Alma Mater Studiorum – Università di Bologna
in cotutela con Università di Groningen

DOTTORATO DI RICERCA IN

Scienze Statistiche

Ciclo XXVIII

Settore Concorsuale di afferenza: 13/D1

Settore Scientifico disciplinare: SECS-S/01

STATISTICAL MODELLING OF SPATIO-TEMPORAL DEPENDENCIES IN NGS DATA

Presentata da: Saverio Ranciati

Coordinatore Dottorato

Alessandra Luati

Relatore

Cinzia Viroli

Relatore

Ernst Wit

Esame finale anno 2016

Contents

From where we stand the rain seems
random.
If we could stand somewhere else,
we would see the order in it.

TONY HILLERMAN

Chapter 1

Introduction

1.1 Statistics in biological systems

In the last decades we have witnessed an exponential growth of two big forces driving our knowledge enrichment: *computational efficiency* and *big data*. While the former encompasses a broader range of particulars - raw speed improvement, parallel computing, GPU processing - when compared to the latter, the sole term *big data* has rapidly capitalized all the attention of statistical community (among others), even if being able to precisely tell how big is 'big' - in some situations - has been proven to be quite difficult. It is worth noticing that neither of them, however, when purely approached without careful reasoning, can guarantee a clearer insight about complex systems that naturally describe interesting phenomena, such as those arising in biological contexts. Sure enough, when faced with a complex task, the first - and strongest - instinct has dangerously shifted towards throwing more and more data to the model, backed-up by both the vast amount of information available and the improved computational capacity of handling it. The first issue is that under the dome of *big data* most of the traditional - in the broadest meaning of the word - statistical techniques and methodologies lose their feasibility, both for practical and theoretical reasons. A perilous behavior threatening statisticians nowadays is to rely on complicated models to answer (simple, albeit not easy) questions about complex systems, calling on 'bigger' dataset for support on their assumptions and beliefs. In an attempt to balance out complexity, interpretability of the results and feasibility of a statistical analysis, a reasonable and careful step would be instead to start approaching the problem by modelling the structures underlying the data at our disposal. Effectively, knowledge may very well not be strictly equal to perfect description of a phenomenon but, instead, conceived as the understanding of the relationships between its fundamental components (be them genes, proteins, or generic statistical units per se). As an example, we may think about the complexity of some biological processes: gene-gene or gene-protein interactions, cellular signaling, circadian clock cycles, molecules dynamics and so forth. Especially in this fascinating contexts, along with - for example - *social networks*, we have been flooded with information: faster and cheaper technologies to biochemically analyze genetic data, labelled under the term *next-generation sequencing* (NGS), are introduced with increasing pace. The number of observations (i.e. the number of statistical units) and

'features' observable in a single experiment are skyrocketing. What kind of dependencies should we incorporate in our model, it being a simplified version of such a complex reality? A preliminary answer to this question, when thinking about the description of a process governed by stochasticity, is the spatio-temporal dimension. As natural as it may seem, however, without a solid *a priori* idea of the underlying process we might be forced to forego this 'quick and safe' structure given by time and space: that is, the strongest focus needed at the beginning is about the ever present trade-off between the idea of exploring data and looking for confirmatory results. If we want to better understand the spatial dynamics of interactions between genetic markers, for example, it might be sensible to encode a rich and flexible dependency structure in the model while keeping a simple level of description of the actual measurements comprising the dataset. On the other hand, if we are trying to analyze specific aspects of a process - again, for example, gene interactions - it is a powerful approach to encode in the model as much prior knowledge we have about the underlying dependency structure governing the units' relationships, while aiming for a richer description of the observable level of the model. The main flavor about this idea is to build a model in a hierarchical fashion, with a distinction between unobservable (latent) quantities and observable information, and to fit our idea of the spatio-temporal dimension where it truly belongs with respect to this two layers; the modularity of the aforementioned scheme gives also rise to another encouraging aspect: to seek extensions of the model proposed, updating step-by-step the levels of the hierarchy. Out of the possible paradigms to follow, we will embrace the Bayesian one, mostly for two reasons: the built-in quantification of uncertainty of the parameters in the model and the more straightforward possibility to tackle a complex model. We will also switch from the epitome of *big data* to *high-dimensional data*, which bears a more focused phrasing on what could be the associated characteristics of this kind of data (curse of dimensionality, sparsity, number of features higher than available samples and so forth).

1.2 Thesis outline

We will outline a path to follow while exploring the aforementioned challenges, with a focus on two main tasks: clustering genes and retrieving information about unobservable quantities. While introducing the spatio-temporal structure, we will move from considering both time and space dependencies, but in a simple fashion, to model them separately yet in a richer way.

The PhD project is structured in three parts. In the next chapter we will present a paper that employs a Markov Random Field (that is, a graphical model) as a latent structure to describe the spatial relationships between locations of the genome and the temporal dependency of experimental replicates of the very same strand of DNA; faced with the task to understand if a location is configured as enriched or not by a protein, we will use a mixture of two discrete distribution to describe the observed counts of a particular genetic measurement, while simultaneously dealing with specific issues intrinsic to the NGS context.

In the third chapter, a non-conventional model-based clustering scheme will be presented, that allows for a unit (a gene, in our case) to belong to more than one cluster at a time. In allocating these units, however, we will take into account the

natural neighboring spatial dependency between them and we will model it so that it plays a role into the in the way we cluster our observations through the weights of the chosen mixture of discrete distributions.

In the forth chapter, we will propose a Bayesian hierarchical model for Ordinary Differential Equations (ODE) describing the temporal dynamics of a continuous process: while searching for a flexible description of the data through the use of penalized spline regression, we will indirectly solve the ODE and quantify the uncertainty about the solution obtained with respect to the noisy observations we have at our disposal.

1.3 Chapter 2: Mixtures and graphs

In the second chapter of this thesis, we first introduce and model data coming from chromatin immunoprecipitation and sequencing (ChIP-Seq) experiments. Whereas years ago microarrays were considered 'gold standards' for genetic related analysis, Next-Generation Sequencing has now become the prominent broad class of biological techniques employed to study the complex relationships between the DNA, RNA, proteins and cell functionality. In this context, with ChIP-Seq experiments, we are interested in discovering if a protein of interest is binding or not to the RNA and to which portion of it. More specifically, if a region has an associated 'high' count in the data it is more likely the protein has enriched (i.e., bound to) that region. If we know the aforementioned protein to be associated with a particular disease and we find out which portions of the DNA of a cell, from an affected tissue, the protein is enriching we may infer that those regions - and, more in depth, the individual genes - are associated with the disease itself. From a statistical point of view, a not to be overlooked feature that sets ChIP-Seq (and in general NGS experiments) apart from microarray data is that they yield discrete observations instead of continuous measurements. Other peculiarities are: *overdispersion*, that is an amount of variability greater than expected when a basic model (like the Poisson distribution) for the counts is assumed; even though usually raw data are pre-processed, the units studied relate to region of the DNA that are contiguous to each other and thus a need for a spatial pattern to be included in the model arises; zero-inflation, that is an abundance of zeros as observed values. Also, as efficiency progresses and costs decrease, more biological or technical replicates are made available to the researchers from the laboratory, demanding for statistical methodologies able to consider jointly all the information at their disposal. Last but not least, the measurements are collected at different time points throughout the whole experiment, giving the opportunity to look for a temporal dependency in the data. To face these challenges, we propose a hierarchical mixture model that fuses together two distinct layers: a latent structure, devoted to infer the unobserved protein binding process we are interested in, built by considering both the spatial and temporal dependencies that connect the regions of the DNA analyzed in the experiment; and secondly, a measurement model, characterized by means of a mixture of discrete distributions that can accommodate for overdispersion and jointly model all the technical or biological replicates. For the hidden layer, we choose a Markov Random Field, to reflect the assumption of first order spatio-temporal dependency: that is, each region (unit) depends only on its left and right contiguous neighbors - in a spatial sense - and previous or following

neighboring time point - in a temporal sense. The latent structure can be represented through a graphical model and thus all the related theory can be used to translate it into probabilities. As for the mixture model, we consider Negative Binomial distributions for their flexibility and innate description of overdispersion. Following a Bayesian approach and all the standard derivations needed to implement an MCMC procedure, we first assess the performance of our proposed method in a simulation environment and then we move to the analysis of differential roles in gene regulation of two transcription factors, p300 and CBP by means of ChIP-Seq data.

1.4 Chapter 3: Spatial dependency and beyond

In the third chapter of the thesis, we look at the same experimental framework, ChIP-Seq data, but from a different perspective. We are now interested in classifying the units (genes, regions of the chromosome, etc.), based on the observed counts, into groups (*clusters*) that have a meaningful biological interpretation. What we have in mind about "meaningful biological interpretation" is mostly related to a simple paradigm: cell functionalities (for example mitosis, self-destruction, stress response) are regulated with complex patterns of signaling by genes, proteins and other important biological 'actors'. It is not unlikely that a gene, a protein, a biological 'actor', participate in more than one cell functionality. If we now translate "gene, protein, actor" and "cell functionality" into a more statistical language as "units, observations" and "clusters" we can then interpret the original problem from a more methodological point of view. Model-based clustering is a widely used technique that has found application in a broad class of problems in many different contexts. However, in his simplest description, the framework assumes the groups are mutually exclusive and this is a restriction that does not fit the motivating research question. In the literature, some authors proposed extension to overcome this limitation; having in mind the same challenges and peculiarity of the data that we already described in Section 1.3, we propose a model that can accommodate at best all of them: allocation of units into - potentially - more than one group; discrete measurements; overdispersion; clusters that are interpretable as "meaningfully" linked to each other (where, again, meaningful is a term recalling the cell functionality paradigm). We pursue (as before) the solution of the task by employing a hierarchical structure. We use a mixture of Negative Binomials to flexibly model the observations and a latent allocation structure that classifies each unit into groups. What we define as groups can either be: 'primary clusters', which represent the principal functionalities or prototypical functions that we have in mind in our research question (for example, cluster of enriched regions and cluster of non-enriched regions); 'multiple allocation clusters', that are collection of units that belong simultaneously to more than one of the primary clusters. The main idea is to move from the original representation of a mixture model for clustering with, let's say, $k = 3$ groups to a mixture that has $2^k = 8$ groups, where $2^k = 8$ is the number of all possible configurations of allocation we could have for one unit. In doing so, we create multiple allocation components (clusters) having parameters that are completely specified from the primary clusters originating them. In this case, thus, we are actually not introducing in the model new parameters that have to be estimated. As a by-product of this approach, we obtain an 'outward' cluster: a group in which outliers could be allocated or, as we do, ob-

servations that would otherwise be account for with a zero-inflation adjustment of the model. A further step that we take is introducing the spatial dependency among the units we are studying. We do so by allowing the weights of the mixture, which reflect the prior probability to be allocated to a cluster, to vary according to a spatial pattern that we assume to be a conditional autoregressive model. More specifically, this pattern takes into account the position and the relative distances between the units and encodes them as an additional latent layer into our hierarchical structure. To study the performance of the proposed model we test it in a simulation environment and then we proceed to inspect the same dataset as in Chapter 2 from this alternative point of view.

1.5 Chapter 4: Time dynamics and complex systems

In the fourth chapter, we focus on the dynamics of biological systems that evolve through time. In many fields of application, such as engineering and the study of dynamical systems in biology, chemistry and physics, researchers often describe the behavior of complex systems with a set of equations called ordinary differential equations (ODE's). These mathematical objects are equations that attempt to model the changes of the state of the system with respect to time by considering a set of parameters, unknown quantities governing the law of the process itself. There is a dual aspect to be considered when looking at the system of ODE's: the known relationships are expressed at the derivative level of the components of the process but what we actually observe are the states at fixed time points, not the evolution in time. Even if the functionals in the equations are known, expressing the dependency among the components, we still need to estimate the parameters in order to fully describe the dynamics. If an analytical solution of the system is available, those parameters can be directly recovered from the observed data; if such a solution is not readily available in closed form, as it is often the case, numerical integration is needed. In many contexts it is also likely that the data we collect are affected by noise, perturbing the real underlying dynamic. Another way to look at the task at hand, a perspective more deeply connected with the statistical methodology, has been followed in the literature in the last decade: it is possible to avoid direct integration of the equations by *smoothing* the data and many methods have been proposed, either within a frequentist and Bayesian framework. We follow this aforementioned idea by proposing a two-step Bayesian Smooth-and-Match strategy: at a first stage, we smooth the observed data in order to reconstruct a noiseless sequence of the states of the system for all its components; at a second stage, we use this states as inputs for the known functionals that appear in the ODE's and we 'match' this temporary description of the dynamic with the data we have. We thus move the focus from solving the system to directly infer the parameters that describe the process, obtaining the solution as a by-product of all the procedure. Delving briefly into some technicalities: we adopt penalized Bayesian smoothing with cubic splines as our first step of the procedure; as for the second step, we obtain the parameters of the system through a penalized (ridge) regression approach. The two compartments of the strategy are finally connected by assuming a common noise term that acts as a built-in quantification of the uncertainty associated to the solution of the system that we have (again, indirectly) reconstructed within the MCMC sampling scheme. In order

to assess the performance of the proposed method we set up three different simulation studies and we then proceed to compare the results we obtain on another dataset previously analyzed by other authors.

Chapter 2

Spatio-temporal model for multiple ChIP-Seq experiments

Saverio Ranciati[1,2], Cinzia Viroli[2], Ernst C. Wit[1]

[1] *University of Groningen, Johann Bernoulli Institute of Mathematics and Computer Science, Groningen, the Netherlands.*
[2] *Department of Statistics, University of Bologna, Bologna, Italy*

[I]Published in "Statistical Applications in Genetics and Molecular Biology. Volume 14, Issue 2, Pages 211–219, ISSN (Online) 1544-6115, ISSN (Print) 2194-6302, DOI: 10.1515/sagmb-2014-0074, February 2015"

Abstract

The increasing availability of ChIP-Seq data demands for advanced statistical tools to analyze the results of such experiments. The inherent features of high-throughput sequencing output call for a modelling framework that can account for the spatial dependency between neighboring regions of the genome and the temporal dimension that arises from observing the protein binding process at progressing time points; also, multiple biological/technical replicates of the experiment are usually produced and methods to jointly account for them are needed. Furthermore, the antibodies used in the experiment lead to potentially different immunoprecipitation efficiencies, which can affect the capability of distinguishing between the true signal in the data and the background noise. The statistical procedure proposed consist of a discrete mixture model with an underlying latent Markov Random Field: the novelty of the model is to allow both spatial and temporal dependency to play a role in determining the latent state of genomic regions involved in the protein binding process, while combining all the information of the replicates available instead of treating them separately. It is also possible to take into account the different antibodies used, in order to obtain better insights of the process and exploit all the biological information available.

Keywords: ChIP-seq, mixture distributions, Markov Random Field model, MCMC.

2.1 Introduction

In the context of genetic analysis, data from a biological technique known as Chromatin ImmunoPrecipitation-sequencing (*ChIP-Seq*) are becoming increasingly more frequent. These experiments arise from the combined process of Chromatin ImmunoPrecipitation (ChIP) and massive parallel DNA sequencing (Seq), providing as a final result the number of tags (reads) or fragments of DNA aligned to each region of the strand of genome inspected ([6]). This technique is employed to provide insight about DNA methylation, chromatin and histone modifications and the interactions between proteins and DNA: in particular, certain proteins are known to be participating in various biological processes and thus being able to detect which regions are 'activated' by the former can explain better the role of the latter into the process itself. The aim of the statistical analysis is thus to distinguish between enriched regions (bound by the protein) from those who are not. There are inherent features of the problem outlined that have to be taken into consideration. The chromatin modifiers and transcription factors involved in the experiment interact with broad regions of the DNA: this highlights the spatial dependency component that characterizes the phenomenon and also means that the usual peak-detection algorithms will not be able to retrieve correctly the enriched state of the locations. The ChIP phase of the experiment is performed with specific antibodies that carry different efficiencies, where the concept of efficiency is defined as the capability of interacting only with the protein analyzed and providing the bits of DNA effectively bound by the transcription factor or chromatin modifier. Thus, the amount of background noise and quality of the signal in the data are affected by the ImmunoPrecipitation (IP) efficiencies of these antibodies. The spatial dependency component in the observations is further enhanced due to a pre-processing of the data: the raw results of the ChIP-seq are summarized in fixed-width windows and the original counts for the tags at a higher resolution are summed over into smaller regions that may contain more than one gene. In this experimental setting, it is possible to have observations from two or more different time points (e.g. immediately after the protein-DNA interaction and after 30 minutes): this naturally introduces a temporal dependency structure that needs to be modeled in order to correctly perform statistical inference about the enrichment behavior of the process analyzed. Multiple technological or biological replicates are also available, demanding for a tool that can take into account jointly all the information available instead of relying on separate analysis for each instance. Given the features of the problem, it is a common approach to model the data using a mixture of discrete distributions. There are already a number of methods that adopted this procedure, using different parametric densities for the mixture components: [8], [11]. They employ Generalized Poisson or Negative Binomial densities ([4])to model the data, without though including spatial and temporal dependency. A few exceptions to this common approach are proposed by [1] and [12]: in this works, the spatial structure feature has been considered, as it has been already done for other ChIP experiments in the literature, by using a Hidden Markov Model framework ([1], [10], [11]). Most of them however do not take into account the joint information of multiple biological or technical replicates: instead, they just analyze each experiment and retain only the

common regions found to be enriched: they exploit a control sample in order to compare the separate results obtained from the replicates available and they asses a differential enriching behavior between them (see [2]). There is, thus, a lack of a statistical approach to jointly use the information from all the replicates with the aim to obtain a more robust inference procedure. The model proposed by [1] is able to tackle all the aforementioned issues, except for the temporal aspect of the problem: this model allows to take into account both the spatial dependency and the different IP efficiencies belonging to each individual replicate and/or antibody used in the experiment. One missing aspect remains the temporal dependency structure arising when the experiment is performed at multiple time points. In this work we propose a more general version of the model showed in [1], in which the novelty is the introduction of the temporal dependency directly into the latent structure that characterizes the underlying biological process of ChIP-seq experiments. In order to do so, we employ a Markov Random Field ([7]) to describe the enrichment behavior and a flexible mixture model as a measurement model to account for the specific features and IP efficiencies of the antibodies used while jointly considering all the technical and biological replicates available. In Section 2, the model is described and the statistical aspects are discussed while describing the main quantities used in the implementation; in Section 3, results from a simulation study are showed to asses the performances of our model with respect to other existing algorithms; in Section 4, we summarize the output after applying our model to a real dataset; in Section 5, we discuss the features of the method proposed and the further developments.

