbol{\pi}, \theta, \theta_1, \theta_2, v$, thus leaving the complexity on the same order of magnitude, $O(NM) + O(M^2)$, that is.
If $N \ll M$, then by storing all the values of ζs and ξs we would have a space complexity of $N \cdot O(M^2)$, while by storing just the sums we will need only $O(M^2)$ space. So, depending on how big the constant factor N is, it will increase significantly the space requirements for the first case. The situation worsens when $N \approx M$, where in the first case the space storage needed will be $O(M^3)$ while in the second $2 \cdot O(M^2)$. Lastly, if $N \gg M$, the space complexity for the first case will be $M^2 \cdot O(N)$, while in the second $M \cdot O(N) + O(M^2)$, degenerating even to $M \cdot O(N)$ (if $N \gg M^2$). Again, we see that substantial amount of space is saved with the second approach (of storing just the sums).

Lastly, the computational analysis for the multi-condition case is the following:
Storing all the values of ζs and ξs would take $CN(M + 1) + CN(M + 1)^2 = O(CNM^2)$

space. However, after observing the Equations 2.44, 2.45, 2.46, 2.47, 2.48, we can reduce substantially the space needed. In more detail, from Equation 2.44, we see that $M+1$ values are needed for the evaluation of $\boldsymbol{\pi}_t$. Thus, we would need $C(M+1)$ values to store. Also, from Equations 2.46, 2.47 and 2.48, we see that the storage of $M+1$, $M+1$ and $M^2+M+M+1$ is required, respectively. Similarly, to the single-condition case, $O(CNM)$ will be needed for the storage of $\hat{\alpha}(z_{t,n}=j),\ c_{t,n},\ \hat{\beta}(z_{t,n}=j) \quad ,\forall t \in \{0,\ldots,C-1\}, \forall n \in \{1,\ldots,N\},\ j \in \{0,\ldots,M\}$. Thus, totally we need $O(CNM)+M^2+(C+4)M+C+3 \overset{M \gg C}{=} O(CNM)+O(M^2)$, which is substantially less than the originally required space of $O(CNM^2)$. As in the single condition case, we would additionally need the storage of $CM+4$ values for the $\boldsymbol{\pi}_t s$, $\theta, \theta_1, \theta_2$ and v, something that does not change the space complexity of the multi-condition case.

2.4 Temporal Coupling

In the temporal coupling case, we are interested in aligning events of different time conditions together. The biological motivation for that is that events among different time conditions that are in very proximal regions (between each other) should be aligned since there is no strong reason for believing otherwise, e.g. that the binding motif of the TF has been changed over time. Furthermore, we will test the hypothesis that by this method better spatial resolution, sensitivity and specificity are achieved.

Therefore, we will first apply a simple independent mixture model (*global* mixture model) in all conditions with a sparse prior (see Subsection 2.1.3 for a brief discussion on sparse priors). This will provide all the candidate events that each condition can have. In a second step, we will apply a simple independent mixture model on each condition separately without a sparse prior (*condition–specific* mixture model) only on the candidate events that previously the global mixture model allowed. So, by applying the global mixture model on all conditions, we, first, enforce sparseness, a property that is desirable for this problem, and, second, we create a set of candidate events that is accepted from the consensus of all conditions. Afterwards, by running each condition on a condition–specific mixture model without a prior (on prior weights) only on the components allowed by the previous step, we guarantee that the discovered events would be condition–specific yet as constrained as possible to be in a very close if not the same location with the same event on the other conditions. The plate notation of a simple mixture model is shown in Figure 2-6. Remember from Section 2.2 that with $\boldsymbol{\alpha}$ we represent the parameters of the sparse prior, with $\boldsymbol{\pi}$ the prior weight of the components, with z_n the assignment of a read to a component and with x_n the position of a read.

Obviously, the independence assumption between observed and hidden variables is quite simplistic but as it will become clear later in the Results Section, it is well applicable for this problem.

Besides, as it was mentioned, we let each condition determine its events based only on the condition–specific data (without the use of a sparse prior). In that way, for each condition

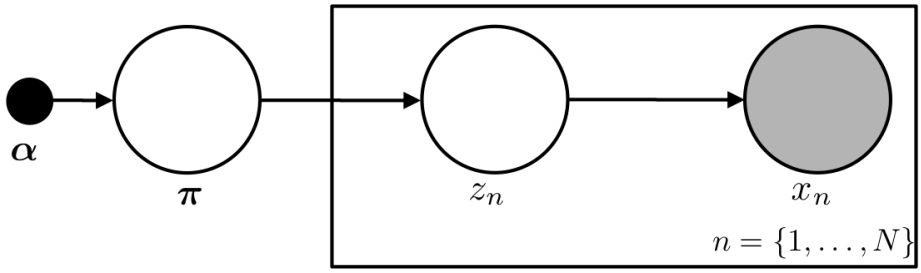


Figure 2-6: **Simple Mixture Model.**
This figure shows a simple mixture model where independence between different observed ($x_n s$) and hidden ($z_n s$) variables is assumed. There also could be a prior ($\boldsymbol{\alpha}$) on the prior weighting ($\boldsymbol{\pi}$).

we find the events that best fit the data of this condition but are still permitted from the consensus of all the conditions through the use of the global mixture model at the previous step. The aforementioned are pictorially summarized in Figure 2-7):

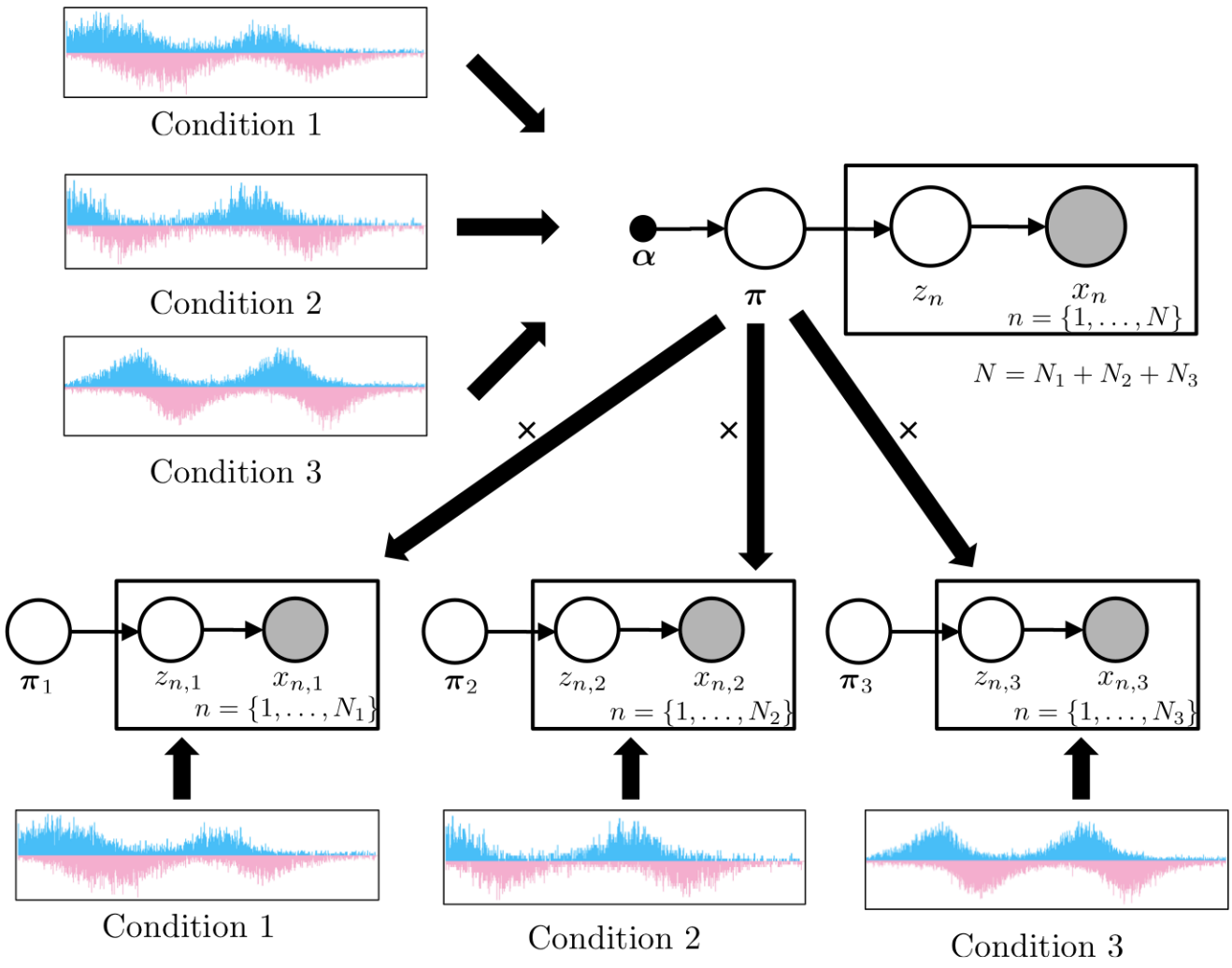


Figure 2-7: **Temporal coupling between conditions.**
In the first step of this method, "candidate" events are first discovered based on all conditions by a simple mixture model with a sparse prior (global mixture model). In the second step, these events are used as a "mask" for the discovery of events on each condition separately. That is to say, the events of each condition are discovered based only on the data of this condition after searching from the set of "candidate" events that the global mixture model generated.

2.4.1 Temporal Coupling Algorithm

At this point, we will cite the algorithm of this method.

Temporal Coupling Algorithm

while (EM not converged yet) {

For iteration i:

1. Evaluate $\boldsymbol{\pi}^{(i)}$ on all conditions using a Simple Mixture Model with a negative Dirichlet-type sparse prior
2. Evaluate $\boldsymbol{\pi}_t^{(i)}$ on each condition using a Simple Mixture Model without a prior as:
   (a) $\boldsymbol{\pi}_t^{(inter)} \propto \boldsymbol{\pi}_t^{(i-1)} .* \boldsymbol{\pi}^{(i)}$ [a]
   (b) $\boldsymbol{\gamma}(z_{t,n})^{(i)} \propto \boldsymbol{\pi}_t^{(inter)}$
   (c) $\boldsymbol{\pi}_t^{(i)} \propto \boldsymbol{\gamma}(z_{t,n})^{(i)}$
3. Check convergence based on $\boldsymbol{\pi}$

}

[a] $\mathbf{A} .* \mathbf{B}$ indicates component–wise multiplication. That is, every element of matrix (vector) $\mathbf{A}$ is multiplied with the corresponding element of matrix (vector) $\mathbf{B}$.

It is worth reminding that with $\boldsymbol{\pi}^{(i)}$ and $\boldsymbol{\pi}_t^{(i)}$, we denote the probabilities of the components for all the data and the data at t^{th} condition in i^{th} iteration of the algorithm, respectively. Also, $\boldsymbol{\gamma}(z_{t,n})^{(i)}$ denotes the posterior probability of a read being assigned to a component given that it is placed in position $x_{t,n}$ ($p(z_{t,n} = j|x_{t,n})$) in i^{th} iteration of the algorithm.

In the place of $\boldsymbol{\pi}^{(i)}$ in (2.a), we could have $\mathbf{1}(\boldsymbol{\pi}^{(i)} \neq 0)$, where we would have ones only on the places of $\boldsymbol{\pi}^{(i)}s$ with non zero components. Thus, the global prior could also be used as a "mask" allowing only "candidate" events supported by the consensus of all conditions.

Besides, we expect the algorithm to converge since convergence is sought on all conditions, however, the log-likelihood of the data in each condition is not guaranteed to monotonically increase because of the intervention of global $\boldsymbol{\pi}$*'s* in the evaluation of $\boldsymbol{\gamma}(z_{t,n})$*'s* during the E-step in each condition.

Chapter 3

Results

In this chapter, we are going to show results on the previously described methods. First, we are going to talk about the read distribution ($p(x_n|z_n)$) we used in our experiments in Section 3.1. Later on, we will present results based on the Spatial Coupling method (Section 3.2) and afterwards results based on the Temporal Coupling one (Section 3.3).

In more detail, in Section 3.2, we will show that Spatial Coupling clearly underperforms the Simple Mixture, in general, both in terms of failing to achieve sparseness and having much greater running time. The events, however, having the highest probabilities seem to be satisfactorily close to the true ones and even outperform the ones predicted by the Simple Mixture as shown in Figure 3-7.

For the case of Temporal Coupling, we both tested it on synthetic and real data. On synthetic data, we created a region where true events were aligned (Z:0–800) and a region where true events were not aligned (Z:0–900). In almost all cases, except for the alignment feature that the algorithm enforces, it manages to restrict the number of False Positive (FP) events as shown in Figures 3-14c, 3-16c, 3-19, 3-22. In addition, Temporal Coupling in the case of aligned true events (Z:0–800), showed more robustness in calling events on the same locations across repetitions. See Figure 3-14b. The distance also between predicted and true events is lower than the Simple Mixture (Figure 3-17), although the Simple Mixture performs better when events are not aligned (Z:0–900). See Figure 3-20.

3.1 Read Distribution

In this section, we will briefly describe how we obtain the read distribution, which essentially represents the emission probability $p(x_n|z_n = j)$ for each read, indicated by the symbol $H_{x_n,j}$.

In short, we first find the peaks of the regions (we are interested in) with one of the tools we previously described. Then, we keep the most enriched ones and after having determined the centers of each one of these, we sum the read counts (for each strand) together to obtain a general peak profile for both strands as depicted in the midpart of Figure 3-1. Lastly, we mirror the reverse read distribution w.r.t. the start of the origin to obtain the general read distribution H, which we are going to use in our model. The above are summarized in Figure 3-1.

3.2 Spatial Coupling Results

In this section, we are going to compare the Spatial Coupling method with the simpler one of a Simple Mixture model with a sparse prior (as presented in Figure 2-7). As we will see, the Simple Mixture model will prove superior to the Spatial Coupling one in terms of running time and event discovery power. In addition, the sparseness characteristics are much more apparent in the Simple Mixture model than the Spatial Coupling one.

Besides, it is worth mentioning that in this set of experiments, we use three different flavors of the Spatial Coupling method. The first one was described in Figure 2.3, the second one still retains the error model introduced in the original method, but it considers the transition probabilities as discrete random variables learned by the model. The Gaussian dependence of a component based on its surrounding ones is introduced during initialization. Lastly, in the third one, we additionally ignore the error model, degenerating it to a simple HMM with a sparse prior.

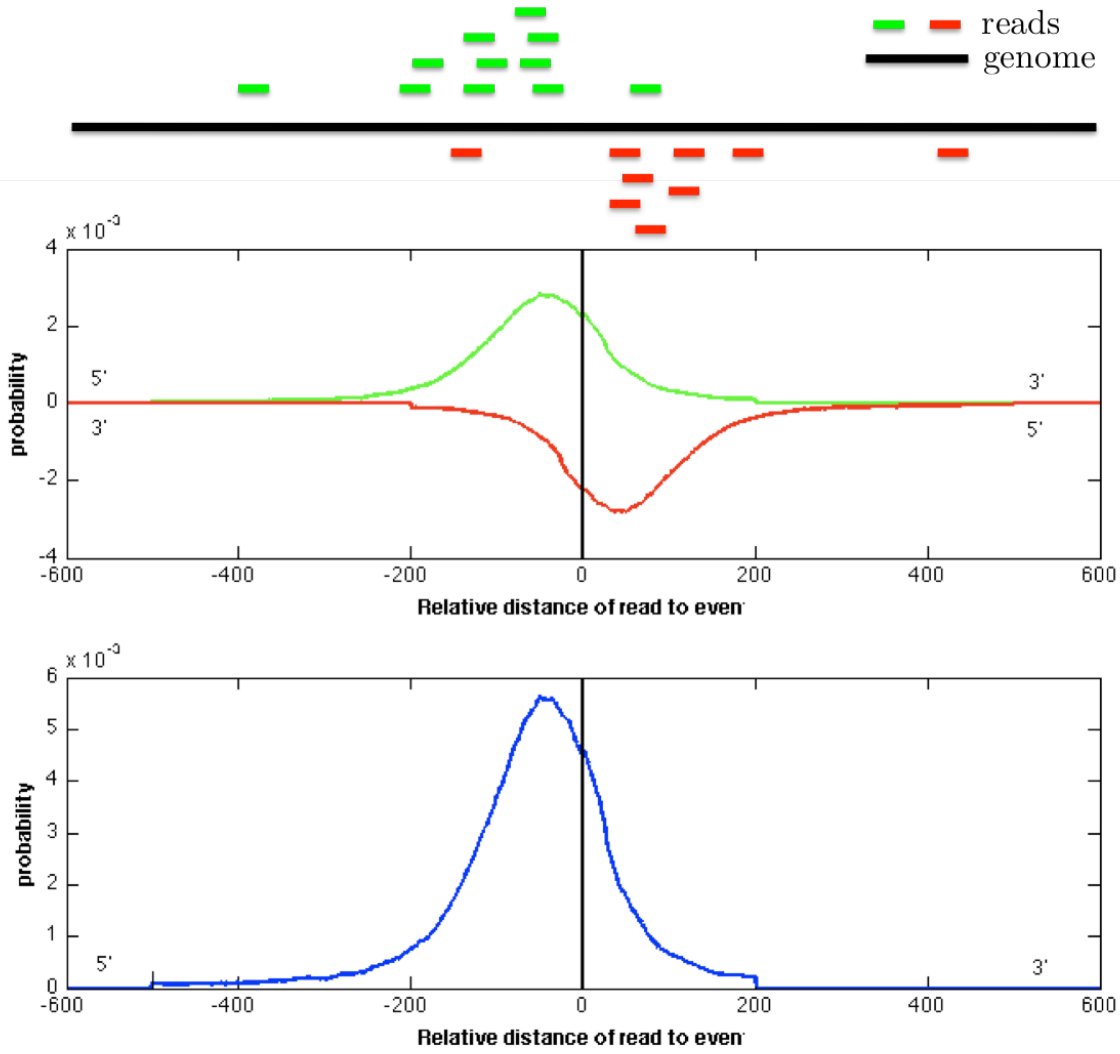


Figure 3-1: **Read Distribution.**
This figure illustrates how the distribution of a read is generated. The reverse read distribution (curve in red in the mid part of the figure below the x-axis) is mirrored w.r.t. y-axis and flipped w.r.t. x-axis and then the two are summed up to give the final read distribution (curve in blue in the bottom part of the figure).

3.2.1 Toy Example

Before presenting the results, we will briefly explain the process of generating the toy data. We chose to create a region of length of 60 bp and (true) events placed at positions 1, 20, 21 and 50. We assigned some strength to each event constraining all strengths to sum to one. Afterwards, we sampled a number u from a uniform distribution in the range $[0, 1]$, and assigned a hidden variable to the position of the event with cumulative probability that was the least greater to this uniform number u. Lastly, we assigned a value for an observed variable based on the value of its corresponding hidden one and using a Binomial distribution as our emission probability with a standard deviation of 5. With this process, we created $10,000$ reads.

During the learning phase, we trained these data both on the Simple Mixture and the simplest version of the Spatial Coupling method. For both methods, the value for the sparse prior was set to 10 ($\alpha = 10$. See Subsection 2.1.3). The results are depicted in Figure 3-2. As we see, the Simple Mixture model (upper subfigure) discovers all the true events plus an additional one (False Positive) in position 5, while the Spatial Coupling one (bottom subfigure) fails to identify the events at positions 20, 21 and calls many more around event 50, however, with much lower weight, due to the chaining nature of an HMM. That is, if a (False Positive) event is initially called in a position, then because there is a dependency between the assignments of neighboring reads (favoring the calling of potential event positions) which are close to an already called event, this cascading is spread throughout the subsequent positions as well, assigning a very low yet non zero probability to these positions.

In addition, the "sparsity" feature does not work as effectively in the Spatial Coupling case as the Simple Mixture one. This is mainly because the sparse prior on the Simple Mixture model performs on all variables, while the sparse prior on an HMM performs only on the first variable, and, subsequently, expects to cascade the "sparseness" to the rest of the variables as well through their dependencies expressed by the transition probabilities.