2.2 Model and Methods

We propose a more general version of the model used in [1] by extending the latent variables structure, allowing it to take into account also a temporal dependency between the same region at different time points. Let Y_{mtr} be the number of reads (*tags*) for the m-th bin $(m = 1, \dots, M)$, at time point t $(t = 1, \dots, T)$ for the replicate r $(r = 1, \dots, R_t)$. The subscript r includes also the specification of the antibody or the conditions/treatments that share the same latent structure (e.g. $R_1 = 2$, for two replicates with the same antibody at the first time point; $R_2 = 6$, for two replicates for each of three different antibodies used at the second time point). We consider a joint mixture distribution as follows:

$$Y_{mtr} \sim p_t f^S(y|\theta^S_{tr}) + (1 - p_t) f^B(y|\theta^B_{tr}) \tag{2.1}$$

where $p_t = P(X_{mt} = 1)$ is the probability of region m to be enriched at time point t, with respect to a latent binary variable X_{mt} that represent the underlying biological process of protein binding. The vectors θ^B_{tr} and θ^S_{tr} contain the mean and dispersion parameters corresponding to the specific background or signal component of the mixture. We use two discrete densities f^B and f^S respectively to model the background and the signal components of the mixture in (Eq.2.1): for the background component f^B we consider a Zero-Inflated Poisson (ZIP) or a Zero-Inflated Negative Binomial (ZINB), in order to account for both the presence of overdispersion in the data and abundance of zeros, features that are quite common in this context; as for the signal component f^S, we employ a Poisson (P) or Negative Binomial (NB) distribution. A further latent variable Z is introduced to represent the

zero-inflated density f^B as a mixture itself of a zero-mass distribution and a discrete density as the ones previously described. In the case of a ZINB, we have the following representation:

$$f^B(y_{mtr}|\pi_{tr},\mu_{tr},\phi_{tr}) = \begin{cases} (1-\pi_{tr}) + \pi_{tr}\left(\frac{\phi_{tr}}{\phi_{tr}+\mu_{tr}}\right)^{\phi_{tr}} & \\ & \text{if } y_{mtr}=0 \\ \pi_{tr}\frac{\Gamma(y_{mtr}+\phi_{tr})}{\Gamma(\phi_{tr})\Gamma(y_{mtr}+1)}\left(\frac{\mu_{tr}}{\phi_{tr}+\mu_{tr}}\right)^{y_{mtr}}\left(\frac{\phi_{tr}}{\phi_{tr}+\mu_{tr}}\right)^{\phi_{tr}} & \\ & \text{if } y_{mtr}>0 \end{cases}$$

with μ^B_{tr} being the mean parameters and ϕ^B_{tr} the dispersion parameters of the Negative Binomial distribution. The parameters $\pi_{tr} = P(X_{mt}=0, Z_{mtr}=1)$ and $(1-\pi_{tr}) = P(X_{mt}=0, Z_{mtr}=0)$ are the weights of the mixture representing the zero-inflated distribution: the latter is the proportion of the background which is due to an abundance of zeros, while the former is related to the raw noise in the data. The conditional distribution of $\mathbf{Y}$, given both the latent binary variables $\mathbf{X}$ and $\mathbf{Z}$, is then the following:

$$\begin{aligned} Y_{mtr}|\, X_{mt}=0, Z_{mtr}=0 &\sim \mathbb{1}(y=0) \\ Y_{mtr}|\, X_{mt}=0, Z_{mtr}=1 &\sim Poisson(\lambda^B_{tr}) \quad or \quad NB(\mu^B_{tr},\phi^B_{tr}) \\ Y_{mt}|\, X_{mt}=1 &\sim Poisson(\lambda^S_{tr}) \quad or \quad NB(\mu^S_{tr},\phi^S_{tr}) \end{aligned}$$

The latent structure $\{X_{mt}\}$, with $X_{mt}=1$ for enriched region m at time point t and $X_{mt}=0$ otherwise, can be represented as an undirected graph with nodes corresponding to each bin at a specific time point or also as a lattice with a generic site (m,t) (Fig.2.1). The edges of the graph connect each X_{mt} only to adjacent regions in the genome and one-step backward/forward time points and they also are the *cliques* used to factorize the whole graph ([5]). Each node represent a *separator* of these cliques, connecting two or more conditional relationships between neighboring variables.

These assumptions can be jointly expressed as a first-order markovian property as follows:

$$P(X_{mt}=s|X_{(-m),(-t)}) = P(X_{mt}=s|N(X_{mt})) \tag{2.2}$$

with $s=\{0,1\}$ and $N(X_{mt}) = \{X_{m-1,t}, X_{m+1,t}, X_{m,t-1}, X_{m,t+1}\}$ being the neighborhood of the generic site (or node) (m,t), where $X_{(-m),(-t)}$ is the set of nodes except for the (m,t) one. This kind of latent structure is also called a Markov Random Field. The joint probability of the latent structure, given the assumption in (Eq. 2.2), can be factorized into:

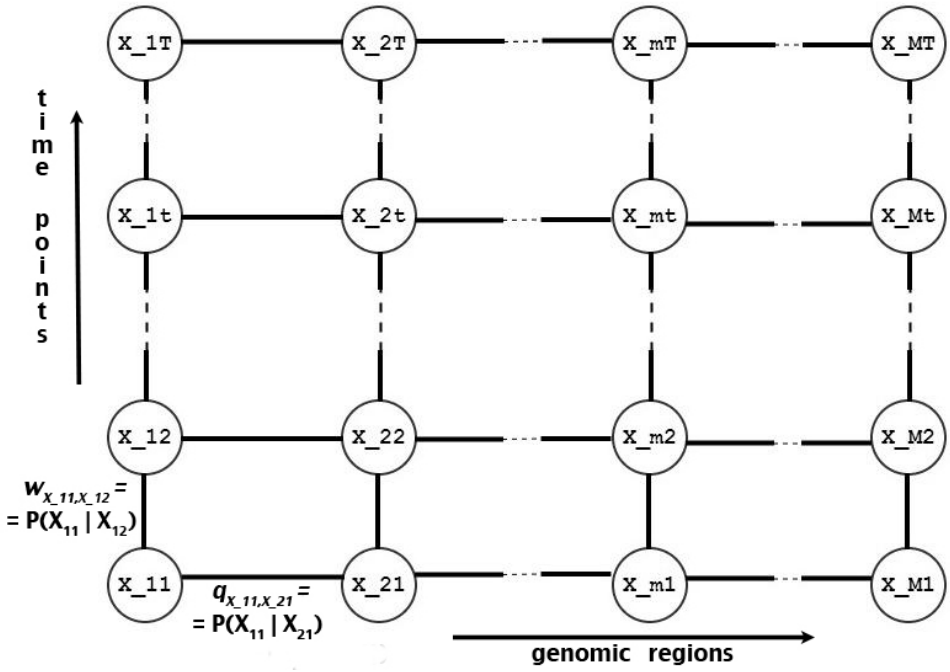

Figure 2.1: The undirected graph representing the latent structure $\mathbf{X}$ of the model.

$$P(\mathbf{X}) = P(X_{11}, X_{21}, \ldots, X_{MT}) = P(X_{MT}) \prod_{m=2}^{M} \prod_{t=2}^{T} \frac{1}{P(X_{mt})} \times$$
$$\times \prod_{m=1}^{M-1} \prod_{t=1}^{T-1} P(X_{m+1,t}|X_{mt})\, P(X_{m+1,t+1}|X_{m+1,t})$$
$$\times \prod_{m=1}^{M-1} P(X_{mT}|X_{m+1,T}) \prod_{t=1}^{T-1} P(X_{1t}|X_{1,t+1}) \tag{2.3}$$

We introduce some notation to rewrite the conditional dependencies in (Eq. 2.3):

$$\delta_{j,k} = P(X_{mt} = j, X_{m+1,t} = k) \qquad \tau_{l,h} = P(X_{mt} = l, X_{m,t+1} = h)$$
$$n_{j,k} = \#\{X_{mt} = j, X_{m+1,t} = k\} \qquad u_{l,h} = \#\{X_{mt} = l, X_{m,t+1} = h\}$$

$$q_{j,k} = \frac{\delta_{j,k}}{\delta_j} \qquad w_{j,k} = \frac{\tau_{l,h}}{\tau_l}$$
$$q_0 = q_{0,1} \quad q_1 = q_{1,1} \quad w_0 = w_{0,1} \quad w_1 = w_{1,1}$$

and $j, k, l, h \,\epsilon\, \{0, 1\}$.
We assume that:

$$P\left(X_{mt}=1\right)=\frac{q_0}{1-q_1+q_0}=\frac{w_0}{1-w_1+w_0}$$
$$P\left(X_{mt}=0\right)=\frac{1-q_1}{1-q_1+q_0}=\frac{1-w_1}{1-w_1+w_0}$$

and:

$$q_{0,1}=q_{1,0}\quad w_{0,1}=w_{1,0}$$
$$q_0=1-q_1\quad w_0=1-w_1$$

We do so in order to parametrize the factorization of the latent structure $\{X_{mt}\}$ in (Eq. 2.3) without employing too many parameters; we assume that the spatio-temporal first-order markov dependency is the same for all the time points and regions considered: this means that the parameters q_1 and w_1 do not have any subscript and they are the same across the whole graph. More precisely, we want to capture the spatial dependency with the parameter q_1 while, conversely, retain the temporal information in the data through w_1. These two quantities represent the probability of a region remaining in the same state (bound or not by the protein) while moving respectively along the genome or from a time point to next/previous one. Their one's complements $1-q_1$ and $1-w_1$ measure the probability of the latent state to switch from 0 (not enriched) to 1 (enriched) and viceversa when moving from a node in the graph to its neighbors. In the following, we will only show the derivations for the Negative Binomial distribution used in modelling both the background and the signal component. The joint complete likelihood for this model, considering a ZINB for background and NB for signal, is given by:

$$\begin{aligned}
&P(\mathbf{X},\mathbf{Z},\mathbf{Y}|\mathbf{\Theta})=P(\mathbf{X}|\Theta)P(\mathbf{Z}|\mathbf{X}=0,\Theta)P(\mathbf{Y}|\mathbf{X},\mathbf{Z},\mathbf{\Theta}) \qquad (2.4)\\
&\propto q_1^{n_{1,1}+n_{0,0}}\left(1-q_1\right)^{n_{1,0}+n_{0,1}}w_1^{u_{1,1}+u_{0,0}}\left(1-w_1\right)^{u_{1,0}+u_{0,1}}\\
&\times\prod_{t=1}^{T}\prod_{r=1}^{R}\pi_{tr}^{\sum_{m=1}^{M}\mathbb{1}(X_{mt}=0,\,Z_{mtr}=1)}(1-\pi_{tr})^{\sum_{m=1}^{M}\mathbb{1}(X_{mt}=0,\,Z_{mtr}=0)}\\
&\times\prod_{t,r,m}^{T,R,M}\left[\frac{\Gamma(y_{mtr}+\phi_{tr}^{B})}{\Gamma(\phi_{tr}^{B})\Gamma(y_{mtr}+1)}\left(\frac{\mu_{tr}^{B}}{\phi_{tr}^{B}+\mu_{tr}^{B}}\right)^{y_{mtr}}\left(\frac{\phi_{tr}^{B}}{\phi_{tr}^{B}+\mu_{tr}^{B}}\right)^{\phi_{tr}^{B}}\right]^{\mathbb{1}(X_{mt}=0,\,Z_{mtr}=1)}\\
&\times\prod_{t,r,m}^{T,R,M}\left[\frac{\Gamma(y_{mtr}+\phi_{tr}^{S})}{\Gamma(\phi_{tr}^{S})\Gamma(y_{mtr}+1)}\left(\frac{\mu_{tr}^{S}}{\phi_{tr}^{S}+\mu_{tr}^{S}}\right)^{y_{mtr}}\left(\frac{\phi_{tr}^{S}}{\phi_{tr}^{S}+\mu_{tr}^{S}}\right)^{\phi_{tr}^{S}}\right]^{\mathbb{1}(X_{mt}=1)}
\end{aligned}$$

where the first term $P(\mathbf{X}|\Theta)$, representing the latent structure, is shared among the different replicates that are jointly considered in the model proposed. The features of each experiment are captured in the measurement model, which comprise of as many mean and dispersion parameters as antibodies used and it is characterized by $P(\mathbf{Z}|\mathbf{X}=0,\Theta)$ and $P(\mathbf{Y}|\mathbf{X},\mathbf{Z},\mathbf{\Theta})$. Inference is performed in a Bayesian framework: after choosing priors, the posterior distributions for all the parameters are

derived along with the sampling schemes for the latent variables $\mathbf{X}$ and $\mathbf{Z}$. When the full conditional of the parameter is a non-conjugate distribution, a Metropolis-Hastings sampling procedure is used; otherwise, conjugate prior distributions and Gibbs samplings are involved. First, the latent variable X_{mt} is sampled from its full conditional distribution, according to the position of the node on the graph, the probability of the corresponding generic node (m, t) having state s being:

$$\begin{aligned} P(X_{mt} = s|\dots) &\propto q_{X_{m-1,t},s}\ q_{s,X_{m+1,t}}\ w_{X_{m,t-1},s}\ w_{X_{m,t+1},s} \times \\ &\times \prod_{r=1}^{R_t} P(Y_{mtr}|X_{mt} = s) \end{aligned}$$

Similar derivations are obtained for nodes that lie on the borders and corners of the undirected graph (see Appendix). Given the value of latent state $X_{mt} = 0$, the latent variable Z_{mtr} is then sampled with a Gibbs method from its full conditional distribution:

$$\begin{aligned} P(Z_{mtr} = s|X_{mt} = 0, \dots) &\propto P(y_{mtr}|X_{mt} = 0,\ Z_{mtr} = s, \Theta) \times \\ &\times P(Z_{mtr} = s|X_{mt} = 0) \end{aligned}$$

The inflation parameter π_{tr} is sampled from its posterior Beta distribution

$$\begin{aligned} \pi_{tr} &\sim \text{Beta}\,(A, B) \\ A &= A_{\pi_{tr}} + \sum_{m=1}^{M} \mathbb{1}(X_{mt} = 0, Z_{mtr} = 1) \\ B &= B_{\pi_{tr}} + \sum_{m=1}^{M} \mathbb{1}(X_{mt} = 0, Z_{mtr} = 0) \end{aligned}$$

and it represents the proportion of the background density component, f_B, which do not consist of the zero mass (inflation) distribution. The posterior densities for the transition probabilities are:

$$\begin{aligned} q_1 &\sim Beta(A_{q_1} + n_{1,1} + n_{0,0}, B_{q_1} + n_{1,0} + n_{0,1}) \\ w_1 &\sim Beta(A_{w_1} + u_{1,1} + u_{0,0}, B_{w_1} + u_{1,0} + u_{0,1}) \end{aligned}$$

We select flat prior distributions by setting all the hyperparameters of the Beta distributions equal to unity. The mean and overdispersion parameters of the background and signal distributions are estimated through a Metropolis-Hastings procedure. A Truncated Normal Distribution is used as a proposal to generate new values: this choice ensures that candidate values are always positive. The ratio of the normalizing constants has to be considered into the formula for the acceptance ratios

(see Appendix for further derivations).

2.3 Simulation study

We assess the performance of the model proposed (*stMRF*) and compare it to other existing methods (*iSeq* [2] R package: `iSeq`, *MRF* [1] R package: `enRich`). The scenarios used arise from the combination of many characteristics in the simulated data, such as the proportion of zero-inflation in the background component (π), the propensity to binding (low or high values of the transition probabilities), the difference in the mean level of the background and signal (low, high). We simulated four and two time points ($T = 4$) and two thousand regions ($M = 2000$) and three replicates for each temporal instance ($R = 3$); 5000 MCMC iterations are performed, with a 2500 burn-in window. All the details are available in the Appendix. Given that *MRF* and *iSeq* do not allow for more than one time point to be used, the two models are evaluated for each of the four simulated instances. The latter (*iSeq*), also, does not account for replicates and only the best result is showed for it (chosen between the three replicates available). The mean and overdispersion parameters of the Negative Binomial (μ, ϕ), along with the proportion of inflation (π), are well estimated by *MRF* and *stMRF*, without any pattern associated to the propensity of binding or degree of distance between background and signal components used in the scenarios. *iSeq* only allows for a Poisson distribution as background and signal and is unable to retrieve the true value of the mean parameters without introducing some bias. Between *MRF* and *stMRF*, the latter shows the lowest posterior standard deviations in all the scenarios. The transition probabilities are comparable only between *MRF* and *stMRF*, because *iSeq* do not account for the spatial dependency in the same way as the other two and employs a different parametrization. Both *MRF* and *stMRF* show a posterior mean distributions for q_1 close to the true values used in the simulation process and our model is also able to retrieve the parameter w_1 with a very low posterior standard deviation. The advantage of taking into account the temporal dependency in the model *stMRF* is clear when comparing the observed *False Non Discovery Rate* (FNDR) , which is the fraction of regions that were enriched but classified as not enriched, at a fixed estimated *False Discovery Rate* (FDR)([3])(see Table 2.1). In order to determine if a region m is enriched or not we put a cut-off value on the posterior probability $\hat{P}(X_{mt} = 1)$ while controlling for the FDR to be 5%. If it is not possible to find such a cut-off point, we use a naive approach setting 0.5 as the threshold.

The *stMRF* model outperforms, in term of observed FNDR, the other two in every scenario simulated (three of which are presented in Table 2.1). *iSeq* cannot take into account all the replicates simultaneously, nor the temporal dependency, and thus has poor performances; *MRF* is able to perform as good as *stMRF* only at some instances of specific scenarios: this is due to the capability of the former to jointly use the replicates available. However, given that for *MRF* all the time points are treated as separate conditions/experiments with their own transition probabilities and it cannot model the temporal structure of the simulated data, it has generally higher values of observed FNDR.

Table 2.1: Observed FNDR (in percentage) at a fixed 5% estimated FDR for the three models at different scenarios, for all the time points.

Scenario	t	*iSeq*	*MRF*	*stMRF*
(12b)	1	35.71	3.87	1.68
$\pi = 0.9$	2	35.80	6.56	2.87
Binding Propensity: *High*	3	39.24	19.62	3.80
Ratio Signal/Background: *Low*	4	33.03	1.73	1.12
(5b)	1	26.00	0.10	0.00
$\pi = 0.5$	2	27.57	0.30	0.10
Binding Propensity: *Low*	3	28.72	0.81	0.40
Ratio Signal/Background: *High*	4	23.28	0.00	0.00
(8b)	1	29.80	0.08	0.00
$\pi = 0.5$	2	40.50	0.00	0.00
Binding Propensity: *High*	3	44.75	3.83	0.00
Ratio Signal/Background: *High*	4	42.71	0.00	0.00

2.4 Genome-wide assessment of differential roles for p300 and CBP in transcription regulation

Transcription coactivators p300 and CBP are known to partecipate in the regulation of genes responsible for many roles, especially in embryogenesis. Most of the genes related to the process regulated by these two TFs are bound by both, but [9] showed that some of them may be preferentially bound to one or the other, following a different binding pattern. In particular, CBP has been found to regulate more genes that are involved in the negative transcription. Mostly, p300 and CBP have been showed to take a prominent role in important biological process of the cell such as proliferation, differentiation, and DNA repair mechanism. Also, some studies suggested that p300 and CBP may be involved in the development of cancer or other diseases ([9]). The binding sites for the two transcription coactivators appear to be altered when the cell is stimulated or not, even if they retain an overlapping number of regions bound by both, and thus they are analyzed at different conditions and time points. We use our model to analyze the example dataset available in the package `enRich` ([1]) which contains data for p300 and CBP ChIP-seq experiments. We select data only for the protein p300 at time "zero" as our first time point and the two replicates obtained after 30 minutes as our second time point. The bins are already summarized from the processed data using a 1000bp window. The dataset contains 33916 observations, each one corresponding to a region on the same chromosome 21. We run the MCMC algorithm with 10000 iterations and a burn-in window of 5000, using a Negative Binomial distribution for both the background and signal component.

The posterior mean of π is close to one in both the time points analyzed: this means that a Negative Binomial distribution for the background could be potentially enough to capture the overdispersion in the data, without relying on a zero-inflated

Table 2.2: Number of enriched regions and estimated False Discovery Rate for *MRF* and *stMRF* (*: cut-off on the posterior probability of $X = 1$ equal to 0.5; $\star$: cut-off on the posterior probability of $X = 1$ equal to 0.24).

Model	N. of Enriched Regions	$\hat{FDR}$
MRF	3089	5 %
(*) stMRF	3160	5 %
($\star$) stMRF	3024	2.4 %

version of the model. The posterior standard deviations for the parameters of the measurement model (μ, ϕ) and the transition probabilities (q_1, w_1) are small. The posterior means for q_1 and w_1 are 0.96 and 0.99, the latter meaning that if a bin is enriched at $t = 1$ it will very likely be enriched also at the next time point. There seems to be a strong dependency, thus, in both dimensions of the binding process (spatially and temporally). To decide if a bin m is enriched, we set a 0.5 cut-off to the posterior probabilities obtaining an estimated FDR and the number of enriched regions found by the algorithm. For both time points, the estimated FDR of *stMRF* is equal to 0.024 and the number of regions detected as bound by the protein is 3024 out of the total 33196 observations. The number of regions bound by the protein identified by *MRF* is 3098, controlling for a fixed estimated FDR of 5%: if the same criteria is used for *stMRF*, the number of regions found as enriched is 3160 in both time points, which is slightly higher than the previous result (see Table 2.2). We show some of the bins analyzed (see Figure 2.2): the black dots are the observed counts of the dataset, while the red lines and squared dots represent the latent state of the corresponding bin estimated by the algorithm (model *stMRF*). To regions found to be enriched by our model correspond red dots aligned at the middle of the plot while, conversely, bins which are not bound by the protein (according to the algorithm) have red dots lying on the X axis. As in Figure 2.2, a genomic location could be labelled as not enriched by the algorithm even when presenting a count of tags as high as (spatially) neighboring regions: this is related to a correction effect that the temporal dependency structure can induce on each time point, allowing the detection of enriched locations while avoiding spurious binding that may occur due to the antibody used or other noise in the process. This can help understanding, through a validation process, the dynamics of regulation and activation of genes known to be related to this transcription factor ([9]), while also pointing out new locations that could need a more detailed and specific re-sequencing after the experiments are performed and analyzed.

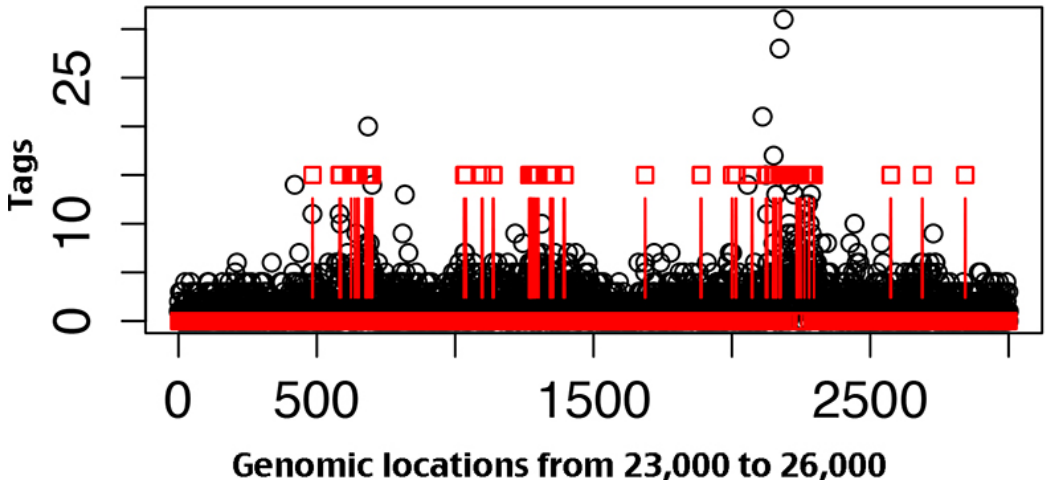

Figure 2.2: Plot of observed counts for the first time point along the Chromosome 21. Estimated latent states of the bins are overplotted.

2.5 Conclusion and discussion

We have proposed a model that extended the one introduced by [1], incorporating in a parsimonious way a new dependency structure that spans through the time dimension, enabling it to account for the temporal aspect of the binding process. This has been done in order to exploit the information available at a specific time point to strengthen the inference on the other ones, while keeping a separate measurement model to better capture the inherent features of each experiment performed. We allowed the model to account for many characteristics that arise in the context analyzed, such as the inflation of zeros, the overdispersion in both the background (noise) and the signal of the data, the presence of different antibodies used and many replicates available for each experimental condition. We showed in a simulated environment the performances of our algorithm in comparison to other existing methods, assessing that it has a better degree of classifying the latent states of the regions as bound or not by the protein, scoring lower values of misclassification error in terms of observed False Non Discovery Rate. We applied then the model to real data and compared the results to the ones obtained with [1], showing a greater capability of detecting enriched bins and providing lower misclassification error in the process. Exploiting the temporal information can help with the detection of the regions that are enriched because bound by the protein and not due to a bias effect induced by a prolonged activity of the antibody used for the experiment. Also, replicates with low efficiency and ratio between signal and background can still be used to strengthen the inference about the parameters of the model, without having a negative huge impact on the recovering of the latent state of the genomic location, enabling the use of all the information available. Further developments include the multivariate modelling of the multiple antibodies, through the use of multivariate discrete distributions or other approach that allows for the specification of a

dependency structure (similarity) between antibodies and/or transcription factors. Extending the mixture distribution to incorporate more components to better characterize the signal in the data can also be done. A more flexible parametrization of the spatio-temporal dependency could be introduced, using more than two transition probabilities in order to relax some of the initial assumptions of our model. The pre-processing of the data, such as the choice of the width of the bins used to summarize the counts, could be incorporated at a modelling level. The MCMC algorithm can also be improved in terms of computational speed.

Bibliography

[1] Bao, Y., Vinciotti, V., Wit, E., & Hoen, P. T. "Joint modelling of ChIP-seq data via a Markov random field model." *Biostatistics* 15.2 (2014): 296-310.

[2] Bardet, A. F., He, Q., Zeitlinger, J., & Stark, A. "A computational pipeline for comparative ChIP-seq analyses." Nature protocols 7.1 (2012): 45-61.

[3] Benjamini, Yoav, and Yosef Hochberg. "Controlling the false discovery rate: a practical and powerful approach to multiple testing." Journal of the Royal Statistical Society. Series B (Methodological) (1995): 289-300.

[4] Hilbe, J.M. Negative Binomial Regression. Cambridge University Press, 2011.

[5] Lauritzen, Steffen L. Graphical models. Oxford University Press, 1996.

[6] Kharchenko, Peter V, Tolstorukov, Michael Y & Park, Michael Y "Design and analysis of ChIP-seq experiments for DNA-binding proteins" Nature Biotechnology 26, 1351 - 1359 (2008)

[7] Kindermann, Ross, Snell, J. Laurie "Markov Random Fields and Their Applications" American Mathematical Society (1980).

[8] Kuan, Pei Fen, et al. "A statistical framework for the analysis of ChIP-Seq data." *Journal of the American Statistical Association* 106.495 (2011): 891-903.

[9] Ramos et al. "Genome-wide assessment of differential roles for p300 and CBP in transcription regulation" *Nucleic Acids Research* 38(16):5396-5408.

[10] Qin, Zhaohui S., et al. "HPeak: an HMM-based algorithm for defining read-enriched regions in ChIP-Seq data." *BMC bioinformatics* 11.1 (2010): 369.

[11] Spyrou, C., Stark, R., Lynch, A. G., & Tavaré, S. "BayesPeak: Bayesian analysis of ChIP-seq data." *BMC bioinformatics* 10.1 (2009): 299.

[12] Zeng, Xin, et al. "jMOSAiCS: joint analysis of multiple ChIP-seq datasets." *Genome biology* 14.4 (2013): R38.

Chapter 3

Mixture model with multiple allocations for clustering spatially correlated observations in the analysis of ChIP-Seq data

Saverio Ranciati[1,2], **Cinzia Viroli**[2], **Ernst C. Wit**[1]

[1] *University of Groningen, Johann Bernoulli Institute of Mathematics and Computer Science, Groningen, the Netherlands.*
[2] *Department of Statistics, University of Bologna, Bologna, Italy*

Abstract

In the context of Next-Generation Sequencing (NGS) experiments, the signal observed in the data might be produced by two (or more) different biological processes operating together and a gene could participate in both (or several) of them. Model-based clustering is a technique widely used to group a collection of units into mutually exclusive groups: there are, however, situations as the aforementioned NGS experiments where an observation could in principle belong to more than one cluster. We propose a novel approach to cluster NGS discrete data, coming from a ChIP-Seq experiment, with a mixture model, allowing each unit to belong potentially to more than one group: these multiple allocation clusters can be flexibly defined via a function combining the features of the original groups without introducing new parameters. The formulation naturally gives rise to a 'zero-inflation group' in which values close to zero can be allocated, acting as a correction for the abundance of zeros that manifest in this type of data. We take into account the spatial dependency between observations, which is described through a latent Conditional Auto-Regressive process that can reflect different dependency patterns. We assess the performance of our model within a simulation environment and then we apply it to a ChIP-Seq experiment to evaluate the genome wide role of the p300 protein in transcription regulation.

Keywords: ChIP-Seq; Heterogeneity; Model-based clustering; Multiple allocations; Spatial dependency.

3.1 Introduction

In the last 15 years, the development of parallel massively sequencing platforms for mapping the genome has completely revolutionized the way of studying gene expression patterns. These recent technologies called Next Generation Sequencing (NGS) allow to simultaneously investigate thousands of features within a single reliable and cost-effective experiment, thus representing a valid alternative to microarray experiments in enhancing our understanding of how genetic differences affect health and disease [15]. Indeed, these innovative platforms have been quickly applied to many genomic contexts giving rise to a large amount of available data in complex form. Data coming from such experiments are highly structured and their analysis has raised an imperative need for specific methodologies: technical or biological replicates can be observed in several experimental conditions [3], and the single observational units, such as genes or exons, are very likely characterized by spatial dependencies and relationships [14]. Moreover, the abundance of a particular transcript is measured as a count and so the single data point is a discrete measurement.

We consider in this paper the Chromatin ImmunoPrecipitation and sequencing (ChIP-Seq) data on proteins p300 and CBP analysed by [17]. In this experiment, two technical replicates are observed after 30 minutes from the initial interaction of the proteins with the human genome, with the aim of discovering the binding profile of these transcription factors. Figure 3.1 displays summarized counts for 1000 base pairs contiguous windows along chromosome 21. The plot shows segments with high read counts and segments where there is a uniformly low level of signal, thus suggesting a potential spatial effect.

In ChIP-Seq data like the one investigated (see [12] for a review), researchers usually associate the counts to two specific components: a background level, which accounts for the noise in the process and the inactivity of the regions with respect to protein binding; a signal level, that is described by a higher counts of sequenced DNA fragments, indicating that the protein is actually interacting with those specific genomic regions. From a statistical point of view, the problem of the detection of such biological processes can be addressed by introducing a mixture model with the aim of identifying groups of genomic regions that exhibit similar expression patterns. Typically, conventional model-based clustering methods perform classification of units into mutually exclusive partitions. However, looking at Figure 3.1, it could be interesting to uncover components that may arise from the multiple overlapping action of the main aforementioned group processes. Multiple partitions can be obtained in (at least) two ways: (a) by fuzzy or 'soft' clustering that is the assignment of a unit to the group with some posterior probabilities [see, for instance, 5, 11] or (b) by accounting for multiple allocations directly within a generative model. Our proposal employs this second perspective that explicitly assumes that genomic units are fully co-actors of multiple processes in a model based framework. The idea stems from earlier contributions aimed at discovering multiple clusters. [4] introduced a probabilistic-based method to discover overlapping cellular processes and the associated gene regulation scheme. Given the complexity of a cellular system, they propose a decomposition of the observed continuous data matrix into layers repre-

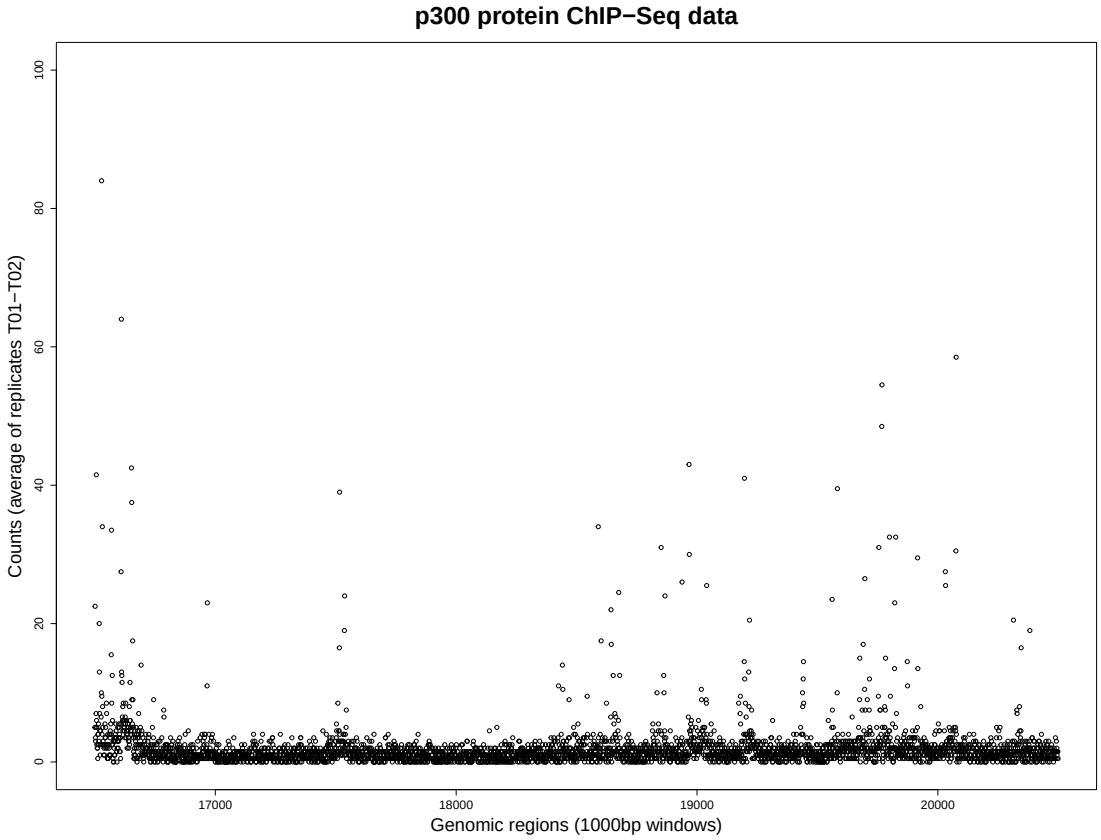

Figure 3.1: Summarized counts for 4000 bins from the chromosome 21. Regions on x-axis are 1000bp windows; averaged counts of the two replicates T01 and T02 reported on y-axis.

senting biological processes and groups of co-regulated genes, allowing every unit to express itself in more than one activity layer and belong to multiple clusters. In [1] and [8], the problem of multiple allocation is solved within a model based clustering strategy, where the distribution of the groups is extended generalizing the Gaussian probability distribution used in [4] to the case of exponential family distributions. The main idea of such approach known as 'Model-Based Overlapping Clustering' is to re-parameterize the density of the overlapped clusters as the product of some primary densities that, being members of the exponential family, still result in the same parametric family. [10] extended these models by employing a nonparametric Bayesian technique to infer the number of groups in their overlapping clusters model, while maintaining the characterization of the mixture densities as members of the exponential family. More recently, [19] proposed the 'epistatic model based clustering' for the analysis of microarray data. In this approach, a more explicit description of the mixed component densities in terms of Gaussians is given; different interactions between the parameters of the primary groups are investigated but the order of the interactions between these original clusters and the overlapped counterparts is practically limited to the second order.