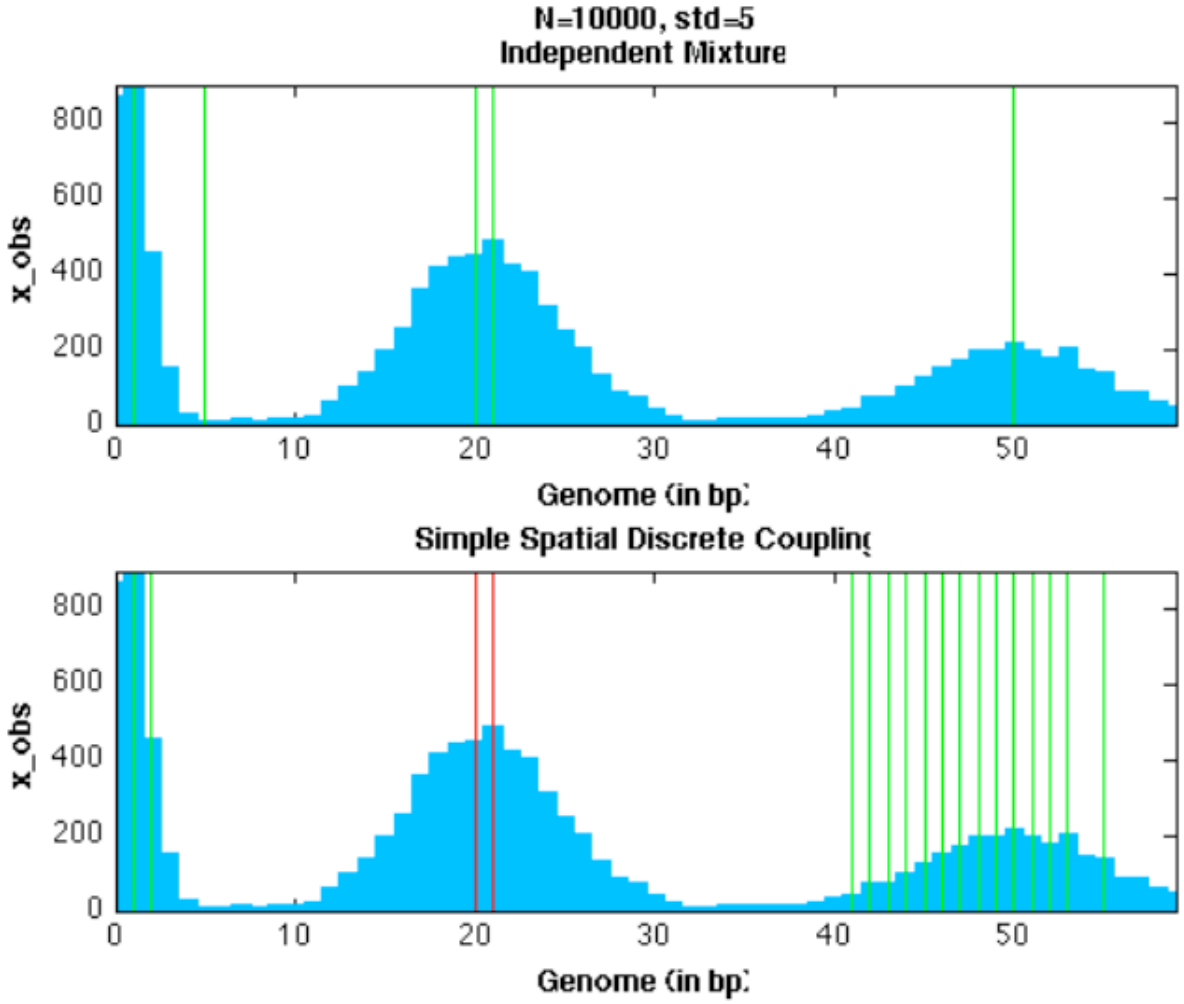


Figure 3-2: **Toy example for the Spatial Coupling method.**
This figure shows the comparison performance between a Simple Mixture model and the Spatial Coupling one in a single condition. The Simple Mixture model (upper subfigure) managed to uncover all the true events plus an additional (false positive) one. On the contrary, the simple Spatial Coupling method (bottom subfigure) fails to detect two of the events while it discovers the other two and a whole lot of other weak ones.
Red lines represent the positions of the true events; that is, 1, 20, 21, 50. Green lines indicate predicted events.

3.2.2 Real Data

We used Oct4 Chip-Seq data coming from murine ES cells which were grown under typical ES cell culture conditions. Treated cells and non-treated controls were stained with primary antibodies against Oct4 and Sox2 and then were chromatin immunoprecipitated, and after sonication, the fragments were processed by the Illumina/Solexa sequencer [1]. We ran the Simple Mixture model and the Spatial Coupling one on several test regions. For both methods, the value for the sparse prior was set to 10 ($\alpha = 10$. See Subsection 2.1.3). Since, as we mentioned, the Spatial Coupling method cannot enforce sparseness in the degree that the Simple Mixture does, we chose the most significant peaks (those with the highest probability) that the two algorithms predicted. Before discussing the results further, it is worth mentioning the color coding we will adopt. With blue color, we will depict the counts of forward ('+') reads, while with red, the counts of reverse ('-') strands. The first lane will represent the IP (Oct4) channel with the second lane the control channel. The third lane represents Oct4's binding motifs. The black color represents a significant motif. In addition, the last two lanes, represent the peaks discovered by the Simple Mixture and the Spatial Coupling model, respectively. Lastly, we will represent a region by `chr_id:region_start-region_end`. E.g., 3:235,000-237,000 indicates that the region is at chromosome 3 and starts and ends in 235,000 and 237,000, respectively. Coordinates are in base pairs (bp).

Following, we present the results on some of these regions. First, we investigated the region 3:34,837,000–34,839,000. As we see, the two methods perform very similarly. In addition, the motif being at position 34,837,955 is closer to the peak predicted by the Spatial Coupling than the one predicted by the Simple Mixture (see Figure 3-3).

In the second example (concerning the region 13:57,371,000–57,373,000), the peaks predicted by both methods are still very close to each other, but this time the peak discovered by the Simple Mixture model falls into the motif location (position: 57,372,394), while the one found by the Spatial Coupling is 10 bp apart (see Figure 3-4).

In the third case, being region 17:53,026,000–53,028,000, there are two Oct4 motifs within

a distance of 36 bp where the predicted peaks by both methods are located. The one predicted by the Simple Mixture model falls into the location of the one of the motifs, while the one found by the Spatial Coupling methods falls in the middle between the two motifs (see Figure 3-5).

Lastly, we examined the region 18:31,711,000–31,713,000. Here, both peaks are located almost on the same position (with 3 bp difference) and are very close to an Oct4 motif (being at position 31,711,952). The peak found by the Simple Mixture model is located at 31,711,959, while the one predicted by the Spatial Coupling one is located at position 31,711,962 (see Figure 3-6).

For a thorough comparison, we tested both methods' performance on 1,503 regions. Out of all these regions, 726 were within a window of 25 bp to a motif using a threshold of 7.99 which corresponds to 0.1% FDR ratio of the motif "Oct4 TRANSFAC 10.4, M01125". As noted previously, we kept only the highest scoring peaks, those with the maximum assigned prior weight since the Spatial Coupling method cannot enforce sparseness in a satisfactory degree as the Simple Mixture method does.

Spatial Coupling seems to do slightly better than the Simple Mixture one but considering the much greater running time it requires and the poor sparseness enforcement, we can say that Simple Mixture does better in general.

The running time of each region was on the order of seconds ($\sim$2.5–6.5 sec) for the Simple Mixture model, while on the order of minutes ($\sim$8–14 min) for the Spatial Coupling one. The experiments were run on a Mac OS X 10.5 (2.4 GHz Intel Core 2 Duo, 4 GB memory).

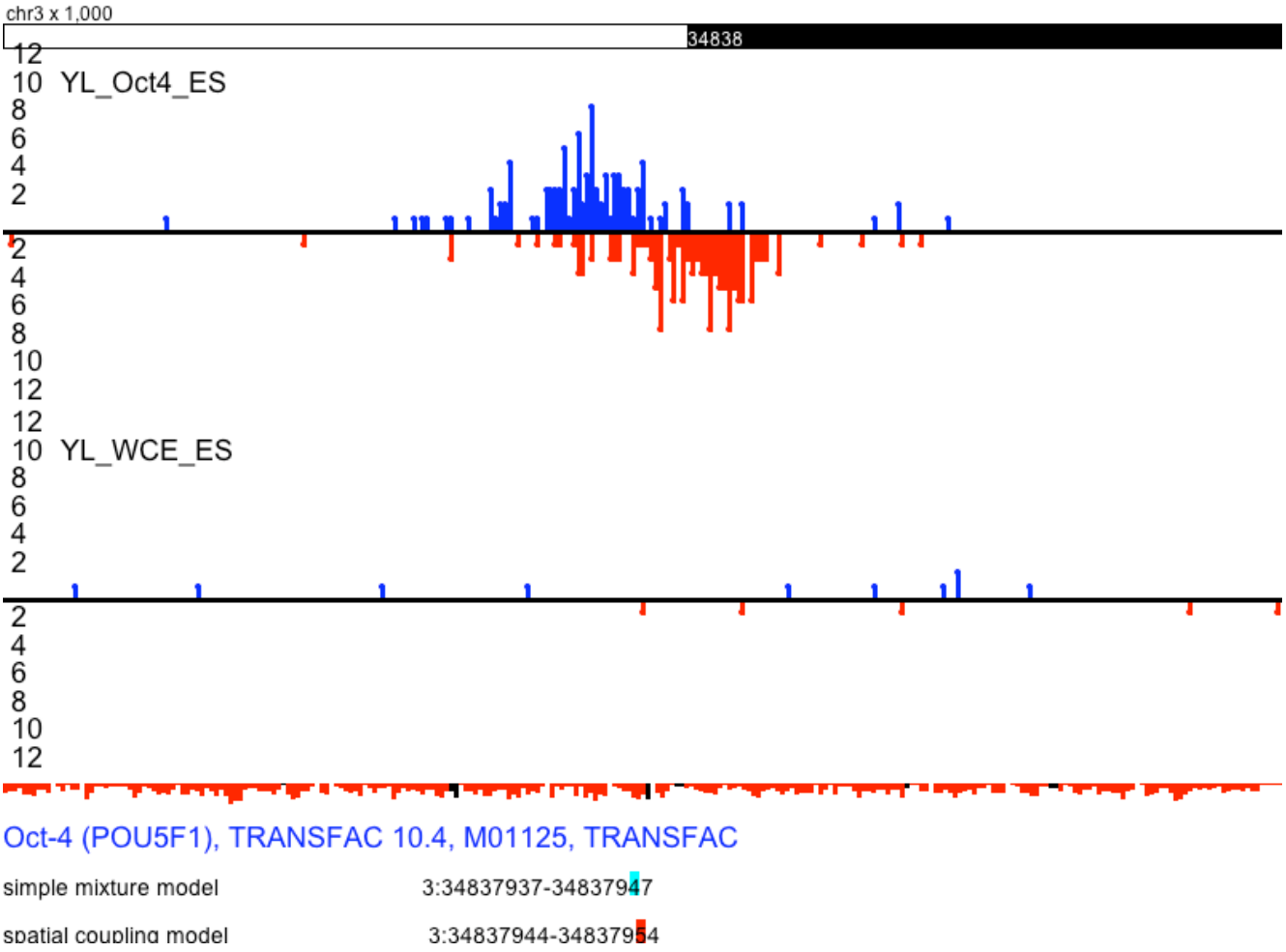


Figure 3-3: **Comparison between Simple Mixture Model and Spatial Coupling. Region 3:34,837,000–34,839,000.**
The two methods predict peaks almost at the same position and very near an Oct4 motif. Yet, the peak outputted by the Spatial Coupling method is a bit closer to the motif than that of the Simple Mixture Model.
Blue lines represent the counts of forward ('+') reads, while red lines the counts of reverse ('−') reads. Black bars indicate Oct4 motifs. The last two lanes show the peaks that each method predicts.

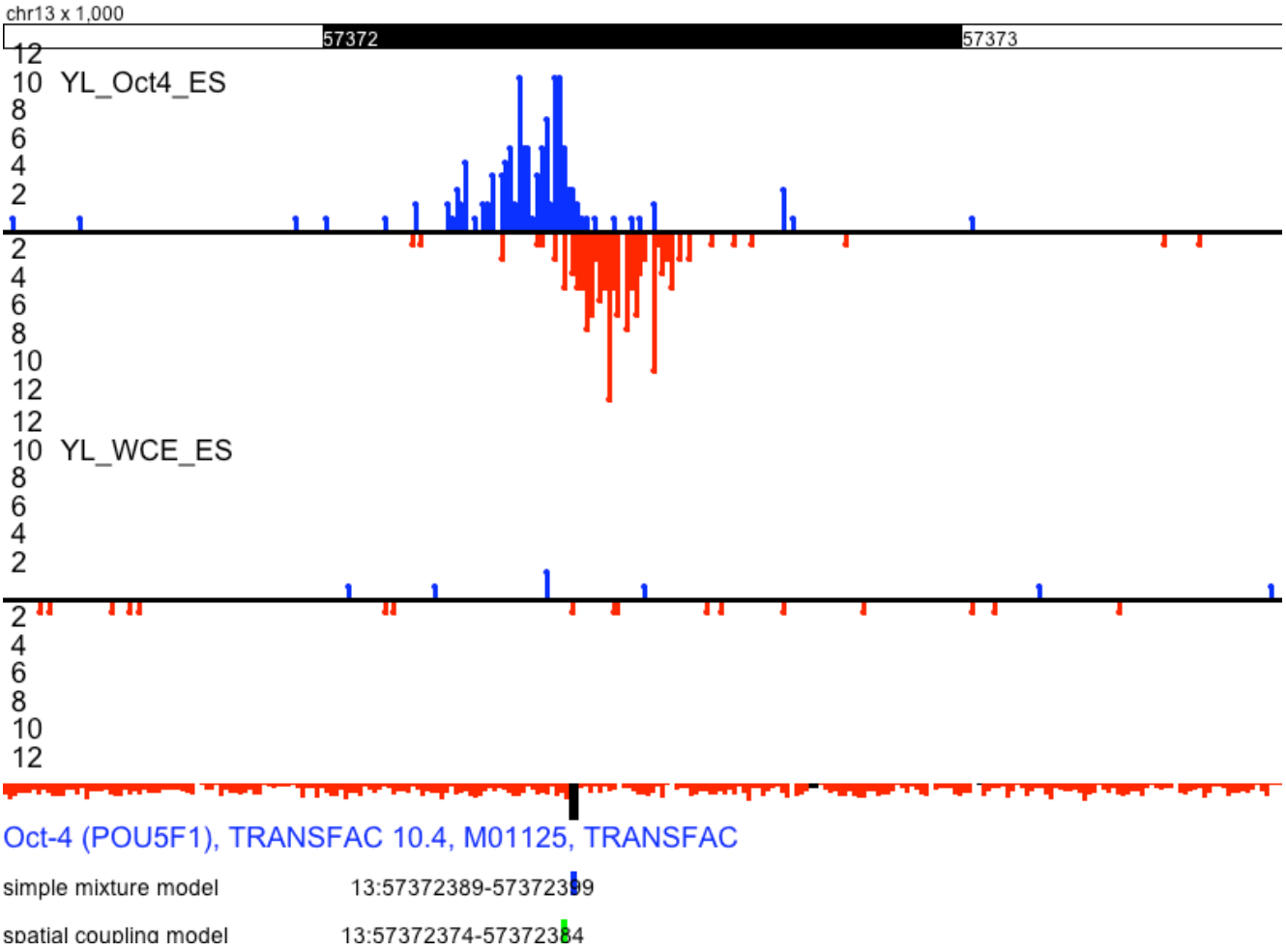


Figure 3-4: **Comparison between Simple Mixture Model and Spatial Coupling. Region 13:57,371,000–57,373,000.**
The two methods predict peaks near the motif, although the one found by the Simple Mixture model performs a bit better by falling exactly inside the motif location.
Blue lines represent the counts of forward ('+') reads, while red lines the counts of reverse ('−') reads. Black bars indicate Oct4 motifs. The last two lanes show the peaks that each method predicts.

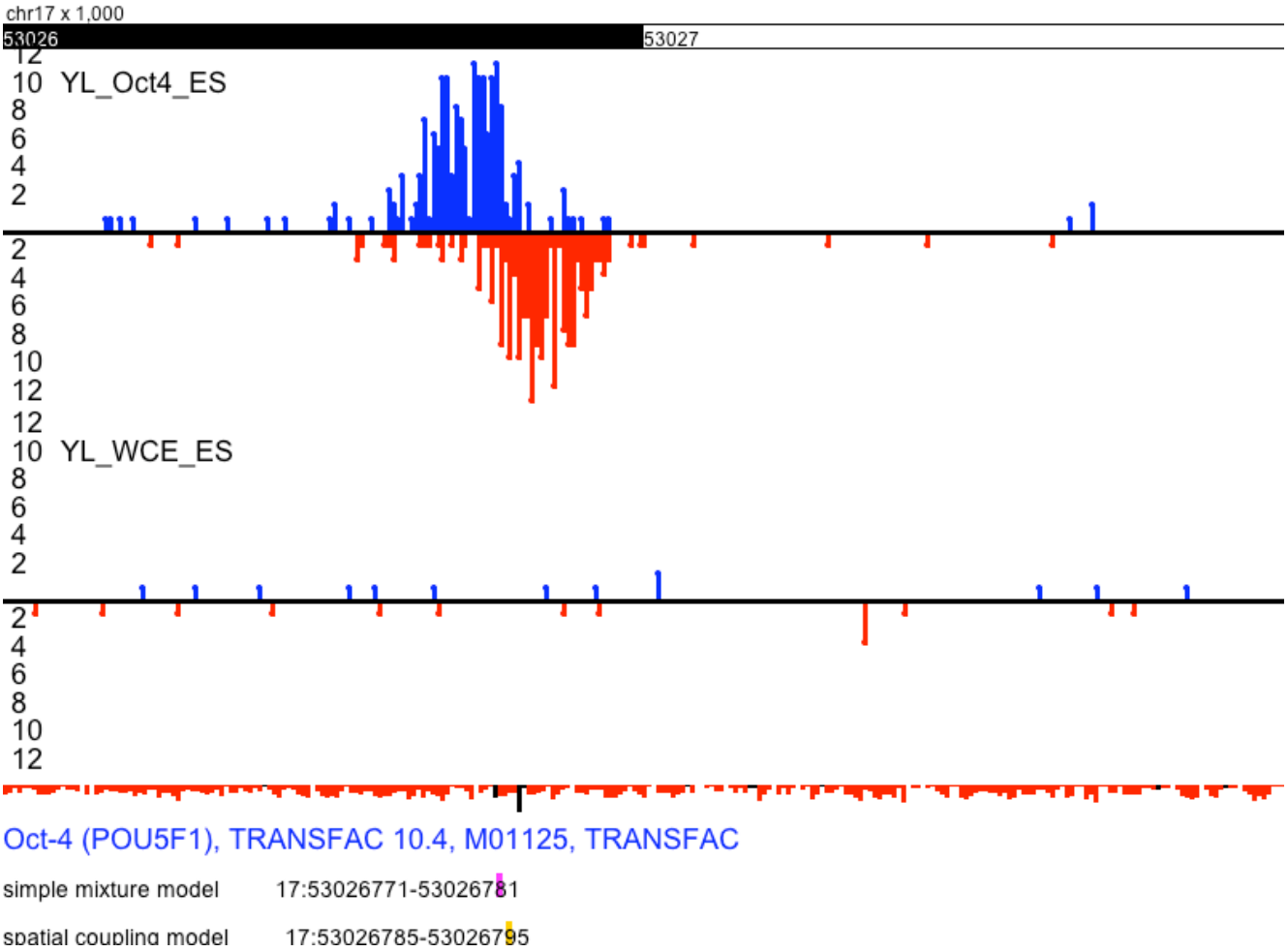


Figure 3-5: **Comparison between Simple Mixture Model and Spatial Coupling. Region 17:53,026,000–53,028,000.**
The peak predicted by the Simple Mixture model falls into the location of one of the motifs, while the peak detected by the Spatial Coupling is located in the middle of the two motifs. Blue lines represent the counts of forward ('+') reads, while red lines the counts of reverse ('−') reads. Black bars indicate Oct4 motifs. The last two lanes show the peaks that each method predicts.

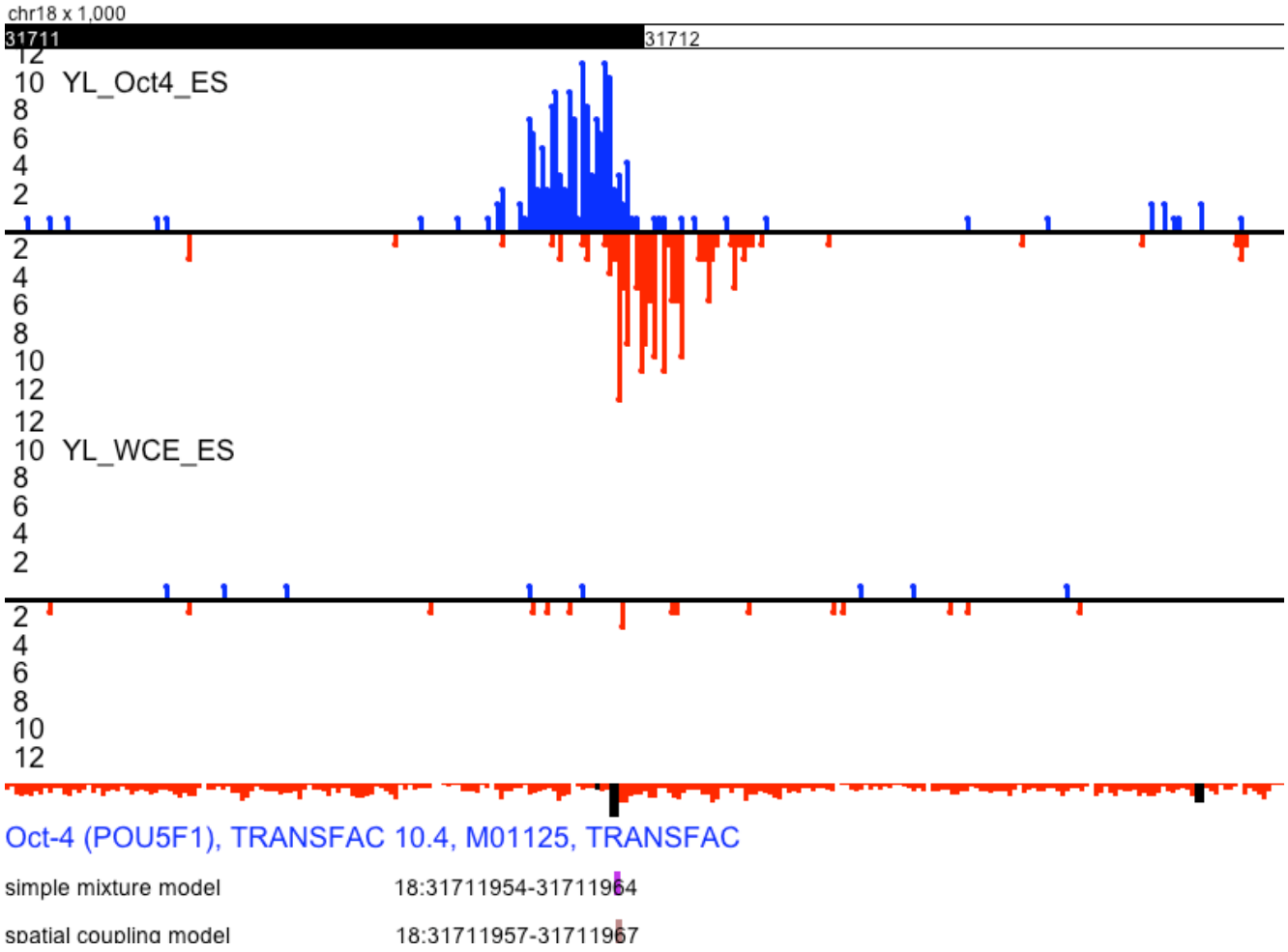


Figure 3-6: **Comparison between Simple Mixture Model and Spatial Coupling. Region 18:31,711,000–31,713,000.**
Here, both peaks are within 10 bp distance from the motif located at position 31,711,952. The peak predicted by the Simple Mixture model performs slightly better by being 3 bp closer to the motif.
Blue lines represent the counts of forward ('+') reads, while red lines the counts of reverse ('−') reads. Black bars indicate Oct4 motifs. The last two lanes show the peaks that each method predicts.

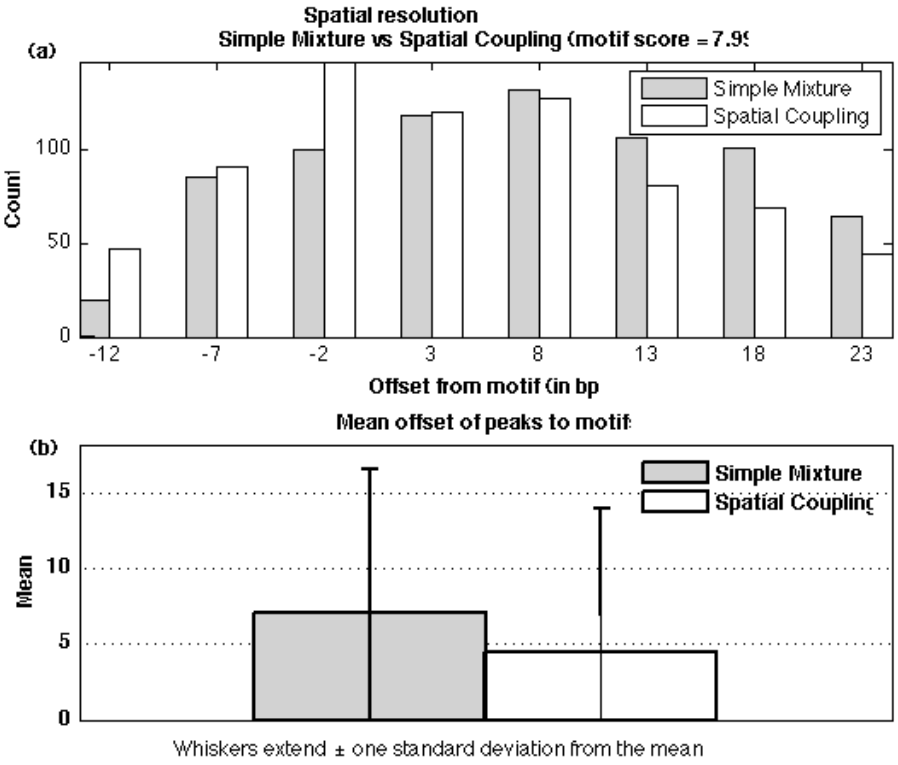


Figure 3-7: **Spatial Coupling versus Simple Mixture histogram.**
This figure shows the histogram of the distances of predicted peaks to motif locations. Both Spatial Coupling and Simple Mixture were run on 1,503 regions. Out of these, 726 were within a window of 25 bp to a motif of value 7.99 (corresponding to 0.1% FDR ratio of the motif "Oct4 TRANSFAC 10.4, M01125"). Only the highest predicted peak for each region (those corresponding to maximum prior weight) was taken into consideration. It is shown that the Spatial Coupling performs slightly better than the Simple Mixture one with a mean distance of 4.5 bp versus 7 bp that the Simple Mixture achieves. However, the poor sparseness that it enforces and the prohibitive running time makes the use of this method inexpedient.

3.3 Temporal Coupling Results

We are going to test the Temporal Coupling method across multiple conditions and compare it against running a Simple Mixture model in each condition both on synthetic and real data. Our goal is to investigate how well aligned the events are across conditions, the sparseness that each method achieves and test the hypothesis that the Temporal Coupling does better compared to the Simple Mixture in terms of spatial resolution, sensitivity and specificity.

3.3.1 Synthetic Data

We created synthetic datasets simulating faithfully the read generating process. In more detail, we determined in advance the true peaks, and then generated reads according to the read distribution discussed in Section 3.1. The span of the read distribution we used was from –300 to +299 bp (relative to an event). Also, in all of our tests, we used a noise probability of 0.1. That is, we assumed that almost 10% of the generated reads were drawn from an error model and distributed uniformly on the test region. The rest of the reads were assumed to be generated from one of the events. Lastly, both for the case of the Simple Mixture and that of the Temporal Coupling, we used the same value for the sparse prior, that is α was set to 10 (see Subsection 2.1.3).

For the purpose of testing the Temporal Coupling algorithm's performance against that of the Simple Mixture we created two regions; one where all true events were aligned across conditions (Z:0–800) and another (Z:0–900) where true events were not aligned across conditions or were not present in all conditions. We created the first region (Z:0–800) to evaluate how well the algorithm performs in a real world scenario, that is, when true events are located on the same position across conditions, while the second one (Z:0–900) to test how robust the algorithm is in an unreal scenario where true events are not aligned or even worse are not present in all conditions. In other words, we would like to test if the algorithm persists in calling False Positive events in positions where true events are not located due to its main functionality of aligning predicted events.

Before discussing the first case, we will briefly explain the color coding we used. With

light blue color, we will depict the counts of forward ('+') reads, while with pink, the counts of reverse ('−') strands. The red bars will represent the positions of the true events, while the differently colored bars the predicted events in each condition. The height of a bar indicates the strength of the event. Only for the case of Temporal Coupling, black bars indicate the events discovered from the aggregation of data in all conditions and represent the candidate events that each condition can have.

In the first scenario, we defined our events to be at positions 100, 200 and 500, thus creating a region of length 801 bp, calling it Z:0–800. We created three datasets representing three different conditions with unequal number of reads (10,000, 5,000 and 20,000 for the first, second and third condition, respectively). Events at positions 100 and 200 were put close enough on purpose so that the generated data in that location would be a convolution of reads obtained by those two events. The third event at position 500 was distant enough from the other two, so that the read landscape in that location would be only the outcome of a large part of reads generated by that event and a smaller one generated from the background (noise) model.
We first run this region on the Simple Mixture model on each condition separately. As it appears on Figure 3-8, the fitting of this method creates many more events than the true ones; 9, 7 and 9 events in conditions 1, 2 and 3, respectively. For the first condition, the ones that are most significant and are also close to the true events are the ones at positions 119, 214 and 499. For the second, the most significant ones are at positions 129, 192 and 500, while for the third one, they are at positions 122, 203 and 505. Also, within a condition around a significant event there are also a few others (much weaker) that are called. Thus, not only the events are not aligned across conditions but within a condition there are many weak false events (False Positives).

The Temporal Coupling method discovers only three events on the same positions for all conditions. In more detail, events are discovered at positions 118, 207 and 507. The events at positions 207 and 507 are very close to the true ones (at positions 200 and 500). However, the event at position 118 is 18 bp away from the true event (at position 100). This may

be due to the fact that reads having the same distance upstream and downstream of two events receive greater probabilities for the event downstream of them rather than the event upstream of them. So, the predicted location is biased towards the downstream event of the reads between 100 and 200 bp which in this case is the event at position 200.

To confirm our allegations on Subsection 2.4.1 about the fact that log-likelihoods of the conditions are not guaranteed to increase, we present the log-likelihoods for the aggregated data and the data for each condition in Figure 3-10. Initially, the condition–specific log-likelihoods increase, but after a while they start to decrease since the algorithm gives optimal solution at each step based on the aggregated data where sparseness is imposed via the negative Dirichlet-type prior. So, the set of candidate events is not optimal for each condition, separately. Nonetheless, after some "burn-in" period, since sparseness has been extensively spread, the subset of candidate events is small, and, so, the algorithm for each condition chooses a setting that does not differ much from the one at the previous step, thus resulting in the increase of the condition–specific log-likelihood as well.

In the second case (Z:0–900), we deliberately chose the true events not to be aligned. In other words, true events were located in slightly different positions across conditions. In addition, not all events were present in all conditions. That way, we could test for the algorithm's robustness. That is, if in this imaginary yet worst-scenario case, the Temporal Coupling algorithm lacked its "event-alignment" capability and if it (falsely) detected events in places where it should not.

In more detail, we considered three events for the first condition: {100, 200, 500}. We set the number of events for the second condition to two. The events were at positions {70, 450} close to the ones in positions 100 and 500 of the first condition. Thus, the event at position 200 was missing. In the third condition, we defined again two events, in positions {230, 600}, thus lacking the event near position 100 present in the other two conditions.

We first ran this dataset on a Simple Mixture model for each condition separately. Here, as in the previous case, many more than the true events were discovered. In detail, 6, 10 and 13 events were discovered for the first, second and third condition, respectively. The most

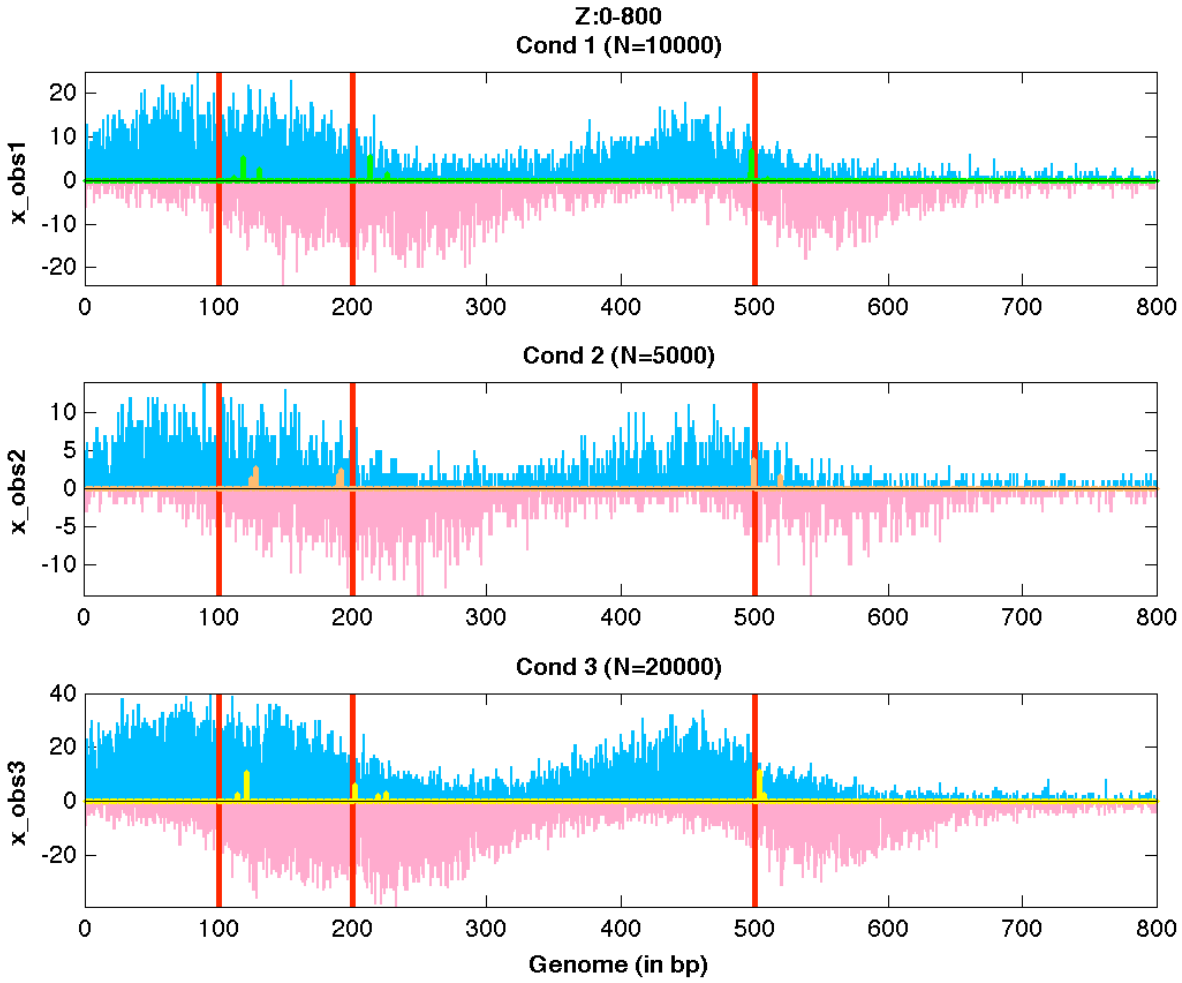


Figure 3-8: **Simple Mixture model on the test region Z:0–800.**
By performing a Simple Mixture model on each condition separately, many more events than the true ones are discovered. In addition, the most significant ones which are at positions {119, 214, 499}, {129, 192, 500}, {122, 203, 505} for the first, second and third condition, respectively are not aligned, whereas it was expected otherwise.
The true events are located at positions {100, 200, 500} in all three conditions. The counts of forward ('+') reads are depicted with light blue color, while the counts of reverse ('−') strands with pink. The red bars represent the positions of the true events, while differently colored bars predicted peaks. The height of a bar indicates the strength of the event.

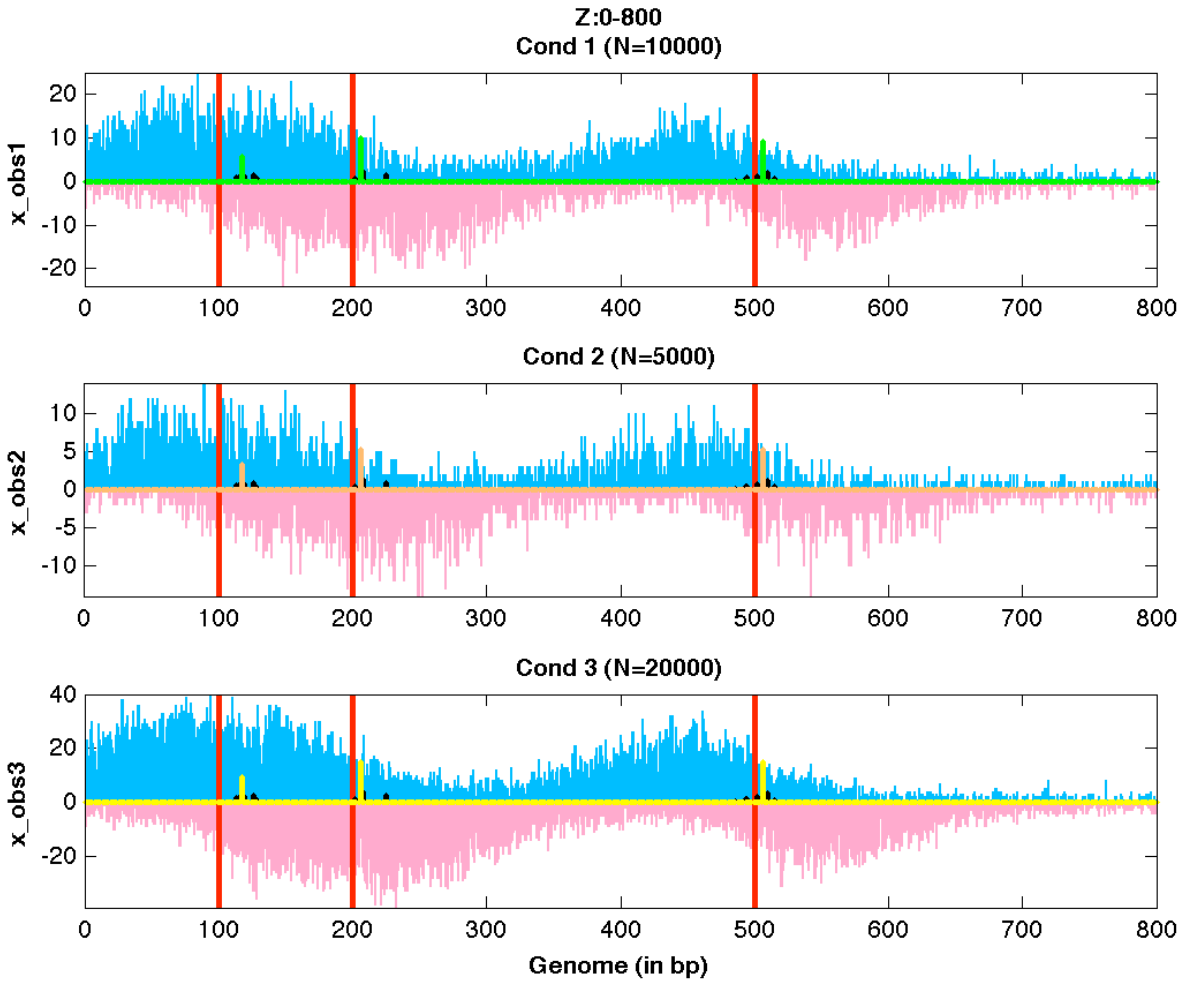


Figure 3-9: **Temporal Coupling model on the test region Z:0–800.**
This method discovers only three events, the same number with that of the true ones. Furthermore, they are aligned and located at positions {118, 207, 507}. The last two predicted peaks are within an accepted range apart from the corresponding true events (at positions 200 and 500), considering the different number of reads for each condition and the noise that is incorporated in the data. The first peak is slightly biased towards the second event due to the larger impact that a downstream (compared to an upstream) event has to a read.
The true events are located at positions {100, 200, 500} in all three conditions. The counts of forward ('+') reads are depicted with light blue color, while the counts of reverse ('−') strands with pink. The black bars indicate the events discovered from the aggregation of data in all conditions and represent the candidate events that each condition can have. The red bars represent the positions of the true events, while differently colored bars predicted peaks. The height of a bar indicates the strength of the event.