The aim of this work is to define a general Multiple Allocation Mixture (MAM) model for analyzing the ChIP-Seq data. The peculiar features of these experimental data demand for specific treatment. First, their discrete nature and a marked overdispersion require a flexible count distribution such as the Negative Binomial, that however does not generally belong to the exponential family, unless its dispersion parameter is known and fixed. To this aim we generalize the model-based overlapping clustering to arbitrary parametric probabilistic functions. In addition, as shown in Figure 3.1, ChIP-Seq data are characterized by the inflation of non-structural zeros. These aspects are naturally taken into account by the proposed model, where each component of the multiple mixture corresponds to a primary or to an overlapping cluster distributed as Negative Binomials with parameters that are function of the primary parameters only. A further important aspect that emerges from Figure 3.1 is that the protein interactions with DNA are spatially correlated. We will show that the model can be easily extended in order to account for the spatial linkage among the genes, via a Conditional Auto-Regressive (CAR) process.

In what follows, we will present our proposal in three gradual steps in order to sequentially address these issues. First, in Section 2, the general MAM model will be presented and then we will adapt the model to the NGS data. We will illustrate how to extend this approach in order to model spatial dependent observations. In Section 3, a simulation study aimed at investigating the flexibility and effectiveness of the proposal is presented. Furthermore, we use the newly developed model to study the genome wide binding of the protein p300 in order to investigate its transcriptional regulatory function. In Section 4, conclusions are discussed.

3.2 Methods

3.2.1 Model-based clustering with mixture model

Finite mixture models have been receiving a wide interest in the statistical literature as a tool for performing model based clustering and density estimation [see 2, 7, 13].

Let $\boldsymbol{y}_j$ ($j = 1, \ldots, p$) be an observed vector of values that we want to classify in some unknown groups. The conventional model based clustering model assumes

$$f(\boldsymbol{y}_j|\Theta) = \sum_{i=1}^{k} \pi_i f(\boldsymbol{y}_j|\boldsymbol{\theta}_i), \tag{3.1}$$

where $f(\boldsymbol{y}_j|\boldsymbol{\theta}_i)$ are the component densities with parameters θ_i and π_i are the prior probabilities to belong to each component, satisfying $\pi_i > 0$ and $\sum_{i=1}^{k} \pi_i = 1$. According to this model, a sample of p observations arises from some underlying populations of unknown proportions and the purpose is to decompose the sample into its mixture components, where each component corresponds to a cluster. In doing so, the data are partitioned into mutually exclusive k groups. This is achieved by introducing a latent variable, say z_{ji}, which allocate each observation j to the component i. More precisely, $\boldsymbol{z}_j$ is a vector of length k that takes the value 1 in correspondence of the cluster assignment, and 0 elsewhere, so that $\sum_{i=1}^{k} z_{ji} = 1$. According to the maximum a posteriori probability (MAP) rule, the partition is then obtained by assigning subjects to their most likely class according to the posterior probabilities of z given the data:

$$f(z_{ji}|\boldsymbol{y}_j; \Theta) = \frac{\pi_i f(\boldsymbol{y}_j|z_{ji}; \boldsymbol{\theta}_i)}{\sum_{i'=1}^{k} \pi_{i'} f(\boldsymbol{y}_j|z_{ji'}; \boldsymbol{\theta}_{i'})}.$$

In this sense the classification produced by a mixture model is 'hard' (because a unit is allocated to the mixture component with the maximum posterior probability of belonging to) but in principles it could be 'soft' by assigning each cluster a weight that equals its posterior probability as in the partial membership model [11]. However, a soft assignment perspective does not mitigate the limitation of the model based classification, that results when data points may simultaneously belong to multiple clusters. In such situations, a change in the generative model is required, by explicitly assuming that the allocation vector $\boldsymbol{z}_j$ may contain several - and not just a single - ones.

3.2.2 Multiple Allocation Mixture model

In order to construct a new generative model for multiple components, we define k in Eq. (3.1) as the number of primary groups which are not mutually exclusive. For such a reason, we assume the prior probabilities of each primary group satisfying the constraint $\pi_i > 0$ but not necessarily summing up to one, $\sum_{i=1}^{k} \pi_i \neq 1$.

The total number of possible single and multiple allocation clusters is $k^* = 2^k$ and $\sum_{i=1}^{k} z_{ji} = n_j$ is the multiplicity of the cluster membership for the unit jth. When $n_j = 1$ we have a single group allocation, otherwise we have multiple group allocations. More precisely, if $n_j = 2$ the unit jth belongs to two groups simultaneously. These two groups altogether may be thought of as a new *secondary* group. If $n_j = 3$ the unit belongs to three groups that jointly define a *tertiary* group and so forth. When $n_j = 0$, the unit is assigned to what we call a 'outward cluster': this group collects observations that ideally do not belong to any clusters, and for this reason their distribution might be described by peculiar parameters depending on the empirical context. For instance, in many applications it could represent a group

of outliers or noisy observations, characterized by high variance. The definition and existence of the 'outward cluster' is particularly relevant for the analysis of ChIP-Seq data, where the clusters are interpretable as biological processes. A gene that does not take part to any biological processes will have extremely low values (close to zero or zero). Thus, the outward cluster has the purpose to describe the group of 'inactive genes' and, in so doing, it acts as a zero-inflation adjustment for the model.

For k fixed, let U be a connection matrix of dimension $2^k \times k$, with elements $u_{hi} \in \{0, 1\}$, containing all the possible assignment configurations. For instance, for $k = 3$:

$$U = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}. \tag{3.2}$$

In this case we define the prior probability to belong to a general (single or multiple allocation) group

$$\pi_h^* = \prod_{i=1}^{k} \pi_i^{u_{hi}} (1 - \pi_i)^{1-u_{hi}} \tag{3.3}$$

with $h = 1, \ldots, k^*$. Notice that these weights are constructed so that $\sum_{h=1}^{k^*} \pi_h^* = 1$, and the definition comes naturally from the representation of clusters as sets in a sample space. The Multiple Allocation Mixture model (MAM) can be defined as a reparameterization of Eq. 3.1 in the new formulation

$$f(\boldsymbol{y}_j|\Theta) = \sum_{h=1}^{k^*} \pi_h^* f(\boldsymbol{y}_j|\psi(\boldsymbol{u}_h, \boldsymbol{\theta}), \phi_h) \tag{3.4}$$

where $\psi(\boldsymbol{u}_h, \boldsymbol{\theta})$ is a location parameter and ϕ_h is a nuisance parameter. More specifically, $\psi(\boldsymbol{u}_h, \boldsymbol{\theta})$ is a function that depends on $\boldsymbol{u}_h$, which is the h-th row of U, and transforms the primary parameters $\boldsymbol{\theta}_i$ into the parameters of the multiple allocation components according to several possible schemes:

- *additive model*: the parameters of the mixed groups could be the sum of the original parameters, that is

$$\psi(\boldsymbol{u}_h, \boldsymbol{\theta}) = \sum_{i=1}^{k} u_{hi} \boldsymbol{\theta}_i + \boldsymbol{\theta}_b \mathbb{1}_{\left[\sum_{i=1}^{k^*} u_{hi} = 0\right]},$$

  where in the second term the indicator function is introduced to account for the parameters of the outward mixture component;

- *co-dominance of order 1*:

$$\psi(\boldsymbol{u}_h, \boldsymbol{\theta}) = \frac{\sum_{i=1}^{k} u_{hi}\boldsymbol{\theta}_i}{\sum_{i=1}^{k} u_{hi}} + \boldsymbol{\theta}_b \mathbb{1}_{\left[\sum_{i=1}^{k^*} u_{hi}=0\right]};$$

- *co-dominance of order 0*:

$$\psi(\boldsymbol{u}_h, \boldsymbol{\theta}) = \left(\prod_{i=1}^{k} \boldsymbol{\theta}_i^{u_{hi}}\right)^{1/\sum_{i=1}^{k} u_{hi}} + \boldsymbol{\theta}_b \mathbb{1}_{\left[\sum_{i=1}^{k^*} u_{hi}=0\right]}.$$

We will focus on the additive scheme through the rest of the manuscript. Finally, the nuisance parameters ϕ_h are taken h-specific because we aim to add more flexibility to the mixture model.

3.2.3 Multiple Allocation Mixture model for ChIP-Seq Data

Suppose we observe the ChIP-Seq counts of p genes in D biological conditions or replicates. We denote Y_{jd} the random variable that expresses the read counts, say y_{jd}, mapped to gene j (j=1, ..., p), in sample d with $d = 1, \ldots, D$. Let $\boldsymbol{Y}_j$ be the random vector of length D denoting the expression profile of a gene. Let $\boldsymbol{y}_j$ be the observed value. We assume that Y_{jd} is distributed according to the Negative Binomial ($\mathcal{NB}$) distribution, given both the discrete nature of the observations and the flexibility of having a specific parameter used to model the overdispersion, which makes this distribution preferable over the Poisson. We further assume that, conditional on the group, the replicates are independent draws so that the mixture model in Eq. 3.4 becomes:

$$f(\boldsymbol{y}_j|\Theta) = \sum_{h=1}^{k^*} \pi_h^* \prod_{d=1}^{D} \mathcal{NB}\left(y_{jd}|\psi(\boldsymbol{u}_h, \boldsymbol{\mu}_d), \phi_{dh}\right), \tag{3.5}$$

where ϕ_{dh} are specific dispersion parameters of the negative binomials and $\mu_{dh}^* = \psi(\boldsymbol{u}_h, \boldsymbol{\mu}_d)$ are the means defined in the extended space of multiple components.

We allow dispersion parameters ϕ_{hd} to vary for each of the k^* possible assignment and replicates because it is not directly clear a priori what is the most reasonable variance structure to define through a function for the multiple allocation clusters. This allows, nevertheless, for a certain degree of flexibility in describing the context-specific variability in the data.

With reference to the means μ_{dh}^* they are modelled as function of primary k means through the combination scheme ψ. More specifically, with reference to the connection matrix U in Eq. 3.2, μ_{d1}^* is the mean parameter in condition d of the outward distribution; μ_{d2}^* is the mean parameter of the units that belong to the first primary group only and not to mixed groups, and so on, till μ_{d8}^* that is the mean of the units that belong simultaneously to the three components under condition d. Given the nature of the data we consider the 'outward' group as the component devoted to describe the inactive genes and for this reason we fix its mean to a very low value, such as $\mu_{d1}^* = 0.01$. The other mean parameters are obtained as combination scheme through ψ of k primary values that represent the means of the units that belong to

the primary groups or related multiple groups. In order to construct an hierarchical form of the model, we define with $\boldsymbol{z}^*$ the allocation matrix in the augmented space given by k^* as a $p \times k^*$ 'multinomial' matrix. While each row of $\boldsymbol{z}$ can have multiple ones according to the assignment of the unit to multiple clusters, the rows of $\boldsymbol{z}^*$ only have a single one and the matrix U allows for a connection between original parametrization allocation and the re-parametrized version. Given the whole set of parameters, say Θ, the joint complete likelihood of our model is the following:

$$P(\boldsymbol{y}, \boldsymbol{z}^*|\Theta) = P(\boldsymbol{y}|\boldsymbol{z}^*, \Theta) \cdot P(\boldsymbol{z}^*|\Theta). \tag{3.6}$$

Inference is carried out within a Bayesian framework. The posterior distribution of the parameters and the latent variables is:

$$P(\boldsymbol{z}^*, \Theta, \boldsymbol{\xi}|\boldsymbol{y}) \propto P(\boldsymbol{y}|\boldsymbol{z}^*, \Theta) \cdot P(\boldsymbol{z}^*|\Theta) \cdot P(\Theta|\boldsymbol{\xi}) \tag{3.7}$$

with $P(\Theta|\boldsymbol{\xi})$ specifying priors for the mixture parameters and $\boldsymbol{\xi}$ as the vector containing the hyper-parameters. The posterior in Eq. 3.7 can then be rewritten as:

$$\begin{aligned}
&P(\boldsymbol{z}^*, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\phi}, \boldsymbol{\xi}|\boldsymbol{y}) \propto P(\boldsymbol{y}|\boldsymbol{z}^*, \boldsymbol{\mu}, \boldsymbol{\phi}) P(\boldsymbol{z}^*|\boldsymbol{\pi}) P(\boldsymbol{\pi}|\boldsymbol{\xi}) P(\boldsymbol{\mu}|\boldsymbol{\xi}) P(\boldsymbol{\phi}|\boldsymbol{\xi}) = \\
&= \prod_{j,d,h}^{p,D,k^*} \left\{ \frac{\Gamma(\phi_{hd} + y_{jd})}{\Gamma(\phi_{hd})\Gamma(y_{jd}+1)} \left[\frac{\phi_{hd}}{\phi_{hd} + \psi(\boldsymbol{u}_h, \boldsymbol{\mu}_d)} \right]^{\phi_{hd}} \left[\frac{\psi(\boldsymbol{u}_h, \boldsymbol{\mu}_d)}{\phi_{hd} + \psi(\boldsymbol{u}_h, \boldsymbol{\mu}_d)} \right]^{y_{jd}} \right\}^{z^*_{jh}} \\
&\times \prod_{j,h}^{p,k^*} \left(\prod_{i=1}^{k} \pi_i^{u_{hi}} (1-\pi_i)^{1-u_{hi}} \right)^{z^*_{jh}} P(\boldsymbol{\pi}|\boldsymbol{\xi}) P(\boldsymbol{\mu}|\boldsymbol{\xi}) P(\boldsymbol{\phi}|\boldsymbol{\xi}),
\end{aligned} \tag{3.8}$$

where $P(\boldsymbol{\pi}|\boldsymbol{\xi}), P(\boldsymbol{\mu}|\boldsymbol{\xi})$ and $P(\boldsymbol{\phi}|\boldsymbol{\xi})$ are prior distributions for these quantities. As prior distribution for the weights we assume a Beta distribution so that we get the following hierarchical structure:

$$\begin{aligned}
\pi_i &\sim \text{Beta}\,(1,1) \\
P(\boldsymbol{z}^*_{jh} = 1|\boldsymbol{\pi}^*) &= \pi^*_h \\
\boldsymbol{y}_{jd}|\boldsymbol{z}^*_{jh}; \boldsymbol{\Theta} &\sim \mathcal{NB}(\psi(\boldsymbol{u}_h, \boldsymbol{\mu_d}); \boldsymbol{\phi}).
\end{aligned} \tag{3.9}$$

For the other two parameters we select conjugate and flat priors. More precisely, $\mu_{id} \sim \text{Gamma}(a_\mu, b_\mu)$ and $\phi_{hd} \sim \text{Unif}(a_\phi, b_\phi)$, for every i, d and h, where $(a_\mu, b_\mu, a_\phi, b_\phi)$ are elements of the vector $\boldsymbol{\xi}$ of hyperparameters. Given these chosen priors, full conditionals can be derived and a Gibbs sampling MCMC algorithm applied to estimate the parameters. At the implementation step of the algorithm we exploit the Gamma-Poisson mixture representation of a Negative Binomial distribution. We introduce a further latent variabile s, specific to each unit j in each replicate/condition d, that

has a Gamma density with shape and rate parameters equal to ϕ_{hd}. It follows that:

$$\begin{aligned} f(s_{jd}) &= \text{Gamma}(\phi_{hd}, \phi_{hd}) \\ p(y_{jd}|s_{jd}) &= \text{Pois}(\psi(\boldsymbol{u}_h, \boldsymbol{\mu}_d)s_{jd}) \\ p(y_{jd}) = \int p(y_{jd}|s_{jd})f(s_{jd})ds &= \mathcal{NB}(\psi(\boldsymbol{u}_h; \boldsymbol{\mu}_d), \phi_{hd}) \end{aligned}$$

which allows us to use a Gibbs sampler for the mean parameters μ_{id}.

3.2.4 Modelling spatial correlation with a CAR structure

The units/observations (in our case, genes or genomic locations) are not independent but spatially correlated. We introduce the spatial correlation by allowing the primary weights π_i to be j-varying and we denote them as π_{ij}. The spatial relationship is taken into account by allowing the weights of the mixture in Eq. 3.4 to vary from one gene to another. The way we formulate it is inspired by the work proposed by [6]. They first introduced k independent p-dimensional latent variables with a Markov random-field distribution. The weights are then a non-linear function of these latent variables and spatial relationships are expressed in terms of neighborhood relationships. This could be restrictive because the correlation between two genes should decrease as their distance increases. We extend the approach by considering a Gaussian conditional auto-regressive model [16], where the distances are directly used to model correlations instead of dummies denoting the neighborhood condition. In a Bayesian framework, this is accomplished by introducing additional layers to the hierarchy formulation shown in Eq. 3.9, with their own set of hyperparameters. With reference to $\boldsymbol{\pi}$, we introduce the spatial latent vectors, denoted by $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_i, \ldots, \boldsymbol{x}_k$, with $i = 1, \ldots, k$: each $\boldsymbol{x}_i$ is a Gaussian conditional autoregressive model given by

$$f(\boldsymbol{x}_i) = \mathcal{N}(\mathbf{0}, Q^{-1}) \tag{3.10}$$

where Q is a precision matrix of order p and $\gamma_{jj'}$ is some non-linear function of $\delta_{jj'}$, which are the distances between all the units. More specifically,

$$Q = I_p + \Delta - \Gamma = \begin{cases} 1 + \sum_{j'=1}^{p} \gamma_{jj'} & \text{if } j\text{-th diagonal element} \\ -\gamma_{jj'} & \text{elsewhere} \end{cases}$$

and

$$\Delta = \begin{bmatrix} \ddots & 0 & \dots & 0 \\ 0 & \sum_{j'=1}^{p} \gamma_{jj'} & \dots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \dots & 0 \end{bmatrix} \quad \Gamma = \begin{bmatrix} 0 & \gamma_{1,2} & \cdots & \gamma_{1,p} \\ \gamma_{2,1} & 0 & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{p,1} & \dots & \dots & 0 \end{bmatrix}.$$

After some algebraic steps it is possible to show that Eq. 3.10 is equivalent to

$$f(\boldsymbol{x}_i) = c \cdot \exp\left\{ -\frac{1}{2} \left[\sum_{j=1}^{p} \sum_{j'=1}^{p} \gamma_{jj'}(x_{ij} - x_{ij'})^2 + \sum_{j=1}^{p} x_{ij}^2 \right] \right\} \tag{3.11}$$

with c the normalization constant

$$c = (2\pi)^{-p/2} \prod_{j=1}^{p} (1 + v_j)^{1/2}.$$

In the previous expression, v_j's denote the eigenvalues of the spatial matrix $\Delta - \Gamma$. Given $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$, the weights for location j can be obtained via logistic formulation

$$\pi_{ij} = \frac{\exp(x_{ij}/\eta)}{1 + \exp(x_{ij}/\eta)}$$

where η is a 'shrink-or-stretch' tuning parameter to be estimated that provides a way to exaggerate the differences in units allocation among the clusters.