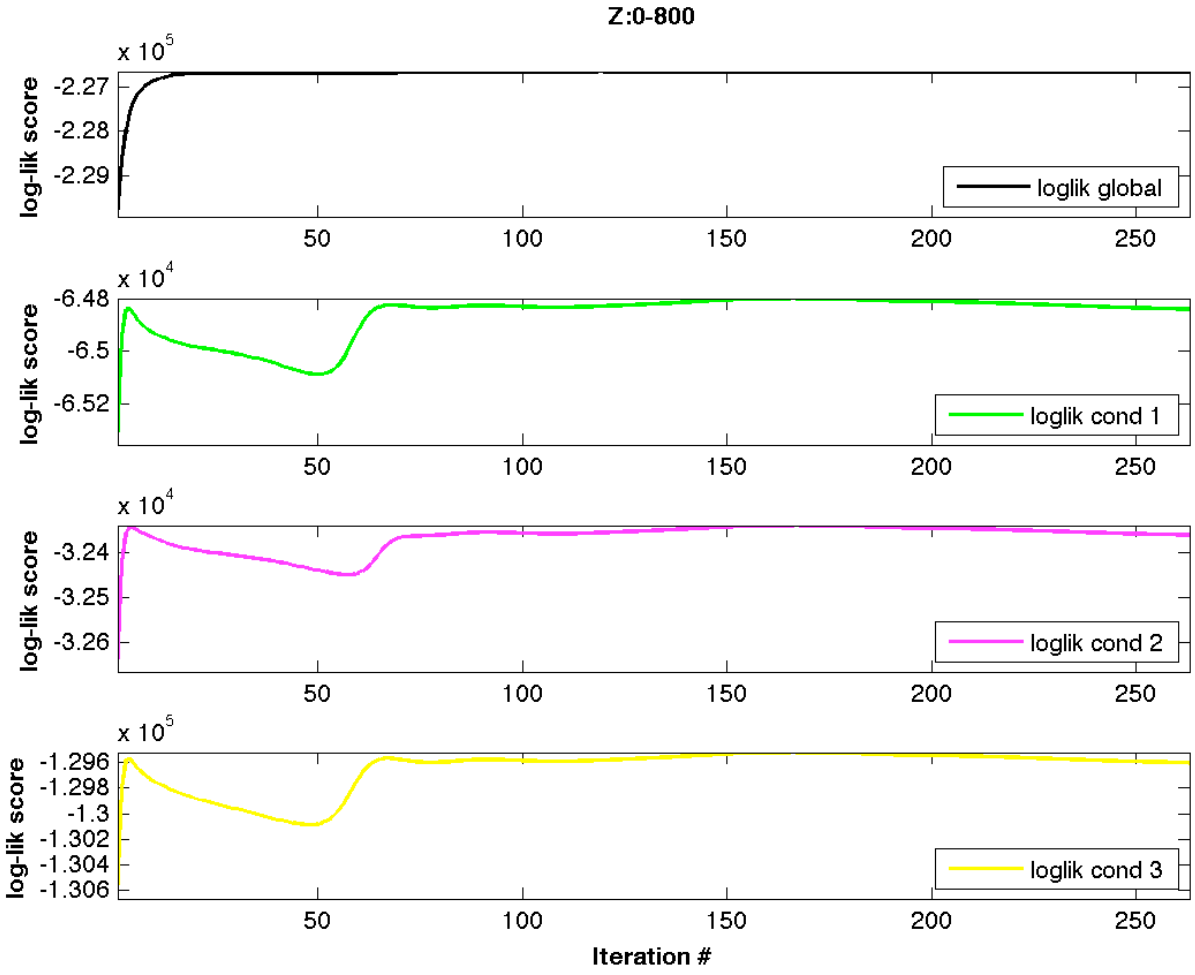


Figure 3-10: **Log-likelihoods of the data for the test region Z:0–800. Log-likelihoods are not guaranteed to increase for condition–specific data.**
The log-likelihood of the aggregated data (upper subplot in black color) is guaranteed to increase since the EM convergence criterion is based on this. However, the log-likelihoods of the condition–specific data initially increase and then decrease for some period since the setting of values is not optimal for each condition separately but for all the data. After some "burn-in" period, since the sparseness effect has been well spread out, the log-likelihods for each condition start to rise again.

significant ones for the first condition were at positions {132, 216, 498, 507}. Especially, for the last two, not only they do not agree with the true event at position 500, but they come in contrast with biological intuition since two events are not expected to be in such a close range.
For the second condition, it is even worse since the seven most significant events are at positions {93, 94, 448, 454, 455, 457, 458}. Of course, events {93, 94} and {448, 454, 455, 457, 458} are close to the true ones 70 and 450, respectively, but again, there is a very close distance between some predicted events, something which is not biologically interpretable.
Lastly, the five most significant events were at positions {222, 224, 244, 600, 606}. Again, events at positions {222, 224, 244} and {600, 606} seem to correspond to true events at positions 230 and 600, but the problem of closely called events pertains. Besides, events are not called at positions way off the true ones, as expected since the Simple Mixture model (with a sparse prior) is performed on each condition separately.

On the contrary, the Temporal Coupling algorithm achieves better results by identifying fewer events and much closer to the true ones. More specifically, 5, 3 and 3 events are detected for the first, second and third condition, respectively. The three most significant in the first condition are at positions {117, 232, 453}, the two most significant ones for the second condition at {117, 453} and for the third at {232, 605}. The algorithm tries hard to align events even if they are not aligned in reality. The alignment is biased towards the greatest event (in terms of position) of the group of events that are closest together across conditions.

So, in the case of events at positions 100 and 70 (in the first two conditions), the place of the alignment of the predicted events would be closer to 100 (because of the nature of the read distribution that favors events downstream of a read as discussed earlier). In addition, since there are also events in positions 200 and 230 for the first and third condition, respectively, the alignment will move further towards point 200, since the assignment of reads in the range [0–200] is also affected by these events. Thus, this results in an event at position 117 for the first and second condition which is fairly close to the true one in the first condition (at

position 100), but a bit distant to the corresponding in the second condition (at position 70). In addition, no event is predicted around this location in the third condition supporting the fact that no true event was present around that location.

The story is similar for the next event predicted at position 232, which is 32 and 2 bp away from the true events on the first and third condition, respectively. In addition, an event at position 453 is found, only 3 bp away from the corresponding true event of the second condition (at position 450). However, this peak is 47 bp away from the true event on the first condition (at position 500). Lastly, an event in position 605 in condition 3 is detected being just 5 bp distant from the true one on this condition (at position 600).

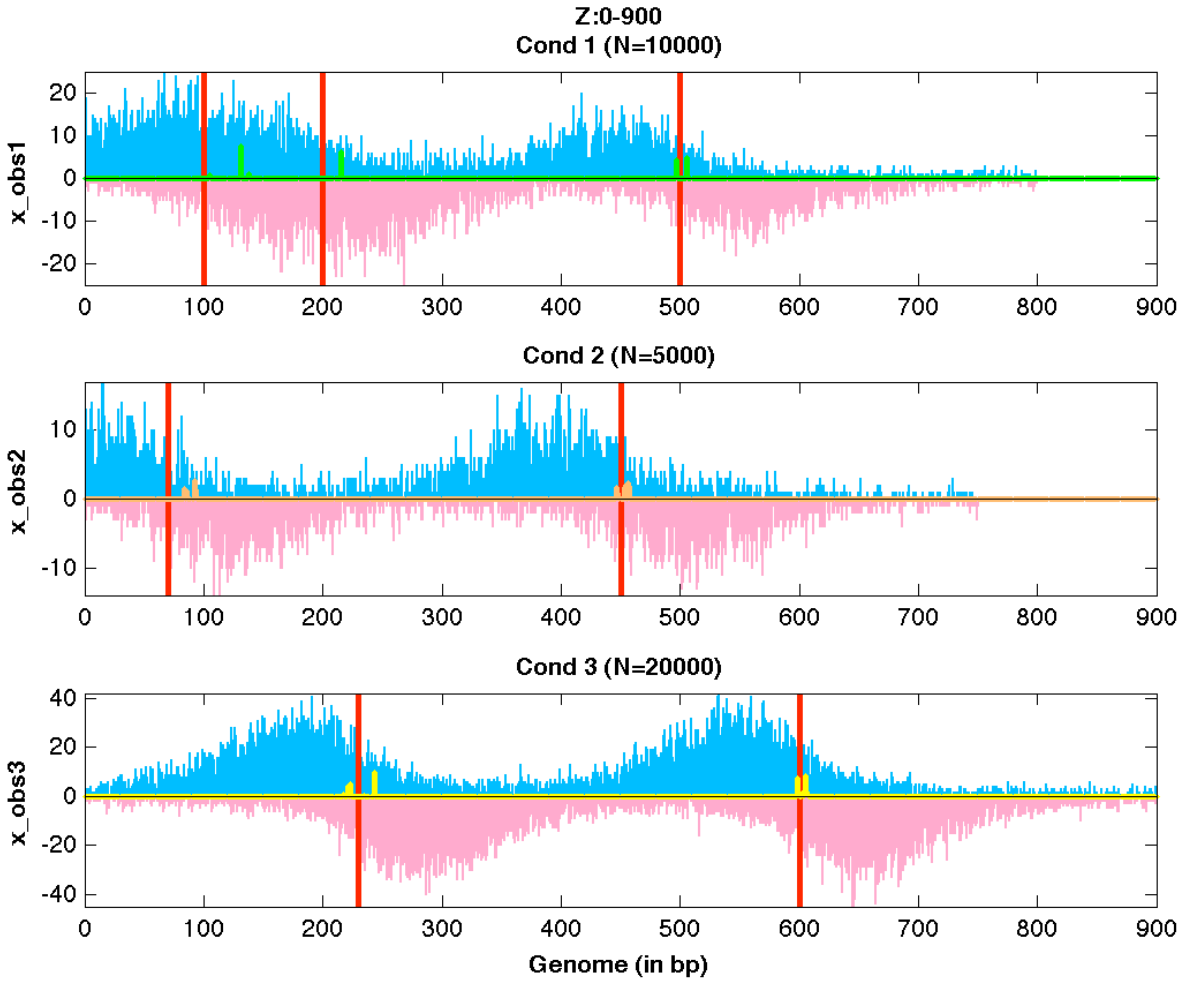


Figure 3-11: **Simple Mixture model on the test region Z:0–900.**
The Simple Mixture model discovers many more events than the true ones. Furthermore, there are events called in such a close distance something that comes in contrast with biological intuition. Again, as in the previous case, events are not aligned across conditions. However, an advantage is that no events are called in positions far away from the true ones. The true events are located at positions {100, 200, 500}, {70, 450} and {230, 600} for the first, second and third condition, respectively. The counts of forward ('+') reads are depicted with light blue color, while the counts of reverse ('−') strands with pink. The red bars represent the positions of the true events, while differently colored bars predicted peaks. The height of a bar indicates the strength of the event.

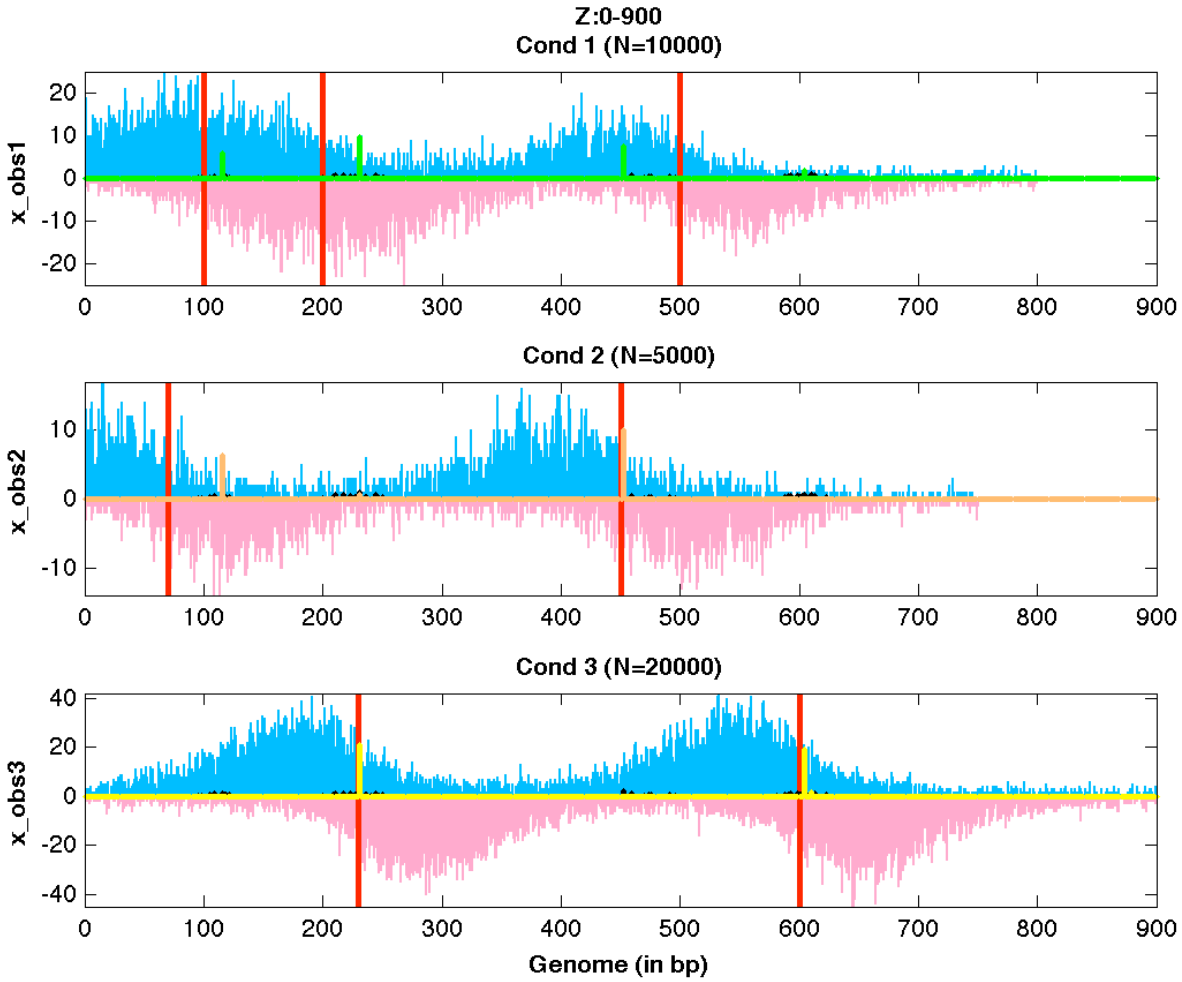


Figure 3-12: **Temporal Coupling model on the test region Z:0–900.**
In the Temporal Coupling case, far fewer events are discovered. Therefore, the algorithm achieves better sparseness. Moreover, the events are aligned across conditions, seeking for a compromise between closely located events in different conditions. In other words, the predicted event is placed in a position between the close true peaks (different conditions). Besides, events are generally called near to their true positions. So, when an event is missing from a condition, it is not being called (in that condition) even if there are true events at the same place in other conditions.
The true events are located at positions {100, 200, 500}, {70, 450} and {230, 600} for the first, second and third condition, respectively. The counts of forward ('+') reads are depicted with light blue color, while the counts of reverse ('−') strands with pink. The black bars indicate the events discovered from the aggregation of data in all conditions and represent the candidate events that each condition can have. The red bars represent the positions of the true events, while differently colored bars predicted peaks. The height of a bar indicates the strength of the event.

Furthermore, to compare the performance of both methods, we created a dataset of three conditions, each assigned 2,500, 1,250 and 5,000 reads, respectively and we run it on each method for 1,000 repetitions. We selected three criteria for comparison; the distance between predicted and true events, the distinct values of distances of a predicted to a true event and the number of predicted events per true event. Moreover, a predicted event was considered to correspond to a true event if it was within distance of 40 bp from it. Also, two predicted events were assumed to be very close to each other, if they were assigned on the same event and they were within distance of 20 bp to each other.We first run the dataset on a region where all true events were aligned (Z:0–800) and a region where events were not aligned (Z:0–900).

In region Z:0–800, the relative distances were comparable as shown in the histogram of Figure 3-13. However, the Temporal Coupling is more robust in locating events in the same position across repetitions as illustrated in Figure 3-14b. In addition, the Temporal Coupling identifies much fewer false events as shown in Figure 3-14c. That is because the alignment of events across conditions achieved by the Temporal Coupling prevents the calling of spurious condition specific events.

In region Z:0–900, the mean distance of predicted to true events is smaller in the case of Simple Mixture. That is because, the true events are not aligned, so the Temporal Coupling makes a big compromise in trying to align the predicted events. This causes some events to be far away from their assigned true events. In addition, this creates a bigger range of distances between a predicted and a true event resulting in the Temporal Coupling having more distinct values of distances than the previous case of Z:0–800. See Figure 3-16b. However, the Temporal Coupling still does as satisfactorily in terms of the number of predicted events per true event as in the previous case of Z:0–800. See Figure 3-16c. As aforementioned, this is due to the intention of Temporal Coupling to align events which results in ignoring spurious ones.

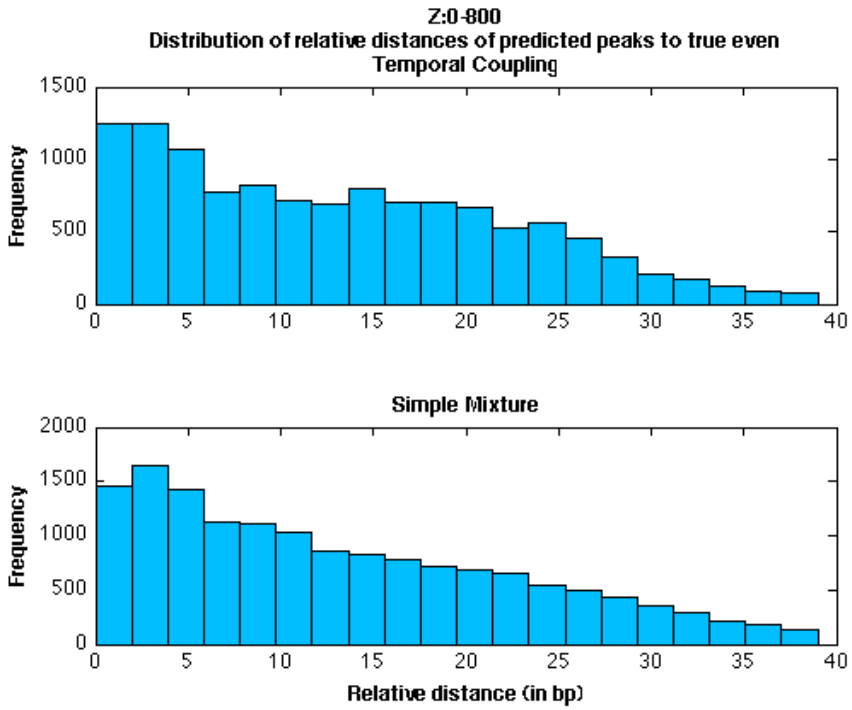


Figure 3-13: **Histogram of relative distances between predicted and true events for the region Z:0–800.**
This figure shows the histogram of relative distances between predicted and true events for the region Z:0–800 running on both the Temporal Coupling and the Simple Mixture for 1,000 repetitions. There are more counts under the histogram corresponding to the Simple Mixture since more events are predicted in this case. As it is shown, both methods perform similarly. The mean distances for Temporal Coupling and Simple Mixture are the same, namely 17 bp.
The dataset was comprised of 2,500, 1,250, 5,000 reads for conditions 1, 2, 3, respectively. For each condition, we set the number of forward and reverse reads to be equal.