3.2.5 Conditional Auto-Regressive Multiple Allocation Mixture (*CAR-MAM*)

In order to account for spatial correlation between the units/observations (in our case, genes or genetic locations), we introduce another layer in the hierarchical structure of the model. Starting from Eq. 3.6, the updated joint complete likelihood is:

$$P(\boldsymbol{y}, \boldsymbol{z}^*, \boldsymbol{x} | \boldsymbol{\mu}, \boldsymbol{\phi}, \eta, \boldsymbol{\xi}) = P(\boldsymbol{y} | \boldsymbol{z}^*, \boldsymbol{\mu}, \boldsymbol{\phi}) P(\boldsymbol{z}^* | \boldsymbol{x}, \eta) P(\boldsymbol{x} | \boldsymbol{\xi}) \tag{3.12}$$

leading to the following posterior distribution

$$\begin{aligned} P(\boldsymbol{z}^*, \boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\phi}, \eta | \boldsymbol{y}, \boldsymbol{\xi}) \quad \propto \quad & P(\boldsymbol{y} | \boldsymbol{z}^*, \boldsymbol{\mu}, \boldsymbol{\phi}) P(\boldsymbol{z}^* | \eta, \boldsymbol{x}) \times \\ & P(\boldsymbol{x} | \boldsymbol{\xi}) P(\boldsymbol{\mu} | \boldsymbol{\xi}) P(\boldsymbol{\phi} | \boldsymbol{\xi}) \end{aligned} \tag{3.13}$$

where the vector $\boldsymbol{\xi}$ now also contains the hyper-parameters for the new latent layer and $P(\boldsymbol{y} | \boldsymbol{z}^*, \boldsymbol{\mu}, \boldsymbol{\phi})$ is defined as in Eq. (3.8). More precisely, the complete latent structure in Eq. 3.13 is equal to:

$$P(\boldsymbol{z}^*, \boldsymbol{x} | \eta) = \prod_{i=1}^{k} c \exp \left\{ -\frac{1}{2} \left[\sum_{j=1}^{p} \sum_{j'=1}^{p} \gamma_{jj'} (x_{ij} - x_{ij'})^2 + \sum_{j=1}^{p} x_{ij}^2 \right] \right\}$$

$$\times \prod_{j=1}^{p} \prod_{h=1}^{k^*} \left[\prod_{i=1}^{k} \left(\frac{\exp(x_{ij}/\eta)}{1 + \exp(x_{ij}/\eta)} \right)^{u_{hi}} \left(1 - \frac{\exp(x_{ij}/\eta)}{1 + \exp(x_{ij}/\eta)} \right)^{1 - u_{hi}} \right]^{z^*_{jh}}$$

where c is the constant defined in Eq. 3.11. As proposed in [6], we integrate out the latent allocation variable $\boldsymbol{z}$ when implementing the Metropolis sampler for $\boldsymbol{x}$ in order to employ the information carried by the data $\boldsymbol{y}$ and bring the two layers of the hierarchical structure of the model closer together.

3.3 Results

3.3.1 Simulation study — Multiple Allocation Mixture (*MAM*) model

We assess the performance of our model (*MAM*) under different scenarios and we compare it with a classical non-overlapping components mixture (*NegBinMix*). Data are generated from two independent Negative Binomial distributions ($D = 2$)

Table 3.1: Misclassification error rate (in percentage) for *NegBinMix* and *MAM* for the three scenarios of activation.

Number of clusters	Model	Degree of activation		
		$\pi_i = 0.25$	$\pi_i = 0.50$	$\pi_i = 0.75$
$k = 2$	*MAM*	**1.05**	2.70	2.65
(i.e. $k^* = 4$)	*NegBinMix*	1.20	**2.65**	**2.55**
$k = 3$	*MAM*	**0.95**	**3.05**	**5.10**
(i.e. $k^* = 8$)	*NegBinMix*	11.00	25.05	9.15

and $p = 2000$ units, allowing for overlapping clusters with a number of groups $k = \{2, 3\}$: in the augmented k^* space this equals to represent a situation, as in a classical model-based clustering framework, where the actual number of groups ranges from $k^* = 4$ to $k^* = 8$. We explore three degrees of clustering between primary and outward/non-primary groups by selecting three scenarios which we call low, medium and high activation: this is achieved by setting all the π equal to - respectively - 0.25, 0.50 and 0.75. We run both algorithms (*MAM* and *NegBinMix*) for 10000 MCMC iterations and a 5000 burn-in window is selected. For every μ_{id}, we choose hyperparameters of the Gamma prior distributions equal to $a_\mu = 1$ and $b_\mu = 0.001$; for every dispersion parameter ϕ_{hd}, we select the ranges of the prior uniform distributions to be $[a_\phi = 100, b_\phi = 2000]$. Convergence is checked for every chain and we assign to the clusters according to the maximum a posteriori rule: we compute the posterior probabilities of the allocation vectors $\boldsymbol{z}_j^*$ given the data and, for every unit, we allocate to the component with the highest probability value (see also Section 3.2.1). We choose as an overall performance indicator the misclassification error rate (see Table 3.1), that is, the average number of units not correctly allocated when compared to the known true membership. The posterior means for the parameters in the selected models (both the overlapping and non-overlapping mixtures) are consistent with the true values and do not show any substantial bias.

We report the misclassification error rates for the estimated models in Table 3.1: the percentage of misclassified units is always lower in the low activation scenario because most of the observations are allocated in the outward component, which has small variance and near zero (fixed) mean, thus simplifying the clustering task. As we can see in Table 3.1, our model has comparable (sometimes better) classification rates, with respect to the conventional mixture of Negative Binomial, in simpler simulated dataset ($k = 2$). When $k = 3$, *MAM* model always outperforms the compared mixture with noticeable improvements on the misclassification error rate.

3.3.2 Simulation study — *MAM* with Conditional Autoregressive model (*CAR-MAM*)

We simulate data from a mixture of Negative Binomials with multiple allocation and spatial information; we choose $k = 2, 3, 4$ ($k^* = 4, 8, 16$) and for every number of groups we adopt two different spatial structures. In the former, the latent variable $\boldsymbol{x}$ is drawn from a CAR model using a reciprocal function $\gamma_{jj'}$ [see 16], which is an inverse function of the distances between uniformly drawn positions (pos_j, $j =$

Table 3.2: Average misclassification error rate (in percentage) over 100 simulated datasets with standard errors in brackets; CAR structure with reciprocal function.

Number of clusters	*CAR-MAM*	*MAM*	*NegBinMix*
$k = 2$ $(k^* = 4)$	**26.58** (*0.001*)	26.87 (*0.001*)	30.40 (*0.001*)
$k = 3$ $(k^* = 8)$	**23.60** (*0.001*)	28.65 (*0.009*)	41.00 (*0.002*)
$k = 4$ $(k^* = 16)$	**36.32** (*0.009*)	49.84 (*0.013*)	60.39 (*0.002*)

Table 3.3: Average misclassification error rate (in percentage) over 100 simulated datasets with standard errors in brackets; CAR structure with sine function.

Number of clusters	*CAR-MAM*	*MAM*	*NegBinMix*
$k = 2$ $(k^* = 4)$	**27.71** (*0.002*)	29.13 (*0.001*)	35.42 (*0.001*)
$k = 3$ $(k^* = 8)$	**31.31** (*0.006*)	35.06 (*0.011*)	40.11 (*0.002*)
$k = 4$ $(k^* = 16)$	**44.93** (*0.005*)	56.12 (*0.007*)	57.18 (*0.003*)

$1, \ldots, p$). In the latter, a sine function is employed in the data generating process, in order to have stronger spatial relationships. More precisely, we assume that the spatial latent vectors, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ are:

$$x_{ij} = sin\left(i\pi \frac{pos_j}{max\left\{pos_j\right\}_{j=1,\ldots,p}}\right).$$

For each scenario we run three algorithms assuming k is known: *NegBinMix*, which is a mixture of Negative Binomials without any further specification, and our two proposed models *MAM* and *CAR-MAM* assuming that the conditional autoregressive structure is computed through a reciprocal function $\gamma_{jj'}$. We average the misclassification error rates across 100 independent datasets simulated for each scenario and we compare the results to assess the performance. Again, we cluster the units according to the maximum a posteriori rule and we measure the performance through the misclassification error, computed for each model. In Table 3.2 and Table 3.3 we summarize the results of 100 runs of both scenarios, where data were generated - respectively - with a reciprocal function for the *CAR* part of the model or a sine function.

As clear from the tables, we achieve improved accuracy in the clustering task with *CAR-MAM* model over the simpler *MAM* model and both of them perform better with respect to *NegBinMix* (Table 3.2) even when the estimated conditional autoregressive structure is not the same as the one used in the data generating process (Table 3.3).

3.3.3 p300 protein binding ChIP-Seq experiment

We apply our model to the data already discussed in the introduction and previously analysed by [17]. p300 is a transcription coactivator associated with many genes that are involved in multiple processes (i.e., differentiation, apoptosis, proliferation); also, the protein serves as a bridge for other transcription factors and it is

involved in the transcription machinery of cells related to the development of cancer and other diseases. The data from p300 ChIP-Seq experiments consist of results from multiple interactions of the transcription coactivator with the chromatin in quiescent and stimulated cells, at different time points. We select two technical replicates (T01, T02) collected 30 minutes after the initial interaction between the chromatin and the protein p300 on a sequence of 1000 base-pairs windows describing the preprocessed raw counts across 4000 regions of the chromosome 21 (see Figure 3.1). We analyze both replicates jointly and we summarize the data of this subsample by computing the average of the counts for the two different replicates T01 and T02. As we can see in the plot, the majority of the observations lie in a 'band' of counts lower than 5, aggregated into segments that are spanned by smaller batch of regions exhibiting a higher count level, thus suggesting a spatial effect with respect to the protein binding process. Since in ChIP-Seq data researchers usually associate the counts to a background group and a signal group, we run the algorithm with $k = 2$ ($k^* = 4$) in order to capture the two aforementioned expected clusters and potentially a better characterization of them through our multiple allocation cluster, while simultaneously taking into account the spatial dependency between the genomic regions. We choose the geometric mean as our combination scheme $\psi(\cdot)$ to lessen the effect (on the multiple allocation cluster mean values) produced by the highest counts. Out of the 4000 genomic regions, 159 are allocated in the outward cluster, which accounts for the zero-inflation in the data, represented by the observations around positions 17200 and 18000 (see Figure 3.2). This group has fixed mean values equal to 0.01 for both replicates T01 and T02, while the dispersion parameters are estimated by the algorithm. The first primary cluster $i = 1$, whose units are indicated by green dots in Figure 3.2, has posterior means equal to $\boldsymbol{\mu}_1 = (1.61, 1.07)$ and represents the background process of the protein binding: 3478 possibly inactive genomic regions, with very low counts, are allocated in this group. The second primary cluster, $i = 2$, is representative of a signaling group of 48 genomic regions having a higher mean count level of $\boldsymbol{\mu}_2 = (19.27, 34.23)$ depicted with red dots in Figure 3.2. The multiple allocation cluster in our analysis is interpretable as a group of 315 units involved in both the background and signal clusters: in this case, given that we are observing these counts after 30 minutes from the initial interaction of the protein with the strand of chromosome 21, the genomic regions allocated in this cluster could be thought as either being locations that were active immediately at the beginning and now not signaling anymore or locations only starting to interact with the p300 protein after 30 minutes. The mean values for this multiple allocation cluster are equal to 5.57 and 6.06 in the two replicates T01 and T02: the units belonging to it are shown as blue dots in Figure 3.2. Finally, the posterior probability for each unit to be allocated in the signaling group is shown in Figure 3.2 as a magenta solid line. We can see from the plot that this probability is higher in those segments where genomic regions with higher counts are observed; moreover, the allocation weight of the signal component follows a spatial pattern, increasing and decreasing across the analyzed strand and thus accounting for the spatial effect occurring among the observations. For comparison, we also run the algorithm for a conventional four components Negative Binomial mixture model, fixing the location parameter of the first component to 0.01 as we do in our model.The result shows that only a total of

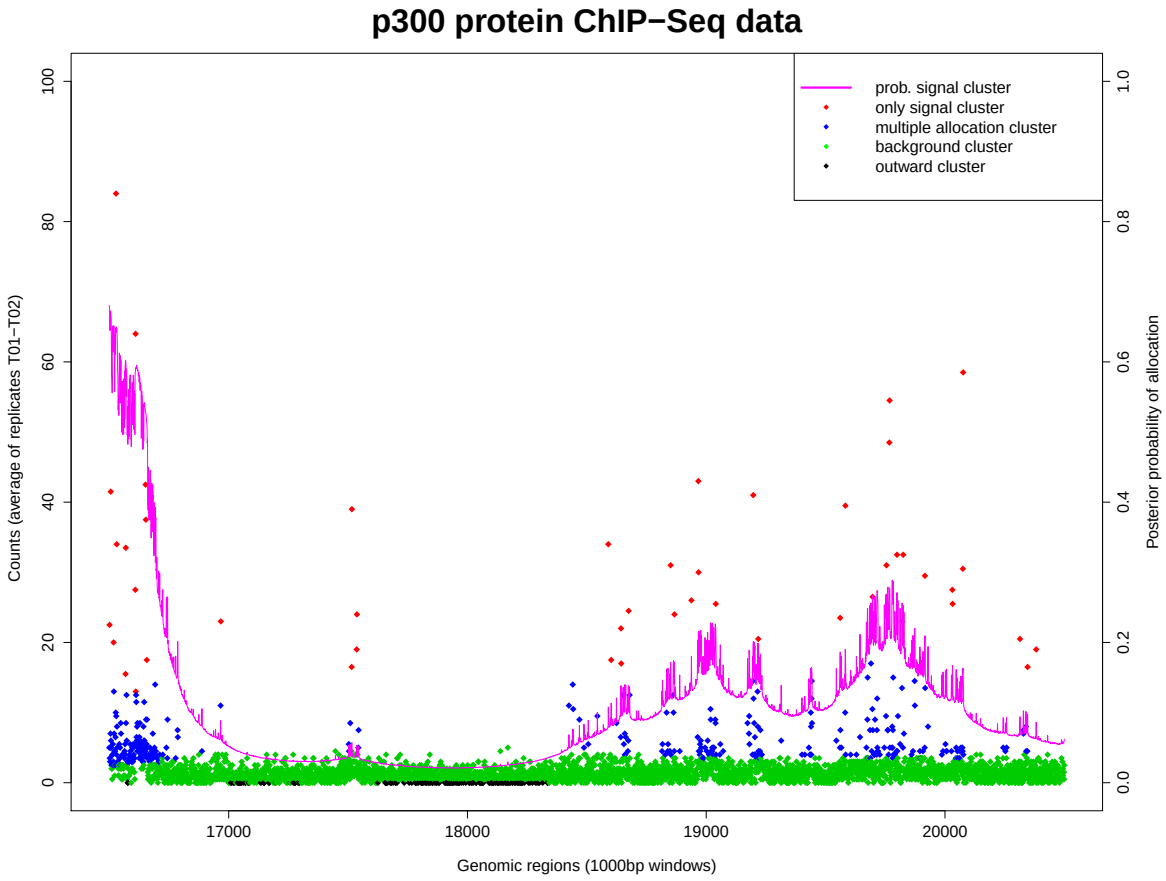

Figure 3.2: Clustering result for the analyzed segment. Counts are reported as average of the two technical replicates T01 and T02; solid line is the posterior probability to be allocated to the signal group.

three clusters are estimated, with one of them capturing the same background mean level shared by the fixed component, thus producing a blurred classification with respect to one of the two main processes of interest.

3.4 Conclusions

Motivated by the analysis of data coming from ChIP-Seq experiments, we proposed an extension of the conventional mixture model that allows for units to simultaneously belong to more than one group. As a by product of the model specification, an outward cluster has been introduced that can be used to describe specific features of the data such as zero-inflation, outliers and so forth. A spatial dependency layer among the units is encoded in the formulation by the means of a conditional autoregressive model, allowing for spatial information to aid the clustering task. We have compared our proposed model with a mixture of Negative Binomials to investigate its advantages with respect to the conventional approach: results on the simulated data show an increase in performance in terms of misclassification error rates. We applied our model to data previously analyzed by [17]: a promising richer description of the signal in the observations is found, calling for a potentially deeper biological investigation of those genomic regions associated with it. A delicate issue that deserves future investigation is the choice of k, which is a fundamental aspect because it identifies the number of primary clusters. Some well-known methods such as reversible jump [9] and birth-death process [18] do not allow an extensive exploration of the range of values for k without incurring in a dramatic increase in computational cost. However, in some applications, this choice could be suggested by the empirical context such as in our case, where a number of primary clusters equal to two was reasonable because they represented background and signal groups.

Acknowledgments

Conflict of Interest: None declared.

Bibliography

[1] Banerjee, A., Krumpleman, S., Basu, S., Mooney, R., and Ghosh, J. Model-based overlapping clustering. In ACM, editor, *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 532–537. 2005.

[2] Banfield, J.P. and Raftery, A.E. Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 49:803–821, 1993.

[3] Bao, Y., Vinciotti, V., Wit, E., and t' Hoen, P.A. Joint modeling of ChIP-seq data via a Markov random field model. *Biostatistics*, 15(2):296–310, 2014.

[4] Battle, A., Segal, E., and Koller, D. Probabilistic Discovery of Overlapping Cellular Processes and their regulation. *Journal of Computational Biology*, 12:909–927, 2005.

[5] Bezdek, J. C. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981.

[6] Fernández, C. and Green, P. J. Modelling spatially correlated data via mixtures: a Bayesian approach. *Journal of the Royal Statistical Society: Series B*, 64:805–826, 2002.

[7] Fraley, C. and Raftery, A.E. Model-based clustering, discriminant analysis and density estimation. *Journal of the American Statistical Association*, 97:611–631, 2002.

[8] Fu, Q. and Banerjee, A. Multiplicative Mixture Models for Overlapping Clustering. In ICDM'08, editor, *Data Mining, 2008. Eighth IEEE International Conference on. IEEE*, pages 791–796. 2008.

[9] Green, P. J. Reversible jump Markov Chain, Monte Carlo computation and Bayesian model determination. *Biometrika*, 82:711–732, 1995.

[10] Heller, A.K. and Ghahramani, Z. A Nonparametric Bayesian Approach to Modeling Overlapping Clusters. In ACM, editor, *International Conference on Artificial Intelligence and Statistics*, pages 187–194. 2007.

[11] Heller, A.K., Wiliamson, S., and Ghahramani, Z. Statistical Models for Partial Membership. In ACM, editor, *Proceedings of the 25th international conference on Machine learning*, pages 392–399. 2008.

[12] Kuan, Pei Fen, Chung, Dongjun, Pan, Guangjin, Thomson, James A, Stewart, Ron, and Keleş, Sündüz. A statistical framework for the analysis of ChIP-Seq data. *Journal of the American Statistical Association*, 106(495):891–903, 2011.

[13] McLachlan, G. J. and Peel, D. *Finite Mixture Models*. Wiley, 2000.

[14] Mo, Q. A fully Bayesian hidden Ising model for ChIP-seq data analysis. *Biostatistics*, 13(1):113–128, 2012.

[15] Nagalakshmi, U., Wang, Z., Waern, K., Shou, C., Raha, D., Gerstein, M., and Snyder, M. The transcriptional landscape of the yeast genome defined by RNA sequencing. *Science*, 320:1344:1349, 2008.

[16] Pettitt, A.N., Weir, I.S., and Hart, A.G. A conditional autoregressive Gaussian process for irregularly spaced multivariate data with application to modelling large sets of binary data. *Statistics and Computing*, 12:353–367, 2002.

[17] Ramos, Y., Hestand, M., Verlaan, M., Krabbendam, E., Ariyurek, Y., van Dam, H., van Ommen, G., den Dunnen, J., and Zantema, A. Genome-wide assessment of differential roles for p300 and CBP in transcription regulation. *Nucleic Acids Research*, 38(16):5396–5408, 2010.

[18] Stephens, M. Bayesian analysis of mixtures models with an unknown number of components - an alternative to reversible jump methods. *Annals of Statistics*, 20:40–74, 2000.

[19] Zhang, J. Epistatic Clustering: A Model-Based Approach for Identifying Links Between Clusters. *Journal of the American Statistical Association*, 108:1366–1384, 2013.

Chapter 4

Bayesian Smooth-and-Match estimation of ODEs' parameters with quantifiable solution uncertainty

Saverio Ranciati[1,2], **Cinzia Viroli**[2], **Ernst C. Wit**[1]

[1] *University of Groningen, Johann Bernoulli Institute of Mathematics and Computer Science, Groningen, the Netherlands.*
[2] *Department of Statistics, University of Bologna, Bologna, Italy*

Abstract

In many fields of application, dynamic processes that evolve through time are well described by systems of ordinary differential equations (ODEs). The parameters governing the equations cannot be immediately extrapolated out of the data when the analytical solution of the ODEs is not available. Different methods have been proposed to infer these quantities, from numerical optimization to more general models that employs regularization as a way to overcome the issue of trying to directly solve the system. We focus on the class of techniques that use smoothing to avoid direct integration and, in particular, on a Bayesian Smooth-and-Match strategy that allows to obtain the ODEs solution while performing inference on the parameters, with a built-in quantification of the related uncertainty. We assess the performance of the proposed approach in three different simulation studies and we compare the results on a dataset on neuron electrical activity.

Keywords: ordinary differential equations; smoothing; penalized cubic splines; MCMC; ridge regression.

4.1 Introduction

Many processes that evolve through time are described by systems of ordinary differential equations (ODEs). Imagine we have a simple case of one ODE where the change, $dx(t)$, of the concentration level of a specific molecule in the cell follows a law that is described by some function $g_\beta[x(t)]$, with a set of parameter $\boldsymbol{\beta}$ governing this law. We could think of this change as depending on the quantity of molecules at time t, that is $x(t)$, a rate β_1 at which new ones are produced by the cell as time passes by and then maybe some 'limit' β_2 on the capacity to contain them. What we observe, in practice, are not the actual changes $dx(t)$ (the derivative of the process) but instead the concentration levels $x(t)$ at sampling time points. This means that, in order to relate the data at our disposal with the parameters of interest, we need a solution for the system of differential equations. Solving the problem analytically however is not always possible and in that case no closed forms are available for the estimation of the parameters. Moreover, our observations may very well be affected by noise that perturbs the true temporal dynamic of the process. There are several techniques in the literature on this topic (see Robinson [25] for an introduction): most of them involve numerical integration, a straightforward approach to the problem that, however, does not take into account in any way the uncertainty about the solution of the system. Moreover, methods relying on this type of solvers require an explicit computation of the solution at every step of the algorithm, severely hindering the procedure in practice. A way to avoid direct numerical integration (or differentiation) is *smoothing* the data. This idea falls under the class of *collocation* methods, in which some of them are called *two-steps* [9, 15]. As in Brunel et al. [4], Madár et al. [20], Varah [27], a first step consist of recovering a temporary solution of the system by smoothing or interpolating the data (i.e. with cubic splines, least squares, nonparametric filters, local polynomial regression and so forth) and then applying nonlinear least squares to infer the parameters of the ODEs. The properties of these methods, such as consistency and asymptotic normality, are discussed in Xue et al. [30]. Other methods following a similar strategy of smoothing and matching are discussed in Campbell and Steele [6], González et al. [13], Liang and Wu [18]. Another approach is to use regularization [14, 24, 28], in order to do inference on the parameters while minimizing - with a frequentist flavor - some measure of distance between the theoretical solution and the estimated one. An initial guess of the parameters of interest is provided to the algorithm, as it is used together with a linear combination of basis functions to solve a penalized optimization problem. From a Bayesian point of view, as in Chkrebtii et al. [7], Gaussian Processes (GP) are prominent tools employed to solve the task. They encode naturally a source of randomness in the solution and simultaneously provide a class of flexible priors for the functions used to smooth the data coming from the ODEs' system. A recent approach using GP is provided by Calderhead et al. [5], with some drawbacks that were later addressed in Dondelinger et al. [10] through the use of adaptive gradient matching. Another advantage of these Bayesian methods is the complete probabilistic phrasing of the problem, allowing for a statistical quantification of the uncertainty about the solution obtained: the core of these procedures are - in fact - probabilistic solvers that can be sampled to explore the parameter space while obtaining indirectly a solution of

the system [8]. For some other (implicit and explicit) probabilistic solvers see Barber [1]; some applications of these methods on real life dataset are presented in Honkela et al. [16] and Titsias et al. [26].

In this work, we propose a two-step Bayesian strategy (Bayesian Smooth-and-Match) that borrows the idea of smoothing to overcome direct integration and, simultaneously, to filter some of the noise in the data. The first step of the method relies on penalized splines to smooth the data and reconstruct the variables of the ODEs; the second step focuses on inferring the parameters of the system through ridge regression, with covariates being known functions of the process that is being studied.

The rest of the manuscript is organized as follows: in Section 4.2, the notation used to build the strategy is introduced and we discuss the distributional assumptions on the data, together with the prior and posterior distributions; in Section 4.3, numerical and visual results on three different simulation studies are reported; in Section 4.3.4, an application to a previously analyzed dataset is presented; finally, in Section 4.4, we provide a summary of the work and we outline some future developments.

4.2 Model formulation

4.2.1 Tools and notation

Suppose we observe a p-dimensional vector $\boldsymbol{y}_i$ containing noisy observations of a dynamic process $\boldsymbol{x}(t_i)$, where t_i is a generic element from the set of ordered time points $\{t_i\} \in [0, 1]$, and $i = 1, \dots, n$ indexing the independent observed vectors. For the sake of simplicity, we will drop the "(t)" argument in the notation for $\boldsymbol{x}(t)$ and its components when not ambiguous, otherwise it will be explicit. We assume that:

$$y_{ik} \sim N\left(x_{ik}, \sigma_k^2\right) \tag{4.1}$$

with $k = 1, \dots, p$, $x_{ik} = x_k(t_i)$ and σ_k^2 a parameter describing the noise level in the data for the component k of the whole process $\boldsymbol{x}$. For every observation t_i, each component $\boldsymbol{x}_{\cdot k}$ can be approximated by a cubic spline [29, p. 122] of t in the time domain

$$x_{ik} = \sum_{h=1}^{q_k+2} \theta_{hk}\psi_{hk}(t_i) \tag{4.2}$$

where the sum is over $q_k + 2$ known basis functions $\psi_{hk}(\cdot)$ that depend on the q_k number of arbitrary knots used to construct them. For an arbitrary fixed t, we assume that the dynamic process $\boldsymbol{x}(t)$ is well described by an ordinary differential equations (ODEs) model defined as:

$$\begin{cases} \boldsymbol{x}'(t) = \frac{d\boldsymbol{x}(t)}{dt} = g(\boldsymbol{x}(t)) \\ \boldsymbol{x}(0) = \boldsymbol{\xi} \end{cases} \tag{4.3}$$

where $\boldsymbol{x}'(t)$ is the first derivative w.r.t. time of a continuous process $\boldsymbol{x}(t) = (\boldsymbol{x}_{\cdot 1}(t), \dots, \boldsymbol{x}_{\cdot k}(t), \dots, \boldsymbol{x}_{\cdot p}($ $\boldsymbol{\xi} = (\xi_1, \dots, \xi_k, \dots, \xi_p)$ is a vector of initial conditions for the system and $g : \mathbb{R}^p \to \mathbb{R}^p$ a (possibly) non-linear function of $\boldsymbol{x}(t)$. We focus on ODEs models that are linear in the parameters. The generic functional form of the derivative for the first variable

$\boldsymbol{x}_{\cdot 1}$ is

$$\frac{d\boldsymbol{x}_{\cdot 1}(t)}{dt} = g_1(\boldsymbol{x}(t)) = \sum_{j=1}^{b} \beta_j h_j(\boldsymbol{x}(t)) \tag{4.4}$$

which involves the first element $g_1 : \mathbb{R}^p \to \mathbb{R}$ of the function g. We think of it as a linear combination of b parameters of interest and some functions h_j, with $j = 1, \ldots, b$, that describe the dynamic evolution of the component $\boldsymbol{x}_{\cdot 1}$. Instead of working with the derivative of $\boldsymbol{x}$ we switch to the integral representation of the system in Equation 4.3:

$$\begin{aligned} \boldsymbol{x}_{i1} = \int_{t_1}^{t_i} \frac{d\boldsymbol{x}_{\cdot 1}(t)}{dt} &= \int_{t_1}^{t_i} g_1(\boldsymbol{x}(s))ds = \\ &= \xi_1 + \int_{t_1}^{t_i} \sum_{j=1}^{b} \beta_j h_j(\boldsymbol{x}(s))ds \end{aligned} \tag{4.5}$$

for $i = 2, \ldots, n$, because we set our first observation t_1 as the starting time point (t_1 could be either zero or not), and $\xi_1 = x_{11}$. The initial condition ξ_1 can be either estimated or assumed known. The solution for the first ODE, given by Equation 4.5, is thus:

$$\boldsymbol{x}_{i1} = \xi_1 + \sum_{j=1}^{b} \left(\beta_j \int_{t_1}^{t_i} h_j(\boldsymbol{x}(s))ds \right) = \xi_1 + \sum_{j=1}^{b} \beta_j H_j(\boldsymbol{x}(t_i)) \tag{4.6}$$

where H_j is the integral of the corresponding function h_j evaluated at time point t_i.

4.2.2 Prior, likelihood and posterior distributions

With reference to a sample of n observations, Equation (4.1) may be rewritten as:

$$\boldsymbol{y}_{\cdot k} = (y_{1k}, \ldots, y_{nk})^T \sim \mathcal{N}_n \left(\boldsymbol{x}_{\cdot k}, \sigma_k^2 \mathrm{I}_n \right)$$

with σ_k^2 the noise level for component k and I_n the identity matrix of order n. From Equation 4.2, $\boldsymbol{x}_{\cdot k} = \Psi_k \boldsymbol{\theta}_k$ where Ψ is the $n \times (q_k + 2)$ matrix of spline basis evaluated at every time point t_i. Thus, assuming that every component is independent from the other given the column vectors $\boldsymbol{\theta}_k$, the likelihood function of the model is

$$P(\boldsymbol{y}|\boldsymbol{\Theta}, \boldsymbol{\sigma}^2) = \prod_{i=1}^{n} \prod_{k=1}^{p} P(y_{ik}|\boldsymbol{\theta}_k, \sigma_k^2)$$

where $\boldsymbol{y} = (\boldsymbol{y}_{\cdot 1}, \ldots, \boldsymbol{y}_{\cdot p})$, $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_p\}$ and $\boldsymbol{\sigma}^2 = (\sigma_1^2, \ldots, \sigma_p^2)$. We choose to tackle the inferential procedure with a Bayesian approach: this allows us to represent the whole process with a fully probabilistic generative model that we can also describe as a graphical model. Furthermore, within the Bayesian framework, we can take into account the variability and the uncertainty at every level of the problem, from the smoothing of the data to the parameters governing the system. We thus assign prior probabilities to the parameters of the splines: we assume, for each $\boldsymbol{\theta}_k$, a

Gaussian prior distribution of the form

$$\boldsymbol{\theta}_k |\, \lambda_{\theta_k} \sim \mathcal{N}_{q_k+2}\left(\mathbf{0}, [\lambda_{\theta_k} \mathrm{S}_{\theta_k}]^{-1}\right).$$

Choosing this prior is equivalent to performing a penalized spline regression on the data: the form of the hyperparameter precision matrix S_{θ_k} defines the way we want the basis to be penalized. We follow this approach to ensure that non-linearity in the components is captured without, however, producing curves that would overfit the noisy data. The parameter λ_{θ_k} penalizes the non-smoothness of the functions ψ_{hk}: the 'wiggliness' of the curve resulting from the spline smoothing is encoded in the precision matrix S_{θ_k} (see Wood [29] for details, p. 126). These penalization terms have prior distributions

$$\lambda_{\theta_k} |\, \alpha_{\theta_k}, \gamma_{\theta_k} \sim \mathrm{Gam}(\alpha_{\theta_k}, \gamma_{\theta_k})$$

for some vectors of shape and rate hyperparameters $(\boldsymbol{\alpha}_\theta, \boldsymbol{\gamma}_\theta)$. These two-dimensional vectors are chosen to represent weakly informative priors on λ_{θ_k}: more specifically, we select values of the hyperparameters that encourage undersmoothing of the data [15], with enough variance for the Gamma distribution to be able to shift to a more penalized curve if needed. A reference improper prior density $P(\sigma_k^2) = 1/\sigma_k^2$ is employed for each σ_k^2. The first vector of observations $\boldsymbol{y}_{\cdot 1}$ has another representation, stemming directly from the ODEs' system solution for $\boldsymbol{x}_{\cdot 1}$ that depends on the integral solution from Equation 4.6, that is

$$\boldsymbol{y}^{\star}_{\cdot 1} |\, \xi_1, \boldsymbol{\beta}, \boldsymbol{\Theta}, \sigma_1^2 \sim \mathcal{N}_n\left(\xi_1 \mathbf{1}_n + \boldsymbol{H}\boldsymbol{\beta}, \sigma_1^2 \mathrm{I}_n\right)$$

where $\mathbf{1}_n$ is a n-dimensional unitary column vector, $\boldsymbol{\beta}$ the column vector of parameters we want to estimate and $\boldsymbol{H}$ a $n \times b$ matrix collecting the integrated functions H_j evaluated for each observation t_i. As in our case, when the starting inverse-problem may be 'ill-posed', sure enough the ordinary least squares estimation leads to overdetermined or underdetermined systems of equations as solution to the regression itself. Regularization is the usual approach to overcome this issue and Tikhonov regularization, in particular, is one of the most commonly used: from a Bayesian point of view, it is equivalent to assume, for $\boldsymbol{\beta}$, the following prior distribution

$$\boldsymbol{\beta} | \lambda_\beta \sim \mathcal{N}_b\left(\mathbf{0}, [\lambda_\beta \mathrm{I}_b]^{-1}\right).$$

The choice of this prior, effectively, induces the Bayesian 'ridge regression' with λ_β acting as a penalizing term for which we assume a prior distribution

$$\lambda_\beta | \alpha_\beta, \gamma_\beta \sim \mathrm{Gam}(\alpha_\beta, \gamma_\beta).$$

As for the other penalization terms, we follow the same approach of using a weakly informative prior. We also estimate ξ_1 instead of rescaling the data and we select a flat prior for it. Dropping hyperparameters from the notation, the joint posterior

distribution of the parameters in our model is:

$$\begin{aligned} & P(\boldsymbol{\Theta}, \xi_1, \boldsymbol{\beta}, \boldsymbol{\lambda}_\theta, \lambda_\beta, \boldsymbol{\sigma}^2 | \, \boldsymbol{y}, \boldsymbol{y}^\star_{\cdot 1}) \propto P\big(\boldsymbol{y}^\star_{\cdot 1} | \, \xi_1, \boldsymbol{\beta}, \boldsymbol{\Theta}, \sigma_1^2\big) \times \\ \times \quad & \prod_{k=1}^{p} \left[P\big(\boldsymbol{y}_{\cdot k} | \, \boldsymbol{\theta}_k, \sigma_k^2\big) \, P(\boldsymbol{\theta}_k | \, \lambda_{\theta_k}) \, P(\lambda_{\theta_k}) P(\sigma_k^2) \right] \times \\ \times \quad & P(\boldsymbol{\beta} | \, \lambda_\beta) \, P(\lambda_\beta) \, P(\xi_1) \end{aligned} \tag{4.7}$$

which is represented in the graphical model in Figure 4.1.