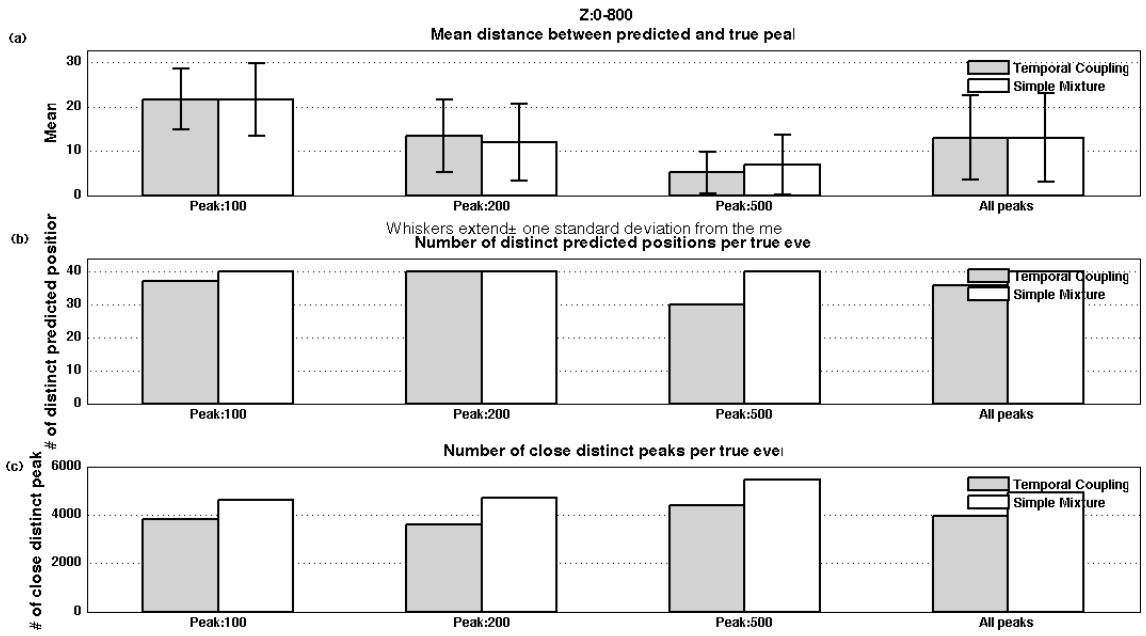


Figure 3-14: **Performance criteria for the Temporal Coupling and the Simple Mixture for the region Z:0–800.**
This figure compares Temporal Coupling and Simple Mixture on several performance criteria for the region Z:0–800 for 1,000 repetitions.
(a) The mean distance of predicted to true events is similar for both methods. This is justified since both methods try to discover events by using the same underlying assumption, that is, the assignment of a read to an event is independent on the assignment of the others. In addition, since true events are aligned across conditions, Temporal Coupling does not have to make a big compromise in aligning them.
(b) Temporal Coupling seems to be more robust than the Simple Mixture since the number of distinct distance values between a predicted and a true event is less than the corresponding number when run on Simple Mixture.
(c) Temporal Coupling performs even better when we compare the number of predicted events per true event. Temporal Coupling is more parsimonious in terms of the number of events it predicts compared to the Simple Mixture one. As shown in the subfigure, there are almost 4,000 and 5,000 predicted peaks assigned to true events across all conditions for the Temporal Coupling and the Simple Mixture, respectively. This accounts to 1.3 and 1.7 predicted events per true event for the Temporal Coupling and the Simple Mixture, respectively.
A predicted event was assumed to belong to a true one if it was within distance of 40 bp to it. Two predicted peaks were assumed to be very close to each other if they were assigned on the same event and they were within distance of 20 bp to each other. The dataset was comprised of 2,500, 1,250, 5,000 reads for conditions 1, 2, 3, respectively. For each condition, we set the number of forward and reverse reads to be equal.

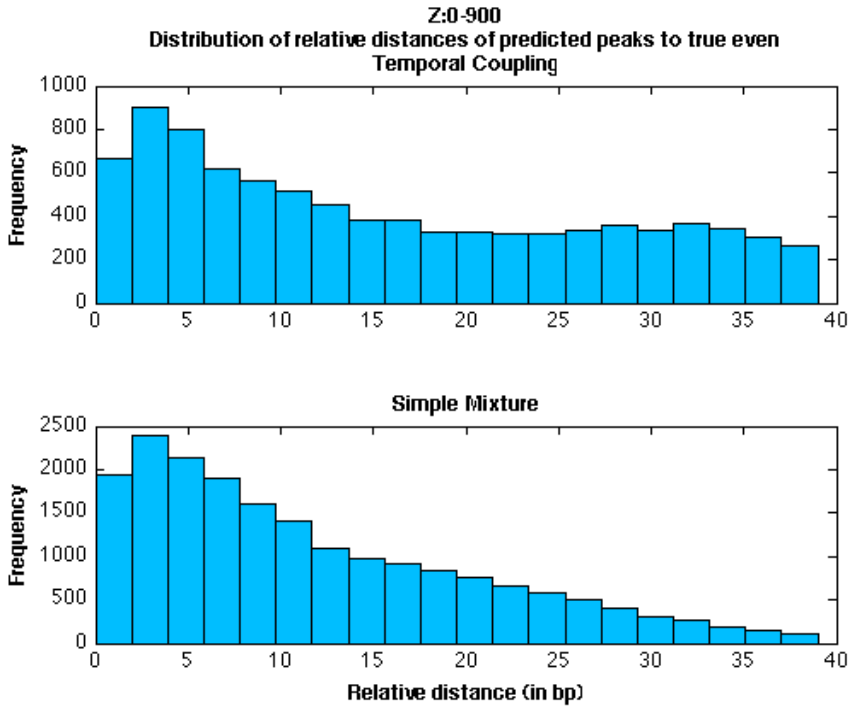


Figure 3-15: **Histogram of relative distances between predicted and true events for the region Z:0–900.**
This figure shows the histogram of relative distances between predicted and true events for the region Z:0–900 running on both the Temporal Coupling and the Simple Mixture for 1,000 repetitions. There are more counts under the histogram corresponding to the Simple Mixture since more events are predicted in this case. Simple Mixture provides better results than the Temporal Coupling because in Simple Mixture events in each condition are predicted based only on the condition-specific data, while in Temporal Coupling predicted peaks are the result of a compromise between differences across data in different conditions. In the case of region Z:0–900 where the true peaks are not aligned across conditions this creates a larger offset between predicted and true events in each condition. The mean distances for Temporal Coupling and Simple Mixture are 16 and 11 bp, respectively.
The dataset was comprised of 2,500, 1,250, 5,000 reads for conditions 1, 2, 3, respectively. For each condition, we set the number of forward and reverse reads to be equal.

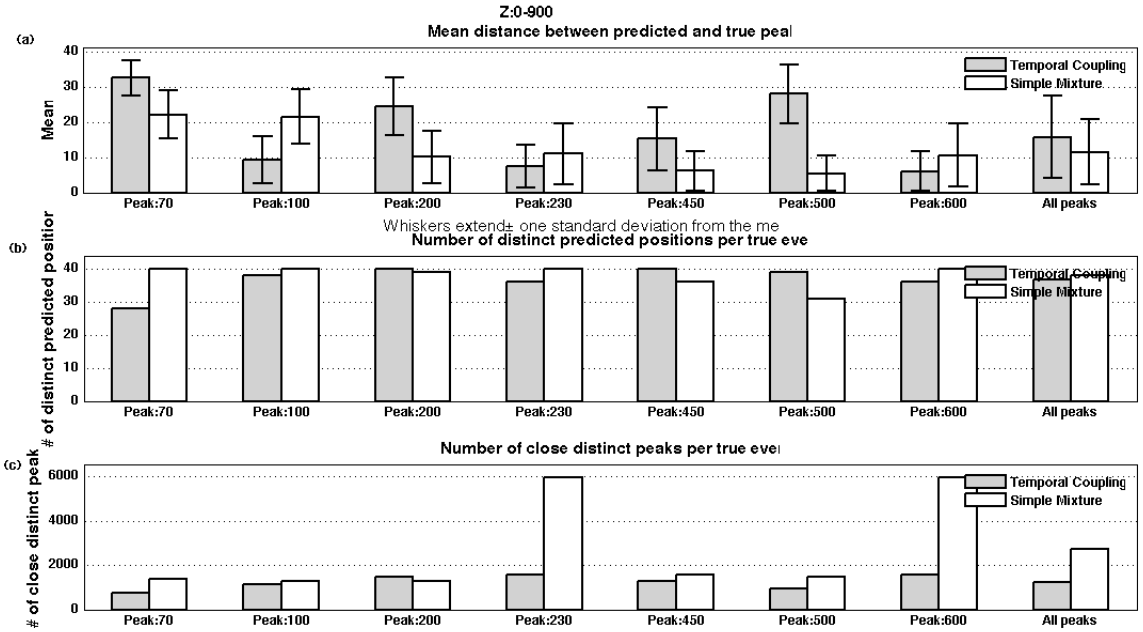


Figure 3-16: **Performance criteria for the Temporal Coupling and the Simple Mixture for the region Z:0–900.**
This figure compares Temporal Coupling and Simple Mixture on several performance criteria for the region Z:0–900 after running it on each method for 1,000 repetitions.
(a) The mean distance of predicted to true events is lower in the case of Simple Mixture. Because in region Z:0–900, the events were deliberately chosen not to be aligned, this forces the Temporal Coupling method to make a big compromise in positioning the predicted events in order to make sure that they will be aligned across conditions. This results in the placement of predicted peaks far away from the true events' locations.
(b) Temporal Coupling and Simple Mixture perform similarly regarding the number of distinct values between a predicted and a true event. True events are not aligned while not all events are present in all conditions. This forces predicted events to be pulled towards only one true event around a region with more than one events across conditions. For example, in region Z:0–900, there is a true event located at position 100 in condition 1 and a true event at position 70 in condition 2. Therefore, when the algorithm tries to locate the predicted events, it inevitably forces some predicted events far way from the true ones by trying to align events together. This increases the potential range of the distance of a predicted to a true event, thus resulting in the dispersion of relative distance values.
(c) In Temporal Coupling less predicted peaks are discovered per event in all cases except one (peak at location 200). There are peaks (at positions 230 and 600), where there is a big overestimation on the number of events predicted around these locations. The Temporal Coupling, however, keeps the number of predicted events (assigned to a single true event) low in all cases.

Lastly, we attempted to assess the performance of each algorithm for different numbers of reads. We created datasets of three conditions on a varying number of reads spanning from 20 up to 7,500 reads for each condition. We run each dataset on each method for 10 repetitions and kept the average of the value that we were interested in. We compared the mean distance between predicted and true events, the True Positive (TP) rate (known also as sensitivity) and the False Positive (FP) rate (inverse of specificity) in two regions, one where true events were aligned (Z:0–800) and one that were not (Z:0–900).

Since the value of alpha (α) essentially represents the minimum number of reads that a component should have in order to be identified as an event, many more false positive events would be expected to be called for an increasing number of reads. That is because components near true events are assigned a larger count of reads as the size of the dataset increases which exceeds the value of alpha by a large number especially if it is small value. Thus, these events are eventually called because the number of their assigned reads did not become less than alpha before the termination of the algorithm. This comes in contrast with common logic since we expect the algorithm to perform better as more and more data points are available. For this reason, we devised a way of evaluating alpha dependent on the number of reads of the dataset and (the assumed) number of true events.

Briefly, we assume that we have N reads, the number of true events is K, and the events are divided between 'strong' and 'weak' ones. Let's also assume that a 'strong' event is ν times stronger than a 'weak' one. We are interested in finding a value of alpha such that it will be able to capture even the worst case which is the one when all true events are strong but one. That is, we seek the least value of alpha such that it allows the discovery of true events but is also big enough to avoid false positives. Skipping the derivation steps, the value of alpha is given by:

$$\alpha = \frac{N}{\nu K - (\nu - 1)}$$

In our case, $K = 3$ and ν is set to 10.

In region Z:0–800, the mean distance of the predicted to true events was lower in the case of Temporal Coupling as shown in Figure 3-17. When true events are aligned, then the

Temporal Coupling by considering reads in all conditions aggregated and in each condition separately achieves first alignment and manages to locate them closely to the true ones. In addition, the TP rates is 16% higher and the FP rate is 10 times lower.

In region Z:0–900, the mean distance in Simple Mixture is much lower than that of Temporal Coupling. The reason is because the Temporal Coupling tries to align events even if they are not truly aligned (as in this case). This results in some predicted events being far away from their corresponding true ones. Nevertheless, the TP rate is only marginally higher in Temporal Coupling and for medium or large number of reads, it is by 5-10% higher.

Regarding the running time of each region, it was on the order of minutes for both methods. However, the constant factor for the Temporal Coupling was larger since in this case, an additional step of running EM on all data at each iteration has been involved. More specifically, the running times for regions {Z:0–800, Z:0–900} were {0.9 min, 1.1 min} and {2.1 min, 1.4 min} for the Simple Mixture and the Temporal Coupling model, respectively. The examples were run on a Mac OS X 10.5 (2.4 GHz Intel Core 2 Duo, 4 GB memory).

In summary, the Temporal Coupling method achieves better sparseness than the Simple Mixture one. It also tries hard to align events in order to satisfy the patterns of data for all conditions simultaneously, but at the same time avoids to call peaks in places which do not correspond to real events. Besides, a similar case to the one previously described holds for the log-likelihood. That is, condition–specific log-likelihood are not guaranteed to constantly increase after each of the algorithm's iteration since the convergence criterion is defined on all data. (Data not shown here).

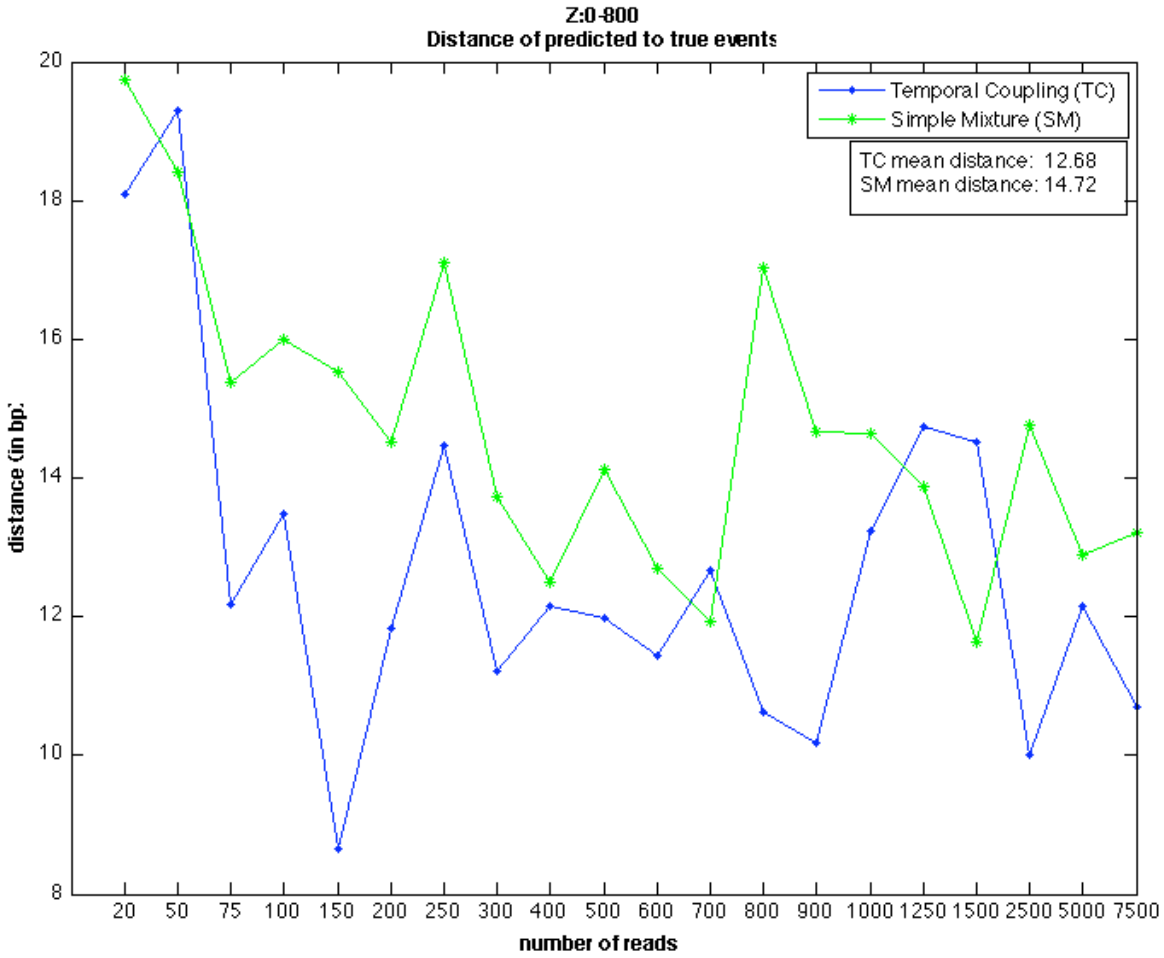


Figure 3-17: **Distance of predicted to true events for different numbers of reads for the region Z:0–800.**
This figure shows the mean distance of predicted to true events for both Temporal Coupling and Simple Mixture on different numbers of reads for the region Z:0–800. The Temporal Coupling method appears to outperform the Simple Mixture since almost for every different number of reads it gives a lower distance of predicted to true events. On average, the mean distance of a predicted to a true event is 12.7 bp in the case of Temporal Coupling, while 14.7 bp in the case of Simple Mixture. That is, on average the events are placed 2 bp closer to their true positions in the case of Temporal Coupling.
The dataset was comprised of three conditions each having the same number of reads. We run each dataset ten times on each method and kept the average distance.