4.2.3 Mimicking the data: relationship with other methods

One feature of this representation is that the vector of observations $\boldsymbol{y}_{\cdot 1}$ appears twice: as a term of the likelihood component, when $k = 1$ in the first product, and then as the noisy solution of the ODEs' system, that is $\boldsymbol{y}^\star_{\cdot 1}$. In Barber and Wang [2], also their GPODE model focuses on a probabilistic generative model for the data and the graph representation contains two nodes for the same quantity (namely, the process itself). The authors model the system $\boldsymbol{x}(t)$ as coming from a Gaussian process (GP) and exploit the fact that differentiating the GP produces derivatives $\boldsymbol{x}'(t)$ that are still modeled as a Gaussian process with available analytical description of its kernel (see the original manuscript for more details). Then, they marginalize over the components $\boldsymbol{x}(t)$ with a standard convolution integral and model the data $\boldsymbol{y}$ with a Gaussian distribution; after that, they reintroduce $\boldsymbol{x}(t)$ and couple it with the obtained derivatives to measure the distance between the deterministic ODEs of the system and the ones estimated by the data. This approach faces some issues. As pointed out in Macdonald et al. [19], where the authors inspect the graph representation in Barber and Wang [2], having two nodes assigned to the same quantity is methodologically inconsistent. To solve the issue, they first introduce a dummy variable that mimics $\boldsymbol{x}(t)$, thus removing the inconsistency. However, the two nodes are still conceptually describing the very same quantity and a natural definition of this dependency would be an undirected edge between them: this addition, unfortunately, changes the graph from a directed acyclic graph to a chain graph which is not a probabilistic generative model anymore. To preserve its nature, they keep the two nodes separate form one another but highlight the consequences of this choice: when some of the noisy vectors $\boldsymbol{y}_{\cdot k}$ are not available (that is the case of partially observed systems), the model itself might be unidentifiable because of the likelihood not depending anymore on the parameters of the ODEs after marginalizing over the unobserved quantities. As we face the same issue with our proposed strategy, we are thus limited to situations where all the components of the process $\boldsymbol{x}(t)$ are observed (with noise). Given that a probabilistic definition of what could be the directed edge between $\boldsymbol{y}_{\cdot 1}$ and $\boldsymbol{y}^\star_{\cdot 1}$ (or viceversa) is not obvious, we couple the two quantities only by assuming they share the same variance σ_1^2 (as described in Figure 4.1). This shared nuisance parameter, however, suffers from the independence assumption on $\boldsymbol{y}_{\cdot 1}$ and $\boldsymbol{y}^\star_{\cdot 1}$. Suppose the simplest case where $\boldsymbol{y}^\star_{\cdot 1} = \boldsymbol{y}_{\cdot 1} \sim \mathcal{N}(\boldsymbol{0}, \sigma_1^2 I_n)$: having no edge between the two nodes is equivalent to $P(\boldsymbol{y}^\star_{\cdot 1}, \boldsymbol{y}_{\cdot 1} | \sigma_1^2) = P(\boldsymbol{y}^\star_{\cdot 1} | \sigma_1^2) P(\boldsymbol{y}_{\cdot 1} | \sigma_1^2)$ which is roughly equal to a Normal distribution with variance parameter $\frac{\sigma_1^2}{2}$. Obviously, this would not happen if a directed edge were to be added to the graph, as the joint distribution of the vector and its copy

would be instead $P(\boldsymbol{y}^{\star}_{\cdot 1}, \boldsymbol{y}_{\cdot 1}|\sigma_1^2) = P(\boldsymbol{y}^{\star}_{\cdot 1}|\boldsymbol{y}_{\cdot 1}, \sigma_1^2)P(\boldsymbol{y}_{\cdot 1}|\sigma_1^2) = 1 \cdot P(\boldsymbol{y}_{\cdot 1}|\sigma_1^2)$, but we already discussed there is no clear way to describe a directed edge between the two nodes. This inconvenience is mostly present in a simulation environment. We start with a deterministic solution of an ODEs' system, $\boldsymbol{x}_{\cdot 1}$, and we perturb it with some noise $\tilde{\sigma}^2$ thus obtaining an observed vector $\boldsymbol{y}_{\cdot 1}$: recovering the noise level in the data is not equivalent anymore to estimating $\tilde{\sigma}^2$, as in our model we are bringing together in σ_1^2 two sources of noise - albeit coming from the very same vector of observations. In fact, σ_1^2 bears both the uncertainty about how good the solution of the set of ODEs and also how good is the smoothing of the first vector of data that we used as a regressor in solving indirectly the system. Such an undesired 'mismatch' effect should not present itself as a problem when dealing with real world observations.

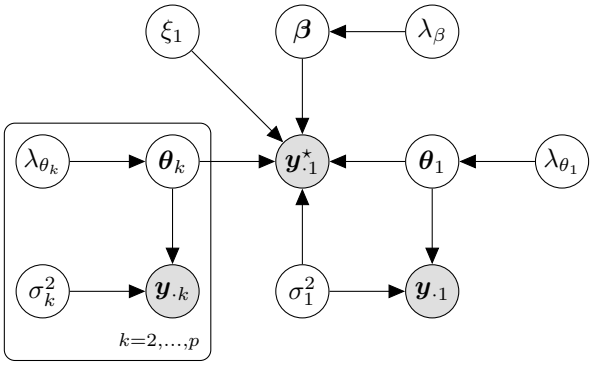

Figure 4.1: Graphical representation of the ODEs and solution model

4.2.4 Bayesian Smooth-and-Match

Following the standard derivations for the full conditionals from Equation 4.7, the samplers for $\boldsymbol{\theta}_k$ should consider the quantities

$$\begin{aligned} P(\boldsymbol{\theta}_k|\sigma_k^2, \lambda_{\theta_k}, \boldsymbol{y}_{\cdot k}) \quad &\propto \quad P\big(\boldsymbol{y}_{\cdot k}|\, \boldsymbol{\theta}_k, \sigma_k^2\big)\, P(\boldsymbol{\theta}_k|\, \lambda_{\theta_k}) \times \\ &\times \quad P\big(\boldsymbol{y}^{\star}_{\cdot 1}|\, \xi_1, \boldsymbol{\beta}, \boldsymbol{\Theta}, \sigma_1^2\big) \end{aligned} \tag{4.8}$$

which demands for a Metropolis-within-MCMC. After some mathematical steps, it is possible to show that, in the previous posterior distribution, the full conditionals of the parameters in $\boldsymbol{\Theta}$ are not in closed form. Our primary focus, however, is on the estimation of the $\boldsymbol{\beta}$ vector that contains the parameters truly describing the dynamic evolution of the ODEs' system. The spline smoothing step (Equation 4.2) is, instead, just a convenient approach to build the regression matrix $\tilde{H}$ in Equation 4.6. Moreover, an undesirable loop-feedback effect from $\boldsymbol{\beta}$ to $\boldsymbol{\Theta}$ (and viceversa) may arise if we aim for the 'true' full conditional obtained from Equation 4.8. So, in order to have a more stable and faster MCMC scheme, we adopt a Bayesian Smooth-and-Match strategy. The procedure consists of two steps:

- first (*smooth* step), we do Bayesian penalized spline smoothing to recover the $\boldsymbol{x}_{\cdot k}$'s through a Gibbs sampler that we obtain for $\boldsymbol{\theta}_k$ by temporarily not considering the integral solution term;
- second (*match* step), we plug-in the $\boldsymbol{x}_{\cdot k}$'s, computed with the sampled values for $\boldsymbol{\theta}_k$ from the previous step, into the sampler for $\boldsymbol{\beta}$ through the matrix $\tilde{H}$.

In practice, when sampling each $\boldsymbol{\theta}_k$, this allows us to drop the term $P(\boldsymbol{y}^{\star}_{\cdot 1} | \, \xi_1, \boldsymbol{\beta}, \boldsymbol{\Theta}, \sigma_1^2)$. A more sophisticated smoothing could be performed (see Morrissey et al. [22] for a state-of-art Bayesian spline regression) but we consider a simpler approach because we are only interested, at the first step, in recovering the components of the process to be used as regressors later at the second step of the procedure. Notice that, from a frequentist point of view, a consistent estimator of the ODEs' parameters can be obtained under mild conditions (namely, on the penalizing terms) when following this plug-in approach [15]. The two steps of the strategy are not completely standalone compartments: as previously said, the quantity σ_1^2 connects both parts and acts as a measure for the uncertainty (from the smoothing and the regression) of the solution indirectly obtained by doing inference on $\boldsymbol{\beta}$. The other parameters of the model are all updated with Gibbs samplers that follow from standard derivations for conjugated priors and likelihood terms.

4.3 Simulation studies

In this Section, we validate our proposed strategy by testing it with three different ODEs' systems, starting from a simple one component model (logistic growth) up to a three components epidemic model (HIV viral fitness). We simulate nine scenarios for each ODEs model, exploring three different level of noise and three sample sizes ($n = 25, n = 100, n = 500$). The level of contamination of the data (low, medium or high), with Gaussian noise, is quantified through signal-to-noise ratio (SNR), that is the ratio between the standard deviation of the deterministic simulated solution (signal) and the standard deviation of the error term (noise) we use to perturb it. An increase in the strength of noise is equivalent to a decrease of the associated SNR, as the standard deviation of the signal at the numerator is fixed for a given sample size. For comparative purposes, we also inspect our Bayesian Smooth-and-Match strategy (*SnM*) together with the collocation method implemented in the R package *CollocInfer* [24]. For each scenario, we run both algorithms on 100 independently simulated datasets and we summarize the results as the average across these replicates. We quantify the uncertainty about the estimated parameters in the same way for both algorithms by computing the associated mean square error (MSE) of these one hundred values. The tuning parameters for the collocation method, such as the number of knots, order of the basis for the splines and the penalization term, are selected with the functions provided in the package. The collocation method also needs initial guesses for the regression coefficients and we provide starting points drawn randomly from uniform distributions over ranges $[\boldsymbol{\beta} - 4\boldsymbol{\beta}; \boldsymbol{\beta} + 4\boldsymbol{\beta}]$; when the lower bound of the range is negative for parameters that only exist as positive we set it to a small value, close to zero. Given the sensibility of *CollocInfer* to the provided starting points, we control if convergence is achieved by the algorithm and we discard datasets that result in degenerate estimates for the parameters by checking the associated likelihood in the output. Those datasets are not considered when computing the averages and a measure of the number of actual datasets (NAD) used is provided. As for the visual representation of the results, Figures 4.2 to 4.4 show the plot mosaics with the results for one randomly drawn dataset of the simulated one hundred.

4.3.1 Logistic population growth

We first focus on a simple ODEs' system. We simulate data coming from the logistic growth model [21], frequently employed in ecology and biology to describe the growth dynamics of a certain population. The system is defined as:

$$\frac{d\boldsymbol{x}_{\cdot 1}}{dt} = a\boldsymbol{x}_{\cdot 1}\left(1 - \frac{\boldsymbol{x}_{\cdot 1}}{K}\right),$$

where a is the growth rate and K the carrying capacity of the population involved. We consider another representation of previous equation, that is, in our notation,

$$\frac{d\boldsymbol{x}_{\cdot 1}}{dt} = \beta_1\boldsymbol{x}_{\cdot 1} + \beta_2\boldsymbol{x}_{\cdot 1}^2,$$

linear in the parameters ($\beta_1 = a, \beta_2 = -a/K$) we want to estimate, together with $x_{11} = \xi_1$. We simulate a noise-free solution of the system and then we perturb it with Gaussian errors at different level of noise (low, medium, high); the three levels are obtained by setting the standard deviation of the errors as increasing proportions of the mean value of $\boldsymbol{x}_{\cdot 1}$. The number of knots for the smoothing step, $q_1 = 2$, is manually selected, aiming for some undersmoothing of $\boldsymbol{y}_{\cdot 1}$ [15]. The functions used to build the regression matrix are $h_1(\boldsymbol{x}) = \boldsymbol{x}_{\cdot 1}$ and $h_2(\boldsymbol{x}) = \boldsymbol{x}_{\cdot 1}^2$; the true values for ξ_1, β_1 and β_2 used in the simulations are reported in Table 4.1. As expected, we can see from Table 4.1 that to increasing levels of contamination of the data with noise correspond higher (on average) mean square errors for both methods. The average posterior mean of ξ_1 is stable throughout all the scenarios, showing some bias only when the SNR is at the highest level; *CollocInfer* does not provide an estimate for the initial condition ξ_1. With *SnM* we can recover the first parameter β_1 in almost any scenario with appreciable quality, also showing better results in comparison with *CollocInfer*; when the noise contamination is at its maximum, however, the MSE computed on the one hundred posterior means is noticeably higher. The algorithm *CollocInfer* seems to be more stable when retrieving the second parameter β_2, providing optimal estimates when the sample size is $n = 100$ and the noise contamination up to a medium level ($SNR = 13$ and $SNR = 6.5$). The solution uncertainty quantified by *SnM*, σ_1^2, appears to be less sensible to changes in sample size and more to the signal-to-noise ratio. In every scenario, the number of actual datasets (NAD) used to compute the results for *CollocInfer* is less than 100, meaning that degenerate solutions were discarded in the process. A visual description of the results is presented in Figure 4.2: in most of the plots, the line describing the true curve (*solid* line), the smoothed version of $\boldsymbol{y}_{\cdot 1}$ (*dotted* line) and the ODE regression solution (*long-dashed*) are undistinguishable from each other. They start to become appreciably different in the right part of the plots mosaic, showing the scenarios with the highest level of noise. We compute the average MSE between each curve (dotted line, MSE_{x_1}, and long-dashed line, MSE_{g_1}) and the true one representing the unperturbed data. For $n = 500$ and $SNR = 1.3$, the two mean square errors are $\text{MSE}_{x_1} = 0.022$, for the smoothed reconstruction, and $\text{MSE}_{x_1} = 0.007$ for the ODEs' system solution; with the same sample size but less noise, SNR=13, the difference between the two curves and the true one is almost negligible ($\text{MSE}_{x_1} = 0.00015$ and

$\text{MSE}_{g_1} = 0.00016$).

Table 4.1: Average posterior means (with *MSE* in brackets) for the parameters of the logistic population growth model; three sample sizes and increasing noise level

Sample size	Parameters	Noise level					
		low ($SNR = 13$)		medium ($SNR = 6.5$)		high ($SNR = 1.3$)	
		SnM	CollocInfer	SnM	CollocInfer	SnM	CollocInfer
$n = 25$	$\xi_1 = 0.1$	0.098 (*0.001*)	-	0.096 (*0.002*)	-	0.078 (*0.043*)	-
	$\beta_1 = 2.5$	2.531 (*0.105*)	3.819 (*6.845*)	2.560 (*0.412*)	5.054 (*14.530*)	2.317 (*4.687*)	4.727 (*12.370*)
	$\beta_2 = -0.125$	-0.180 (*0.271*)	-0.244 (*0.091*)	-0.216 (*1.035*)	-0.420 (*0.228*)	-0.359 (*9.864*)	-0.385 (*0.215*)
	$^{\diamond}\sigma_1^2$	0.064	-	0.244	-	7.054	-
	NAD	100	98	100	96	100	93
$n = 100$	$\xi_1 = 0.1$	0.098 (*0.001*)	-	0.096 (*0.002*)	-	0.078 (*0.046*)	-
	$\beta_1 = 2.5$	2.540 (*0.092*)	5.080 (*15.480*)	2.571 (*0.364*)	5.060 (*15.267*)	2.904 (*9.403*)	5.017 (*14.886*)
	$\beta_2 = -0.125$	-0.193 (*0.213*)	-0.124 (*0.009*)	-0.238 (*0.836*)	-0.124 (*0.009*)	-0.740 (*20.655*)	-0.130 (*0.011*)
	$^{\diamond}\sigma_1^2$	0.065	-	0.245	-	6.364	-
	NAD	100	92	100	92	100	94
$n = 500$	$\xi_1 = 0.1$	0.098 (*0.001*)	-	0.095 (*0.002*)	-	0.077 (*0.048*)	-
	$\beta_1 = 2.5$	2.542 (*0.086*)	2.633 (*1.840*)	2.573 (*0.342*)	4.336 (*6.462*)	2.852 (*8.649*)	5.441 (*14.641*)
	$\beta_2 = -0.125$	-0.197 (*0.188*)	-0.091 (*0.046*)	-0.242 (*0.742*)	-0.103 (*0.003*)	-0.662 (*18.647*)	-0.095 (*0.008*)
	$^{\diamond}\sigma_1^2$	0.066	-	0.248	-	6.126	-
	NAD	100	83	100	80	100	89

$\diamond$ results reported as multiplied by 10^2

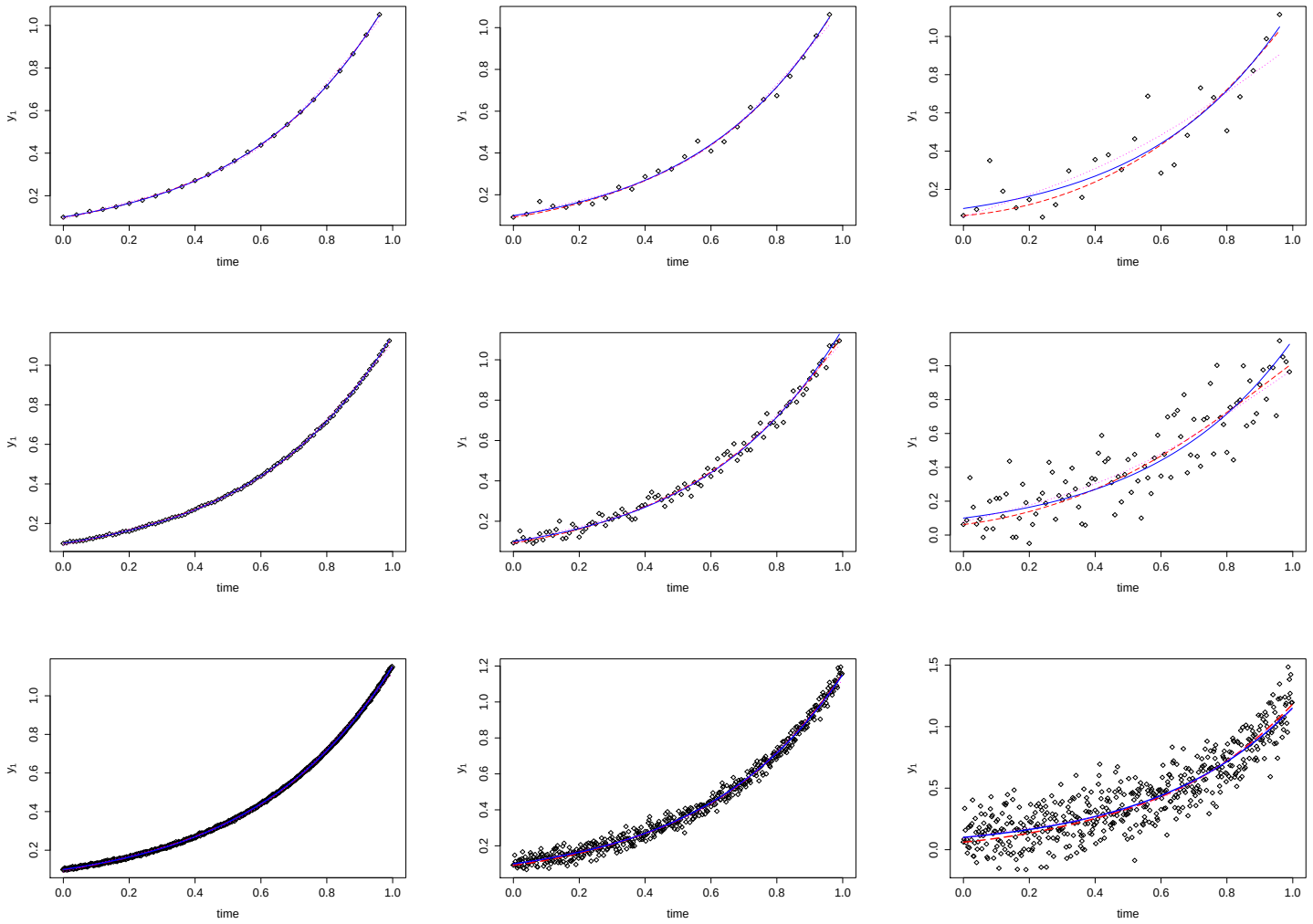

Figure 4.2: Observed noisy data (*dots*), smoothing spline (*dotted* line), true solution of the ODEs' system (*solid* line) and reconstructed solution (*long-dashed* line) for the first variable $\boldsymbol{x}_{\cdot 1}$. Different sample sizes ($n = 25, 100, 500$) from the top to the bottom and noise levels (low, medium and high) from left to right.