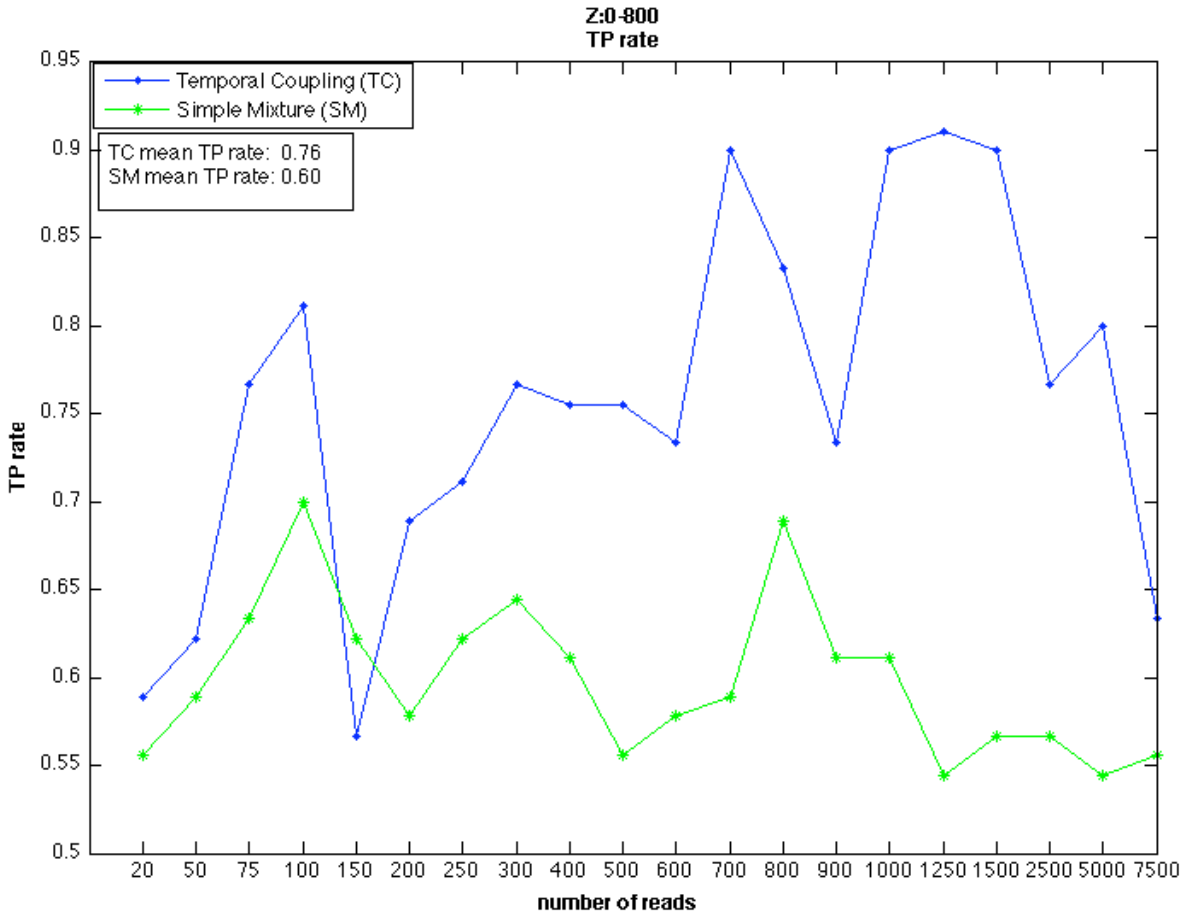


Figure 3-18: **True positive rate (sensitivity) for different numbers of reads for the region Z:0–800.**
This figure shows the true positive rate (sensitivity) for both Temporal Coupling and Simple Mixture on different numbers of reads for the region Z:0–800. TP rate is almost by 30% greater in Temporal Coupling. We furthermore see that TP rate shows a more rapid increase as the number of reads increases, while TP rate in Simple Mixture is kept almost on a constant level for varying number of reads. Besides, the TP rate does not show a monotonical increase since the value of alpha might have been set too high for some number of reads.
The dataset was comprised of three conditions each having the same number of reads. We run each dataset ten times on each method and kept the average TP rate.

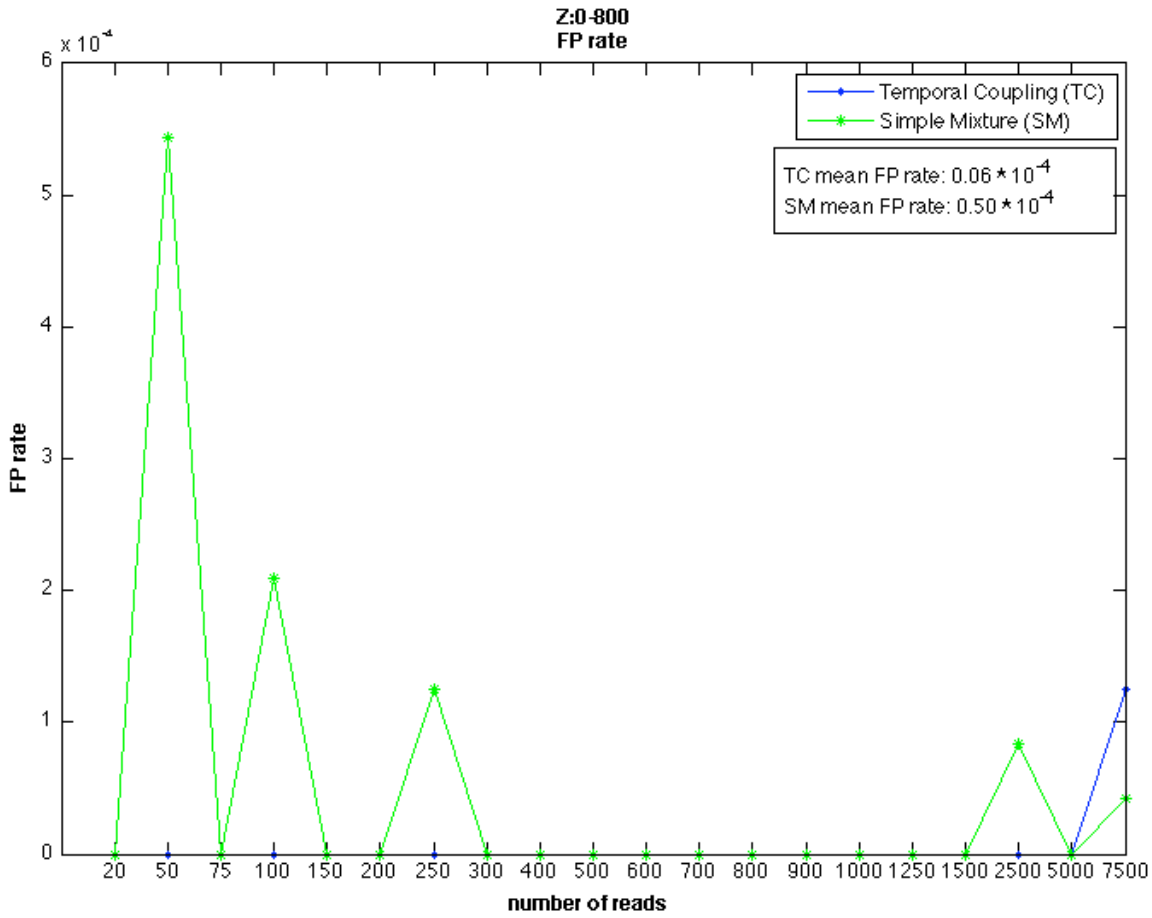


Figure 3-19: **False positive rate (inverse of specificity) for different numbers of reads for the region Z:0–800.**
This figure shows the false positive rate (inverse of specificity) for both Temporal Coupling and Simple Mixture on different numbers of reads for the region Z:0–800. As it is shown, FP rate is kept very low almost in all cases which indicates that the value of alpha has been set satisfactorily high enough to avoid the calling of spurious events. In addition, we see that Temporal Coupling does even better than Simple Mixture on that as it achieves an average FP rate ten times lower than the one found in the Simple Mixture.
The dataset was comprised of three conditions each having the same number of reads. We run each dataset ten times on each method and kept the average FP rate.

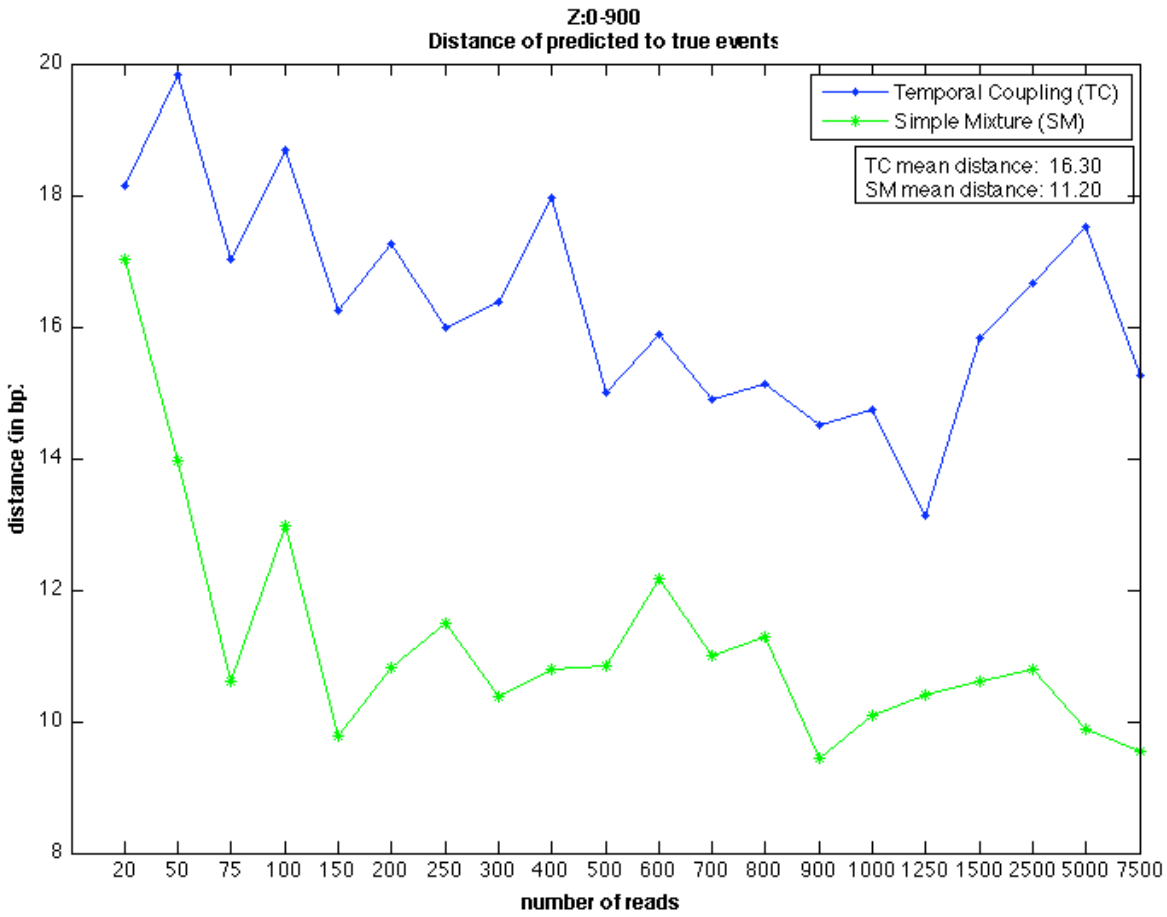


Figure 3-20: **Distance of predicted to true events for different numbers of reads for the region Z:0–900.**
This figure shows the mean distance of predicted to true events for both Temporal Coupling and Simple Mixture on different numbers of reads for the region Z:0–900. As it was expected, the Simple Mixture clearly outperforms the Temporal Coupling one. That is because the true events are not aligned, and therefore the Temporal Coupling makes a big compromise in trying to keep the predicted events aligned. This results in some predicted events being far away from their assigned true events. On the other hand, since the Simple Mixture does not enforce alignment and only predicts events based on each condition separately, it locates events much closer to their true positions. On average, the mean distance of a predicted to a true event is 16.3 bp in the case of Temporal Coupling, while 11.2 bp in the case of Simple Mixture.
The dataset was comprised of three conditions each having the same number of reads. We run each dataset ten times on each method and kept the average distance.

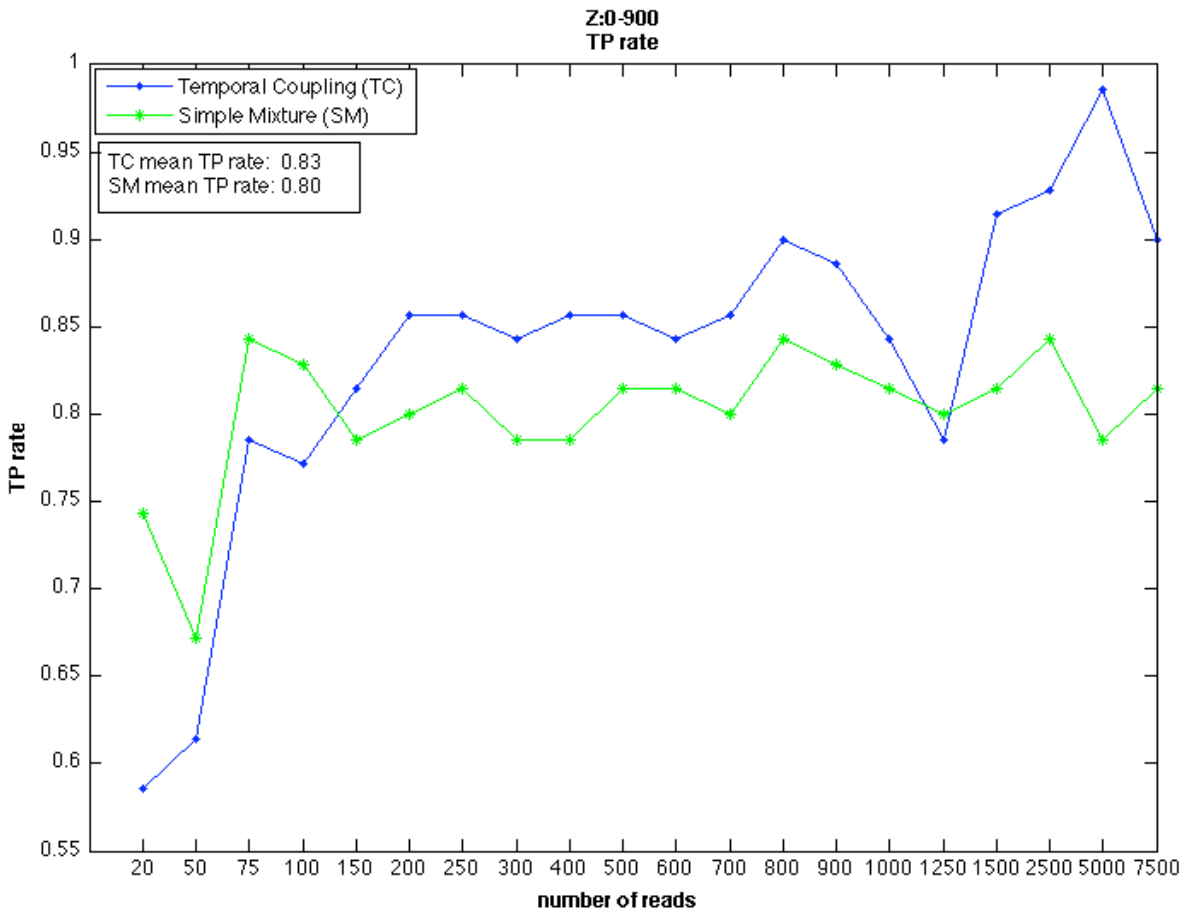


Figure 3-21: **True positive rate (sensitivity) for different numbers of reads for the region Z:0–900.**
This figure shows the true positive rate (sensitivity) for both Temporal Coupling and Simple Mixture on different numbers of reads for the region Z:0–900. Even in this case, where the true events are not aligned, the Temporal Coupling generally outperforms the Simple Mixture. For medium or large number of reads (> 100), the TP rate is on the order of 5-10% higher in the case of Temporal Coupling. Since reads from all conditions are taken into consideration as an aggregate in the case of Temporal Coupling it is more probable to recover even the weakest events.
The dataset was comprised of three conditions each having the same number of reads. We run each dataset ten times on each method and kept the average TP rate.

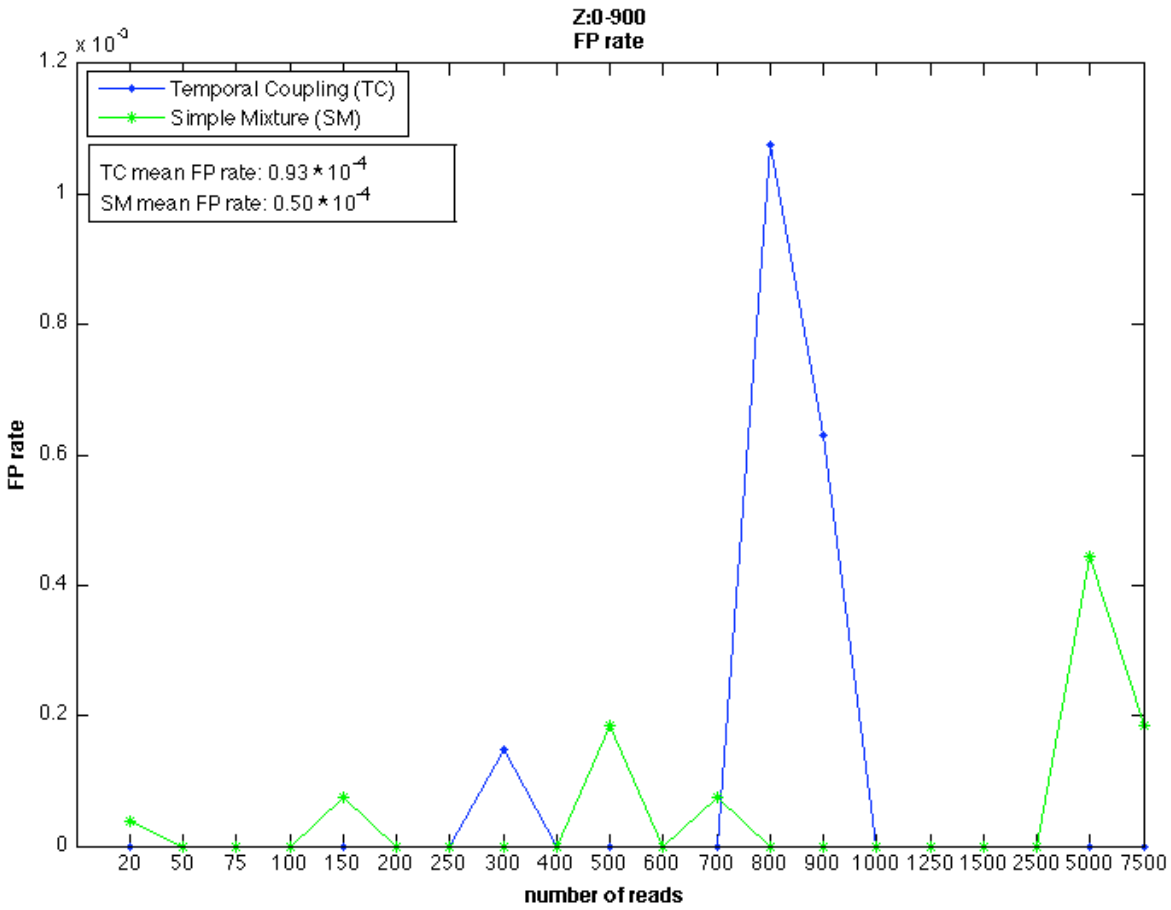


Figure 3-22: **False positive rate (inverse of specificity) for different numbers of reads for the region Z:0–900.**
This figure shows the false positive rate (inverse of specificity) for both Temporal Coupling and Simple Mixture on different numbers of reads for the region Z:0–900. As it is shown, FP rate is kept very low almost in all cases (in Temporal Coupling except for three) which indicates that the value of alpha has been set satisfactorily high enough to avoid the calling of spurious events.
The dataset was comprised of three conditions each having the same number of reads. We run each dataset ten times on each method and kept the average FP rate.

3.3.2 Real Data

For our analysis, we used RXR ChIP-Seq data in 6 conditions, presented in [15]. The value of the sparse prior was set to 10 for both methods ($\alpha = 10$. See Subsection 2.1.3). Furthermore, for better visualization, only the last three conditions which are the most abundant in terms of events will be depicted.

We first examined the region 2:98,462,850–98,464,098 on the Simple Mixture model (Figure 3-23). Four, four and three events were discovered at positions {98,463,248, 98,463,315, 98,463,612, 98,463,799}, {98,463,242, 98,463,310, 98,463,610, 98,463,792} and {98,463,251, 98,463,629, 98,463,796}, for Day 3, Day 4 and Day 6, respectively. The corresponding events across conditions are not well aligned to each other, having a distance of ± 6 bp between each other.

On the contrary, Temporal Coupling discovers events that are all aligned at positions {98,463,245, 98,463,576, 98,463,637, 98,463,796} (Figure 3-24). The event in 98,463,576 is not present in Day 6 since there are less than six reads at each position around this location for this condition. As previously, the log-likelihoods of the condition–specific data are not guaranteed to increase at each iteration, since the convergence threshold is defined on all data (Figure 3-25).

Here, it is worth noting that there are events at some locations which are not discovered by the two algorithms. For example, in positions around 98,463,350, 98,463,550. This is not due to the algorithms' inadequacy of detecting the events but to the value of alpha (value of the sparse prior) chosen each time. Essentially, the value of alpha represents the minimum number of reads that an event should have to be called. So, since in this case the number of reads around these positions does not exceed the number of 10, the events could not be detected. Instead, these reads are assigned to the nearest stronger events which were already assigned a number of reads greater than the value of alpha. In general, larger values of alpha favor fewer number of events, while lower values fewer events.

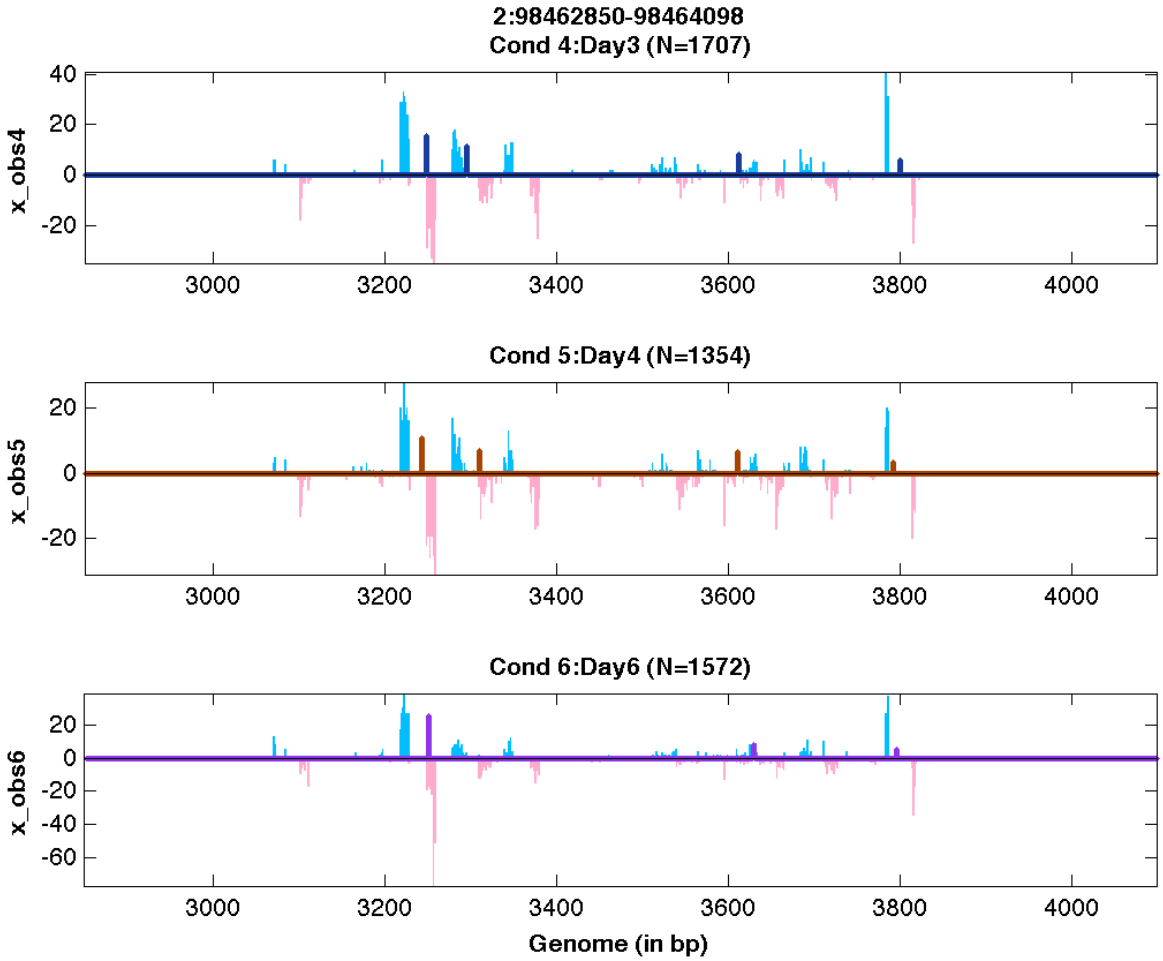


Figure 3-23: **Simple Mixture model fails to align events on the region 2:98,462,850–98,464,098.**
The predicted events are located at {98,463,248, 98,463,315, 98,463,612, 98,463,799}, {98,463,242, 98,463,310, 98,463,610, 98,463,792} and {98,463,251, 98,463,629, 98,463,796} for Days 3, 4 and 6, respectively. Events are not aligned across conditions due to the slightly different nature of the data for each condition something that comes in contrast with biological intuition.
The counts of forward ('+') reads are depicted with light blue color, while the counts of reverse ('−') strands with pink. Differently colored bars represent predicted peaks. The height of a bar indicates the strength of the event.

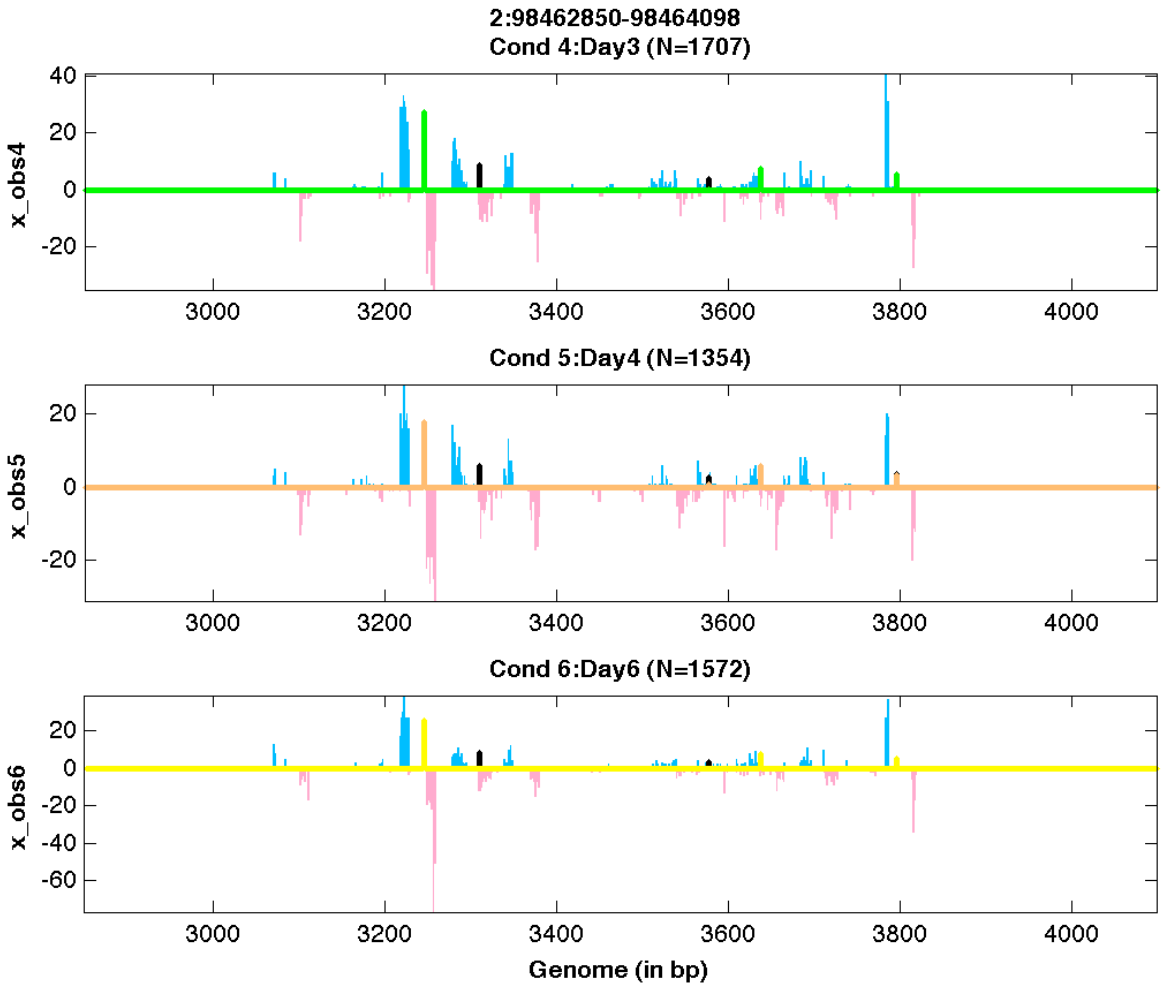


Figure 3-24: **Temporal Coupling model manages to align events on the region 2:98,462,850–98,464,098.**
Events are discovered at positions {98,463,245, 98,463,576, 98,463,637, 98,463,796}. They are aligned across conditions concurring to the fact that the TF motif remains the same in different conditions.
The counts of forward ('+') reads are depicted with light blue color, while the counts of reverse ('−') strands with pink. Differently colored bars represent predicted peaks. The height of a bar indicates the strength of the event.

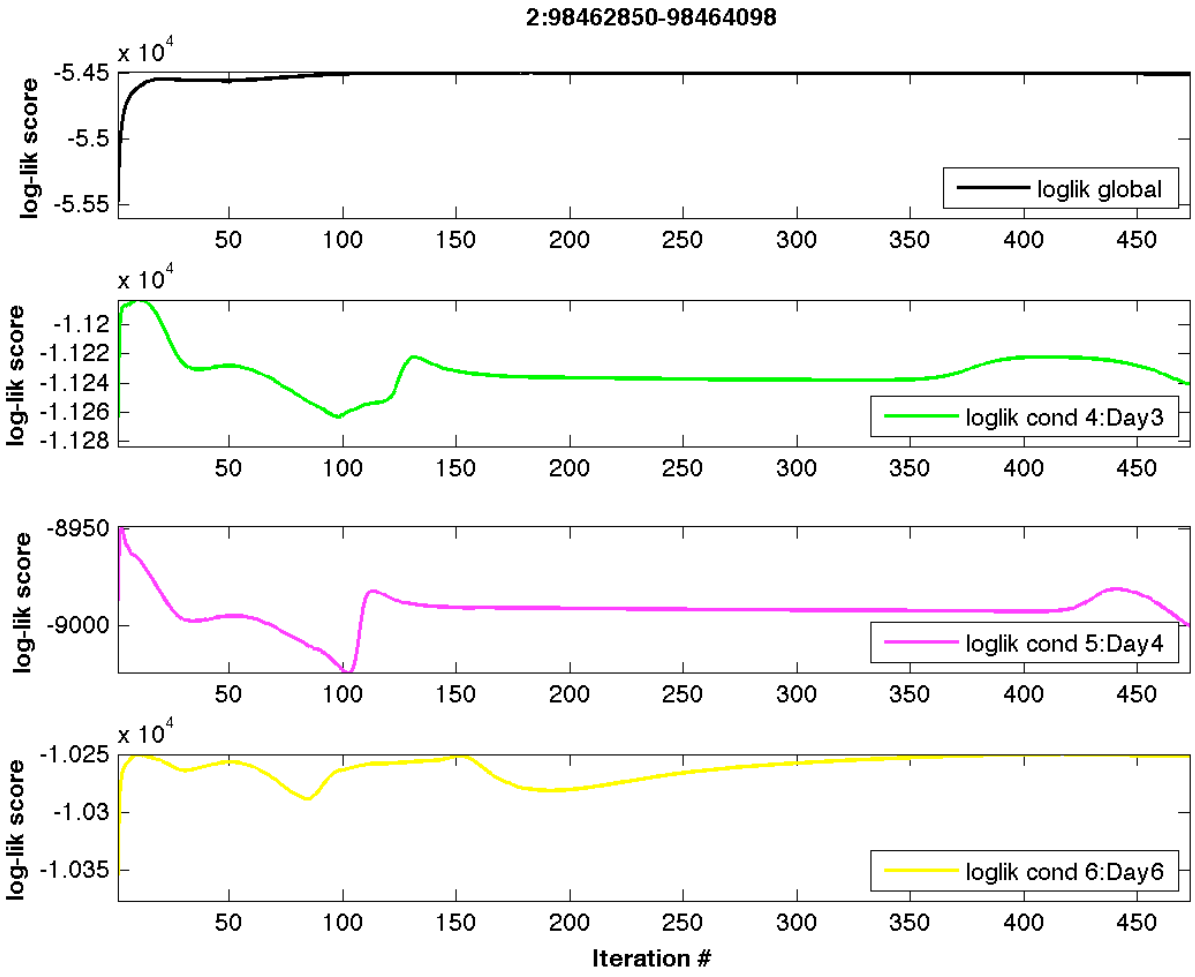


Figure 3-25: **Log-likelihoods of the data for the region 2:98,462,850–98,464,098. Log-likelihoods are not guaranteed to increase for condition–specific data.** The log-likelihood of the aggregated data (upper subfigure) is guaranteed to increase. However, this is not the case for the condition–specific log-likelihoods as well, since the convergence criterion is solely defined on all the data. Initially, the drop-off is steeper since the candidate events that the global mixture allows may not include the events that are optimal for each condition specifically. Nonetheless, after some iterations, the curve of each condition–specific log-likelihood becomes smoother since the set of candidate events after a large number of iterations is relatively small (due to sparseness) and does not change significantly. So, the optimal solution for each condition may indeed fall into the set of candidate events that the global mixture of the method allows.

The second examined region was 9:3,035,450–3,036,848 (Figure 3-26). First, it was run on the Simple Mixture model. The method seems to locate successfully events in regions highly enriched in reads, but still no alignment is achieved. In more detail, seven, five and four events are discovered in Days 3, 4 and 6. Four events are placed in very close locations across all conditions but with a deviation from each other of ± 8 bp which cannot be interpreted biologically. For instance, around location 3,035,900, events are called at positions 3,035,897, 3,035,904, 3,035,896 in Days 3, 4 and 6, respectively. Besides, two very close strong events are called at positions {3,036,552, 3,036,554} in Day 4, which is obviously an artifact, and an event around location 3,035,845 is missed in Day 6.

For the Temporal Coupling model, all events are aligned and located at positions {3,035,896, 3,036,085, 3,036,554} (Figure 3-27). Events around 3,035,850 and 3,036,380 seem to be missed by the method.

Lastly, we consider the region 9:35,054,500–35,055,573 (Figure 3-28). The Simple Mixture model identifies two events, at positions {35,054,938, 35,055,100}, {35,054,927, 35,055,087}, {35,054,963, 35,055,091} for Days 3, 4 and 6, respectively. As in the previous cases, events are not aligned to each other having a distance of ± 15 bp.

Since the read landscape is essentially gathered in a region of width 290 (from 35,054,860 to 35,055,150), and the window of the read distribution is 300 ($\max\{-299, 300\}$), the detection resolution can hardly be even finer. In simple words, the read distribution favors the detection of events which hold some distance between each other comparable to the window of the read distribution. Therefore, if there are non-continuous highly enriched locations within this window, then the reads are assigned to the strongest events abiding by this parsimonious policy.

When running it on the Temporal Coupling method, events are aligned and located at positions 35,054,935 and 35,055,084 (Figure 3-29). Similarly to the Simple Mixture model, two events are only detected due to the aggregation of events in a small region and the window of the read distribution.

Lastly, the running time of each region was on the order of seconds ($\sim$0.9–14.7 sec) for

both the Simple Mixture and the Temporal Coupling model. The experiments were run on a Mac OS X 10.5 (2.4 GHz Intel Core 2 Duo, 4 GB memory).

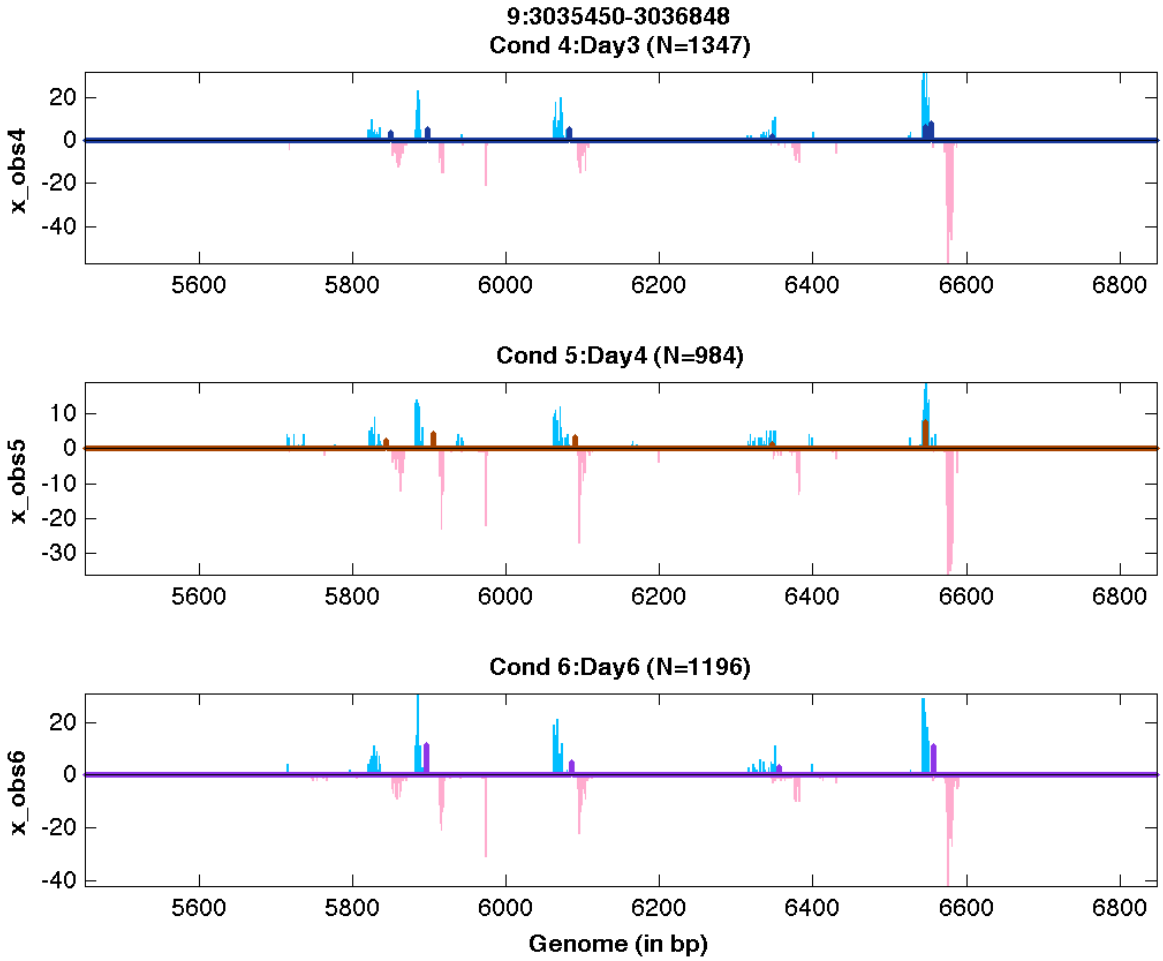


Figure 3-26: **Simple Mixture model does not enforce alignment on the region 9:3,035,450–3,036,848.**
Events are discovered in highly enriched locations but there is a deviation of ±8 bp between corresponding events across conditions, something which does not conform to biological intuition.
The counts of forward ('+') reads are depicted with light blue color, while the counts of reverse ('−') strands with pink. Differently colored bars represent predicted peaks. The height of a bar indicates the strength of the event.

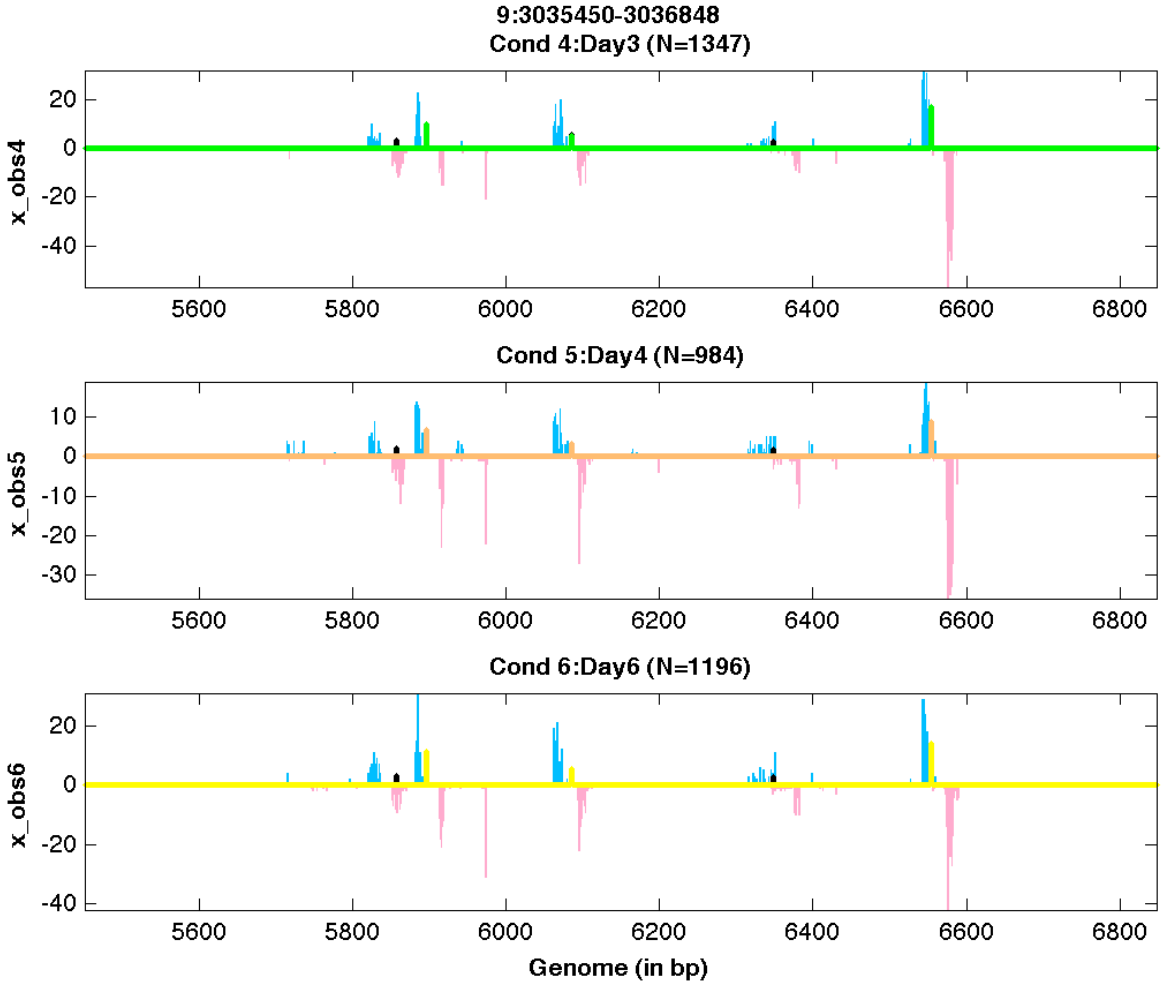


Figure 3-27: **Temporal Coupling model aligns events on the region 9:3,035,450–3,036,848.**
Events, which are all aligned, are discovered in three highly enriched locations. Two events around 3,035,850 and 3,036,380 seem to be missed by the algorithm.
The counts of forward ('+') reads are depicted with light blue color, while the counts of reverse ('−') strands with pink. Differently colored bars represent predicted peaks. The height of a bar indicates the strength of the event.

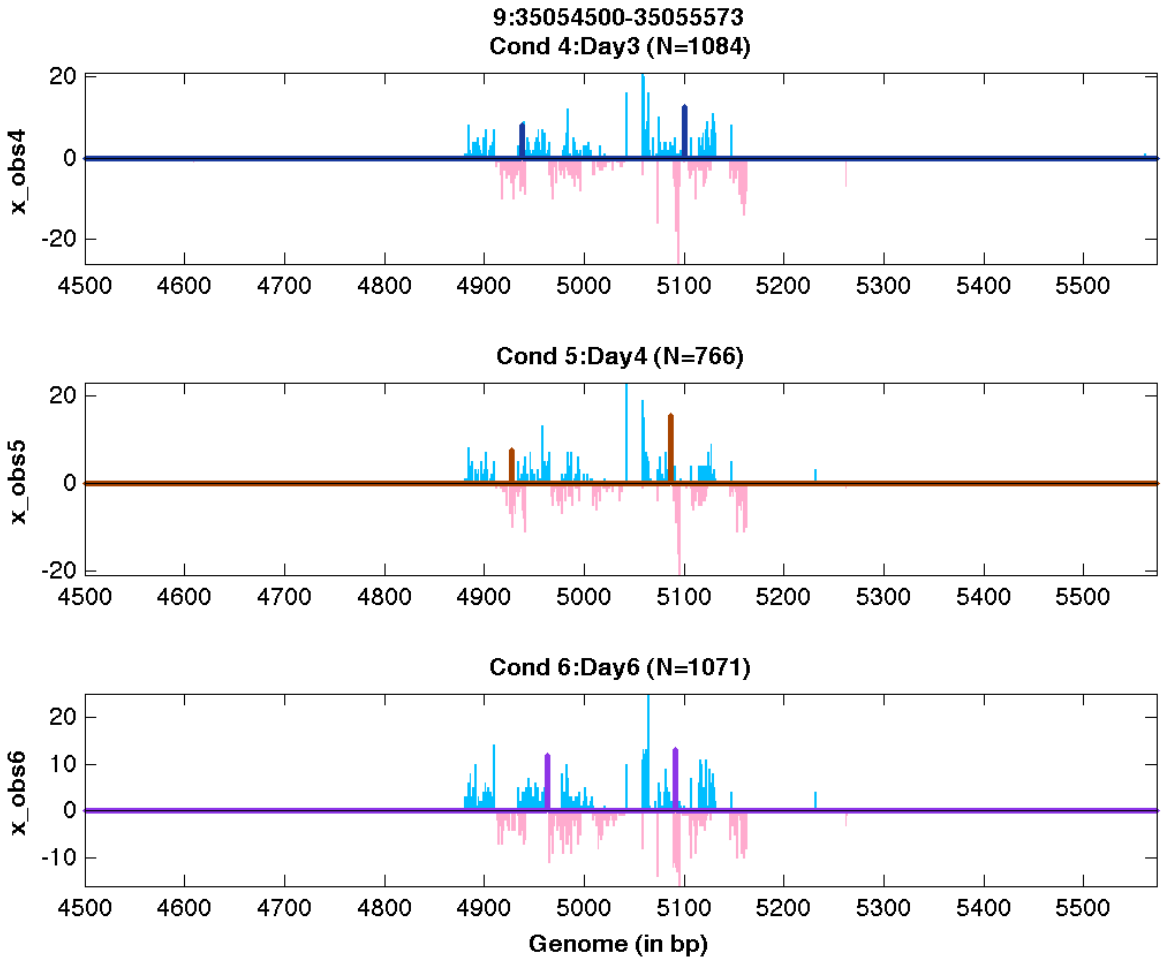


Figure 3-28: **Simple Mixture model fails to align events on the region 9:35,054,500–35,055,573.**
Events are detected at positions {35,054,938, 35,055,100}, {35,054,927, 35,055,087}, {35,054,963, 35,055,091} for Days 3, 4 and 6, respectively. As in the previous examples, alignment is not achieved.
The counts of forward ('+') reads are depicted with light blue color, while the counts of reverse ('−') strands with pink. Differently colored bars represent predicted peaks. The height of a bar indicates the strength of the event.

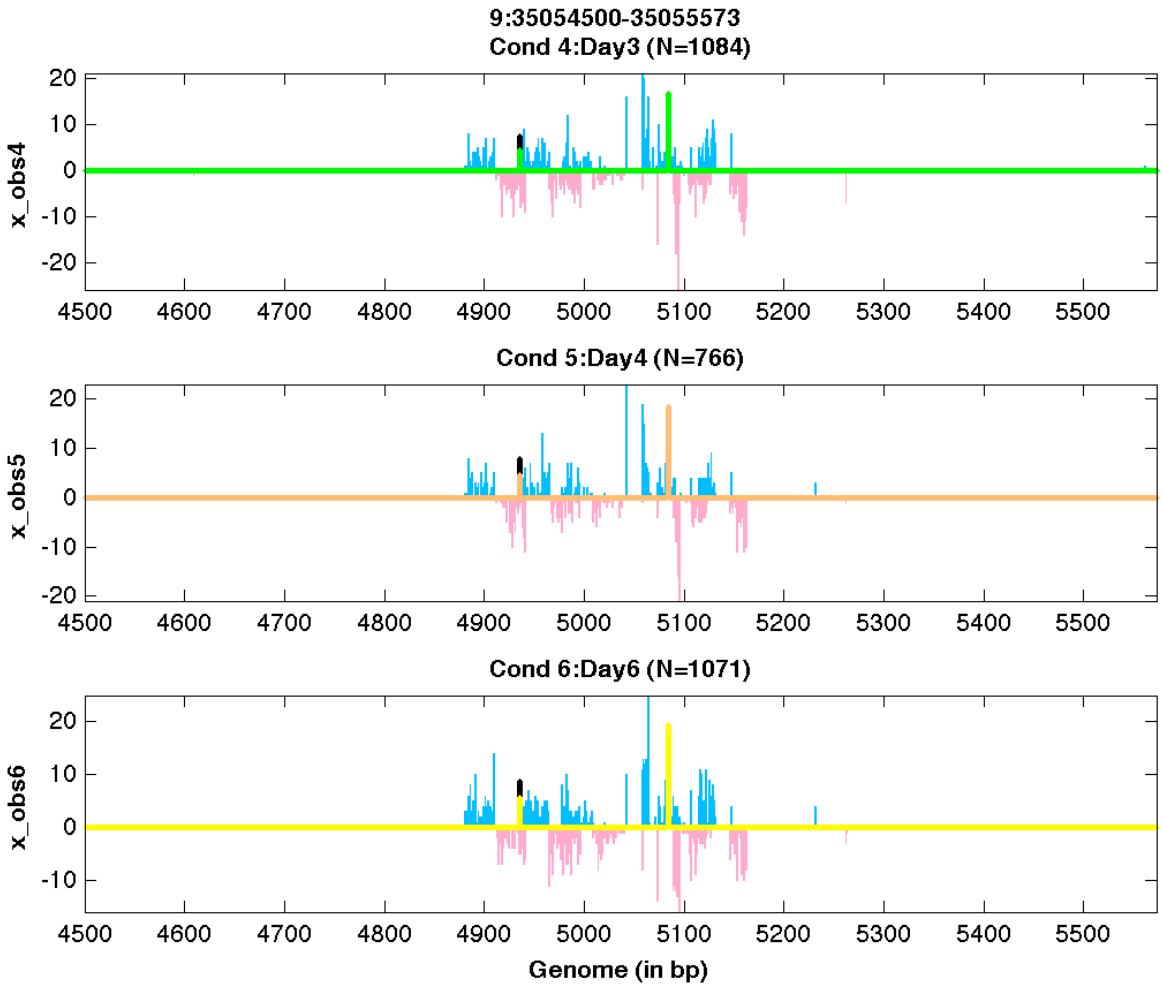


Figure 3-29: **Temporal Coupling model enforces alignment on the region 9:35,054,500–35,055,573.**
Events are detected at positions 35,054,935 and 35,055,084. Alignment is again guaranteed coming into agreement with the fact that TF motifs remain constant across different conditions.
The counts of forward ('+') reads are depicted with light blue color, while the counts of reverse ('−') strands with pink. Differently colored bars represent predicted peaks. The height of a bar indicates the strength of the event.

Chapter 4

Discussion

In this thesis, we dealt with the development of probabilistic graphical models for the identification of binding events in ChIP-Seq data showing that they can effectively simulate the read generating process. We aimed at two different problems; first, to discover binding events in a single condition by exploiting the relative positions of reads, and, second, to ensure that discovered events would be aligned across conditions since the motif of the TF is believed to remain the same across conditions unless data indicate evidence to the contrary.

For the first problem, the so-called Spatial Coupling algorithm was devised where neighboring reads which are sufficiently close together are encouraged to belong to the same event. For this case, we applied an inhomogeneous HMM-like model where the assignment of a read is dependent on the assignment of its neighboring read provided that are close enough to each other.

We compared the performance of this method to the one of a Simple Mixture model, where the independence assumption between all reads is taken into account. It was observed that the Simple Mixture model outperformed the Spatial Coupling one in terms of achieving better sparseness and locating more accurately the true events. In more detail, when both methods were tested on toy data, where the true events were known in advance, the Simple Mixture located successfully all the events (plus a False Positive one). On the other hand, the Spatial Coupling, performed poorly identifying also many more weak events other than

the true ones. In the case of real data, we compared only the most significant peaks of both methods, since the Spatial Coupling detected many more weak events than the Simple Mixture. It was observed that the peaks were called in very close positions for both methods and were very close to an Oct4 binding motif. However, both in toy and real data, the running time of the Simple Mixture model was orders of magnitude lower than the Spatial Coupling one.

Summarizing, the Spatial Coupling does not hold the sparseness features that the Simple Mixture model with a sparse prior does. This is maybe due to the fact that sparseness on the Simple Mixture model is enforced directly on all the components, while sparseness on an HMM is applied on the first data point and then is expected to be propagated on all the others through the transition probabilities. However, this, as shown, has lower performance than that of the Simple Mixture. In addition, if a false positive event receives some probability, then this will be propagated to neighboring positions through the dependence between the assignments of reads, which was a central assumption of this method. Lastly, the running time for the Simple Mixture is on the order of seconds, while that of the Spatial Coupling on the order of minutes, since at each iteration, the Forward–Backward algorithm is invoked for the evaluation of the E step, something which adds considerable complexity to the method, which is inexpedient for its performance results.

However, the assessment for the Temporal Coupling is different since the method achieves exactly its goal; that is, to ensure sparseness and alignment between corresponding events across conditions. As presented in the Temporal Coupling Section, all the data are considered first and trained on a Simple Mixture model with a sparse prior. This creates a set of candidate events, which will be used in the next step as the set where each condition would choose from based on the condition–specific data. That ensures two things; first, the events would be aligned together since they are the outcome of the consensus of all the data (of all conditions), second, the strength (weight) of each event will be determined based on the data for each condition.

For performance comparison, we tested the Temporal Coupling method against the Simple Mixture (by applying it to each condition separately) on both synthetic and real data. In the case of synthetic data, the Temporal Coupling method identifies the same number of events with the true ones. In addition, it locates them close enough to the true ones given that there is convolution between reads belonging to closely located events and some added noise on the data. On the contrary, the Simple Mixture model calls more spurious events and even those being in highly enriched locations are not aligned across conditions.
In more detail, we showed that the Temporal Coupling except for achieving alignment in general, it shows more robustness in terms of locating the predicted events at the same places across repetitions and identifying fewer FP events. In addition, when the true events are assumed aligned, the mean distance between predicted and true events is lower than that of the Simple Mixture, and even the sensitivity and specificity is higher since the consideration of all data across conditions in the case of Temporal Coupling assists in recovering even the weakest events, thus increasing sensitivity and the aim for alignment prevents from calling spurious events, thus increasing specificity. In the case where true events are not aligned, however, it does worse in terms of the mean distance between predicted and true events. However, it still provides better sensitivity. In the case of real data, Temporal Coupling enforces alignment of events as well, while the Simple Mixture fails to do so. The presence of a strong motif for these data could further establish the value of the algorithm.

Appendix A

Abbreviations

Table A.1: **Abbreviations.**

Abbreviation	Meaning
BN	Bayesian Network
ChIP	Chromatin Immunoprecipitation
ChIP-Chip	Chromatin Immunoprecipitation on Chip
ChIP-Seq	Chromatin Immunoprecipitation Sequencing
DBD	DNA Binding Domain
DBN	Dynamic Bayesian Network
EM	Expectation Maximization
ES	Embryonic Stem
FP	False Positive
HMM	Hidden Markov Model
PGM	Probabilistic Graphical Model
PSWM	Position Specific Weight Matrix
TAD	Trans-Activation Domain
SSD	Signal Sensing Domain
TF	Transcription Factor

Bibliography

[1] Marson A., Levine S., Cole M., Frampton G., Brambrink T., Johnstone S., Guenther M., Johnston W., Wernig M., Newman J., et al. Connecting microRNA Genes to the Core Transcriptional Regulatory Circuitry of Embryonic Stem Cells. *Cell*, 134(3):521–533, 2008.

[2] Valouev A., Johnson D.S., Sundquist A., Medina C., Anton E., Batzoglou S., Myers R.M., and Sidow A. Genome-wide analysis of transcription factor binding sites based on ChIP-Seq data. *Nature methods*, 5(9):829–834, September 2008.

[3] Fejes A.P., Robertson G., Bilenky M., Varhol R., Bainbridge M., and Jones S.J.M. FindPeaks 3. 1: a tool for identifying areas of enrichment from massively parallel short-read sequencing technology. *Bioinformatics*, 24(15):1729–1730, July 2008.

[4] Bishop C.M. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer New York, 2006.

[5] Nix D.A., Courdy S.J., and Boucher K.M. Empirical methods for controlling false positives and estimating confidence in ChIP-Seq peaks. *BMC Bioinformatics*, 9(1):523, December 2008.

[6] Reiss D.J., Facciotti M.T., and Baliga N.S. Model-based deconvolution of genome-wide DNA binding. *Bioinformatics*, 24(3):396–403, 2008.

[7] Johnson D.S., Mortazavi A., Myers R.M., and Wold B. Genome-Wide Mapping of in Vivo Protein-DNA Interactions. *Science*, 316(5830):1497–1502, June 2007.

[8] Mardis E.R. Chip-seq: welcome to the new frontier. *Nature Methods*, 4(8):613–614, August 2007.

[9] Robertson G., Hirst M., Bainbridge M., Bilenky M., Zhao Y., Zeng T., Euskirchen G., Bernier B., Varhol R., Delaney A., Thiessen N., Griffith O.L., He A., Marra M., Snyder M., and Jones S. Genome-wide profiles of STAT1 DNA association using chromatin immunoprecipitation and massively parallel sequencing. *Nat Meth*, 4(8):651–657, June 2007.

[10] Rozowsky J., Euskirchen G., Auerbach R.K., Zhang Z.D., Gibson T., Bjornson R., Carriero N., Snyder M., and Gerstein M.B. PeakSeq enables systematic scoring of ChIP-seq experiments relative to controls. January 2009.

[11] Bicego M., Cristani M., and Murino V. Sparseness Achievement in Hidden Markov Models. *International Conference on Image Analysis and Processing*, 0:67–72, 2007.

[12] Farnham P.J. Insights from genomic profiling of transcription factors. *Nat Rev Genet*, 10(9):605–616, September 2009.

[13] Kharchenko P.V., Tolstorukov M.Y., and Park P.J. Design and analysis of ChIP-seq experiments for DNA-binding proteins. *Nature Biotechnology*, 26(12):1351–1359, November 2008.

[14] Jothi R., Cuddapah S., Barski A., Cui K., and Zhao K. Genome-wide identification of in vivo protein-DNA binding sites from ChIP-Seq data. *Nucleic Acids Research*, pages 5221–5231, August 2008.

[15] Nielsen R., Pedersen T.Å., Hagenbeek D., Moulos P., Siersbæk R., Megens E., Denissov S., Børgesen M., Francoijs K.J., Mandrup S., et al. Genome-wide profiling of PPARγ: RXR and RNA polymerase II occupancy reveals temporal activation of distinct metabolic pathways and changes in RXR dimer composition during adipogenesis. *Genes & Development*, 22(21):2953–2967, September 2008.

[16] Feng W., Liu Y., Wu J., Nephew K., Huang T., and Li L. A Poisson mixture model to identify changes in RNA polymerase II binding quantity using high-throughput sequencing technology. *BMC Genomics*, 9(Suppl 2):S23, September 2008.

[17] Zhang Y., Liu T., Meyer C., Eeckhoute J., Johnson D., Bernstein B., Nussbaum C., Myers R., Brown M., Li W., and Liu X.S. Model-based Analysis of ChIP-Seq (MACS). *Genome Biology*, 9(9):R137, September 2008.