4.3.2 Lotka-Volterra

In the second batch of simulations we consider the Lotka-Volterra system [11]. This system of ODEs is used to model the dynamics, with respect to time, of two competing groups categorizable as *preys* and *predators*; setting some of the parameters of the ODEs to zero or imposing constraints, however, produces systems that are also used to characterize epidemic processes. The model is described by the following set of equations:

$$\begin{cases} \boldsymbol{x}'_{\cdot 1} = \beta_1 \boldsymbol{x}_{\cdot 1} + \beta_2 \boldsymbol{x}_{\cdot 1} \boldsymbol{x}_{\cdot 2} \\ \boldsymbol{x}'_{\cdot 2} = \beta_3 \boldsymbol{x}_{\cdot 2} + \beta_4 \boldsymbol{x}_{\cdot 1} \boldsymbol{x}_{\cdot 2} \\ x_{11} = \xi_1 \\ x_{12} = \xi_2. \end{cases} \tag{4.9}$$

We focus the inference procedure on the first ordinary differential equation of the system and thus on the subset of parameters $(\xi_1, \beta_1, \beta_2)$. We explore nine scenarios as in the logistic growth case. We use the same approach to identify the three levels of noise but we select the sample sizes differently: the first one ($n = 25$) reproduces a situation where we have 25 evenly-spaced time points starting from $t_1 = 0$ up to $t_{25} = 24$; for $n = 100$, t ranges from $t_1 = 0$ to $t_{100} = 99$ with unitary step size; the last one ($n = 500$) has the same time range as $n = 100$, with $t_1 = 0$ and $t_{500} = 99$, but a denser sampling grid given by 0.2 as the step size. Going from $n = 100$ to $n = 500$ encompass a situation where the time range is fixed (the maximum observational time point) but the amount of data increases (more observations in the same timeframe). We use as regressing functions the quantities $h_1(\boldsymbol{x}) = \boldsymbol{x}_{\cdot 1}$ and $h_2(\boldsymbol{x}) = \boldsymbol{x}_{\cdot 1}\boldsymbol{x}_{\cdot 2}$; we choose the same number of knots, $q_1 = q_2 = 5$, for both splines. In Table 4.2, averages of the posterior means and corresponding mean square errors are reported for all the scenarios. We see that both algorithms perform well when the sample size is $n = 25$, regardless of the three noise level (SNR=50,5,2.5). The number of actual datasets used to compute averages for *CollocInfer* also shows that convergence was achieved for all the first three scenarios. When evaluating the performance of the two methods, for $n = 100$, we notice that *SnM* performs slightly better in terms of bias of the estimated parameters β_1 and β_2; also, *CollocInfer* shows slightly higher MSEs for the second estimated parameter with respect to *SnM*. This difference is more prominent when $n = 500$. As expected, the quantification of the uncertainty about the solution provided by *SnM*, through σ_1^2, decreases as the sample size grows. A visual appreciation of the results is given in Figure 4.3. The most different performance between the smoothing and the regression curves is evident in the case of $n = 25$ and $SNR = 2.5$ (upper row of the mosaic, rightmost plot): in this case, the mean square errors of the dotted line and the long-dashed line are respectively $\text{MSE}_{x_1} = 0.690$ and $\text{MSE}_{g_1} = 0.252$. Such a discrepancy in the (average) accuracy of the reconstructed curves fades as the sample size grows. For $n = 500$ and $SNR = 50$ (the lowest noise level), the two errors are $\text{MSE}_{x_1} = 0.094$ and $\text{MSE}_{g_1} = 0.084$.

Table 4.2: Average posterior means (with *MSE* in brackets) for the parameters of the Lotka-Volterra ODEs' system; three sample sizes and increasing noise level

Sample size	Parameters	Noise level					
		low ($SNR = 50$)		medium ($SNR = 5$)		high ($SNR = 2.5$)	
		SnM	CollocInfer	SnM	CollocInfer	SnM	CollocInfer
$n = 25$	$\xi_1 = 2.0$	1.996 (*0.001*)	-	1.965 (*0.112*)	-	1.929 (*0.448*)	-
	$\beta_1 = 0.1$	0.101 (*0.001*)	0.095 (*0.001*)	0.101 (*0.001*)	0.093 (*0.001*)	0.091 (*0.001*)	0.863 (*0.001*)
	$\beta_2 = -0.2$	-0.201 (*0.001*)	-0.197 (*0.001*)	-0.197 (*0.001*)	-0.192 (*0.002*)	-0.170 (*0.002*)	-0.183 (*0.008*)
	$^{\diamond}\sigma_1^2$	0.040	-	2.295	-	12.234	-
	NAD	100	100	100	100	100	100
$n = 100$	$\xi_1 = 2.0$	1.996 (*0.001*)	-	1.965 (*0.112*)	-	1.930 (*0.449*)	-
	$\beta_1 = 0.1$	0.089 (*0.001*)	0.063 (*0.001*)	0.088 (*0.001*)	0.060 (*0.002*)	0.086 (*0.001*)	0.061 (*0.002*)
	$\beta_2 = -0.2$	-0.168 (*0.001*)	-0.140 (*0.004*)	-0.167 (*0.001*)	-0.139 (*0.004*)	-0.162 (*0.002*)	-0.138 (*0.004*)
	$^{\diamond}\sigma_1^2$	3.650	-	5.392	-	10.507	-
	NAD	100	100	100	100	100	98
$n = 500$	$\xi_1 = 2.0$	1.996 (*0.001*)	-	1.965 (*0.112*)	-	1.930 (*0.447*)	-
	$\beta_1 = 0.1$	0.095 (*0.001*)	0.080 (*0.001*)	0.095 (*0.001*)	0.080 (*0.001*)	0.094 (*0.002*)	0.077 (*0.001*)
	$\beta_2 = -0.2$	-0.182 (*0.001*)	-0.138 (*0.004*)	-0.182 (*0.001*)	-0.138 (*0.004*)	-0.180 (*0.002*)	-0.138 (*0.004*)
	$^{\diamond}\sigma_1^2$	1.405	-	2.89	-	7.407	-
	NAD	100	100	100	100	100	99

$\diamond$ results reported as multiplied by 10^1

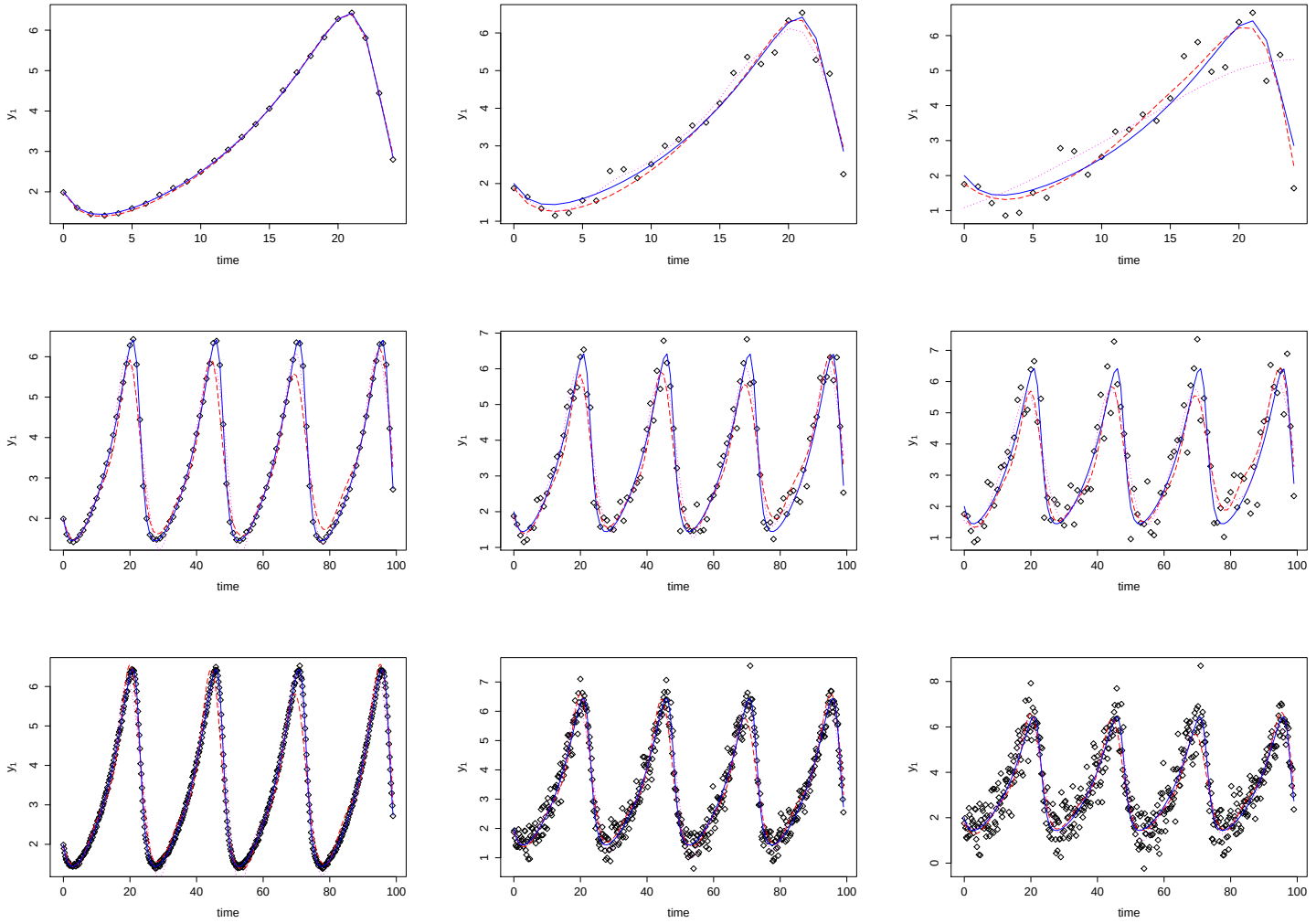

Figure 4.3: Observed noisy data (*dots*), smoothing spline (*dotted* line), true solution of the ODEs' system (*solid* line) and reconstructed solution (*long-dashed* line) for the first variable $\boldsymbol{x}_{\cdot 1}$. Different sample sizes ($n = 25, 100, 500$) from the top to the bottom and noise levels (low, medium and high) from left to right.

4.3.3 HIV viral fitness

Our last batch of data is simulated from a set of ODEs modelling the dynamics of HIV virus [3]. The system is defined as:

$$\begin{cases} \boldsymbol{x}'_{\cdot 1} = \beta_1 + \beta_2 \boldsymbol{x}_{\cdot 1} + \beta_3 \boldsymbol{x}_{\cdot 1} \boldsymbol{x}_{\cdot 3} \\ \boldsymbol{x}'_{\cdot 2} = \beta_3 \boldsymbol{x}_{\cdot 1} \boldsymbol{x}_{\cdot 3} - 0.5 \boldsymbol{x}_{\cdot 2} \\ \boldsymbol{x}'_{\cdot 3} = 0.5 \cdot \beta_4 \boldsymbol{x}_{\cdot 2} + \beta_5 \boldsymbol{x}_{\cdot 3} \\ x_{11} = \xi_1 \\ x_{12} = \xi_2 \\ x_{13} = \xi_3. \end{cases} \tag{4.10}$$

where we focus on the first ODE and on the subset of parameters $(\xi_1, \beta_1, \beta_2, \beta_3)$. The simulation setting is the same as in Section 4.3.2: three levels of noise and three sample sizes where only the last has an effective increase in the number of observations. True parameters' values are tuned according to the ones used in Vujačić et al. [28]. Number of knots selected is the same for all splines $q_1 = q_2 = q_3 = 20$. We use the following regressing functions: $h_1(\boldsymbol{x}) = 1$, $h_2(\boldsymbol{x}) = \boldsymbol{x}_{\cdot 1}$ and $h_3(\boldsymbol{x}) = \boldsymbol{x}_{\cdot 1} \boldsymbol{x}_{\cdot 3}$. We report the results in Table 4.3 and Figure 4.4. The parameter β_2 proved to be difficult to estimate with *CollocInfer* so we decided to fix it to the true value for this algorithm while estimating it in the case of *SnM*. When $n = 25$ we notice that our method breaks down at the highest level of noise contamination (SNR=1.3): set aside the recovered initial condition, which shows good average posterior mean, the estimates for the other parameters are noticeably biased and have large mean square errors (especially in the case of β_1). Even if the NAD for *CollocInfer* is around 85 for the first three scenarios, the algorithm seems to be more robust in this setting. When considering β_1 and β_3, *CollocInfer* achieves lower mean square errors in almost every scenario with the exception of the setting with $n = 500$ and SNR=15, where the MSE for β_1 obtained with *SnM* is slightly lower than the one computed for *CollocInfer*. The number of discarded datasets gets lower as the sample size grows, as expected. For $n = 100$ and $n = 500$ we do not observe the same degrading effect on our algorithm *SnM* even at the highest level of noise, meaning that the information from the bigger sample size is enough to overcome the contamination in the data. As we can see from the mosaic plot in Figure 4.4, for $n = 100$ and $n = 500$ there is an appreciable difference between the two curves with respect to the true one, especially on the right-half portion of each plot. The average mean square errors for the setting with $n = 100$ and SNR=1.3 are $\text{MSE}_{g_1} = 16.179$ and $\text{MSE}_{x_1} = 11.743$, meaning that the smoothing step provides a reconstructed curve slightly closer to the true one (dotted line versus solid line). The opposite happens when the sample size gets to 500, as in the setting with SNR=2.5, the two MSEs are and $\text{MSE}_{g_1} = 1.397$ and $\text{MSE}_{x_1} = 5.859$. As for the setting in which the model breaks down ($n = 25$, SNR=1.3), the average mean square errors are $\text{MSE}_{g_1} = 80.467$ and $\text{MSE}_{x_1} = 104.969$, values much higher than the ones computed on the other eight scenarios (not reported here for brevity).

Table 4.3: Average posterior means (with *MSE* in brackets) for the parameters of the HIV viral fitness ODEs' system; three sample sizes and increasing noise level

Sample size	Parameters	Noise level					
		low ($SNR = 15$)		medium ($SNR = 2.5$)		high ($SNR = 1.3$)	
		SnM	CollocInfer	SnM	CollocInfer	SnM	CollocInfer
$n = 25$	$\xi_1 = 60$	59.879 (*1.35*)	-	59.519 (*21.55*)	-	59.031 (*86.12*)	-
	$\beta_1 = 20$	20.808 (*2.79*)	20.933 (*1.001*)	20.294 (*21.89*)	21.142 (*7.130*)	14.609 (*175.49*)	20.849 (*8.869*)
	$\beta_2 = -0.108$	-0.113 (*0.001*)	-	-0.103 (*0.005*)	-	-0.030 (*0.032*)	-
	$^{\dagger}\beta_3 = -0.095$	-0.106 (*0.001*)	-0.106 (*0.003*)	-0.109 (*0.001*)	-0.109 (*0.001*)	-0.159 (*0.001*)	-0.106 (*0.001*)
	$^{\diamond}\sigma_1^2$	0.169	-	2.346	-	20.398	-
	NAD	100	84	100	85	100	85
$n = 100$	$\xi_1 = 60$	59.882 (*1.18*)	-	59.539 (*18.82*)	-	59.079 (*75.26*)	-
	$\beta_1 = 20$	21.714 (*3.13*)	19.700 (*1.936*)	21.457 (*4.98*)	19.634 (*2.188*)	20.281 (*9.94*)	19.772 (*3.728*)
	$\beta_2 = -0.108$	-0.117 (*0.001*)	-	-0.115 (*0.001*)	-	-0.106 (*0.001*)	-
	$^{\dagger}\beta_3 = -0.095$	-0.103 (*0.001*)	-0.090 (*0.001*)	-0.103 (*0.001*)	-0.089 (*0.010*)	-0.100 (*0.001*)	-0.090 (*0.009*)
	$^{\diamond}\sigma_1^2$	0.640	-	2.748	-	9.473	-
	NAD	100	98	100	98	100	97
$n = 500$	$\xi_1 = 60$	59.882 (*1.18*)	-	59.538 (*18.84*)	-	59.078 (*75.35*)	-
	$\beta_1 = 20$	21.012 (*1.16*)	19.612 (*1.791*)	21.040 (*3.17*)	19.648 (*1.412*)	20.936 (*8.99*)	19.499 (*2.437*)
	$\beta_2 = -0.108$	-0.114 (*0.001*)	-	-0.113 (*0.001*)	-	-0.112 (*0.001*)	-
	$^{\dagger}\beta_3 = -0.095$	-0.100 (*0.001*)	-0.088 (*0.001*)	-0.100 (*0.001*)	-0.089 (*0.001*)	-0.101 (*0.001*)	-0.087 (*0.001*)
	$^{\diamond}\sigma_1^2$	0.437	-	2.312	-	8.326	-
	NAD	100	99	100	99	100	100

† results reported as multiplied by 10^2

◇ results reported as multiplied by $10^{(-1)}$

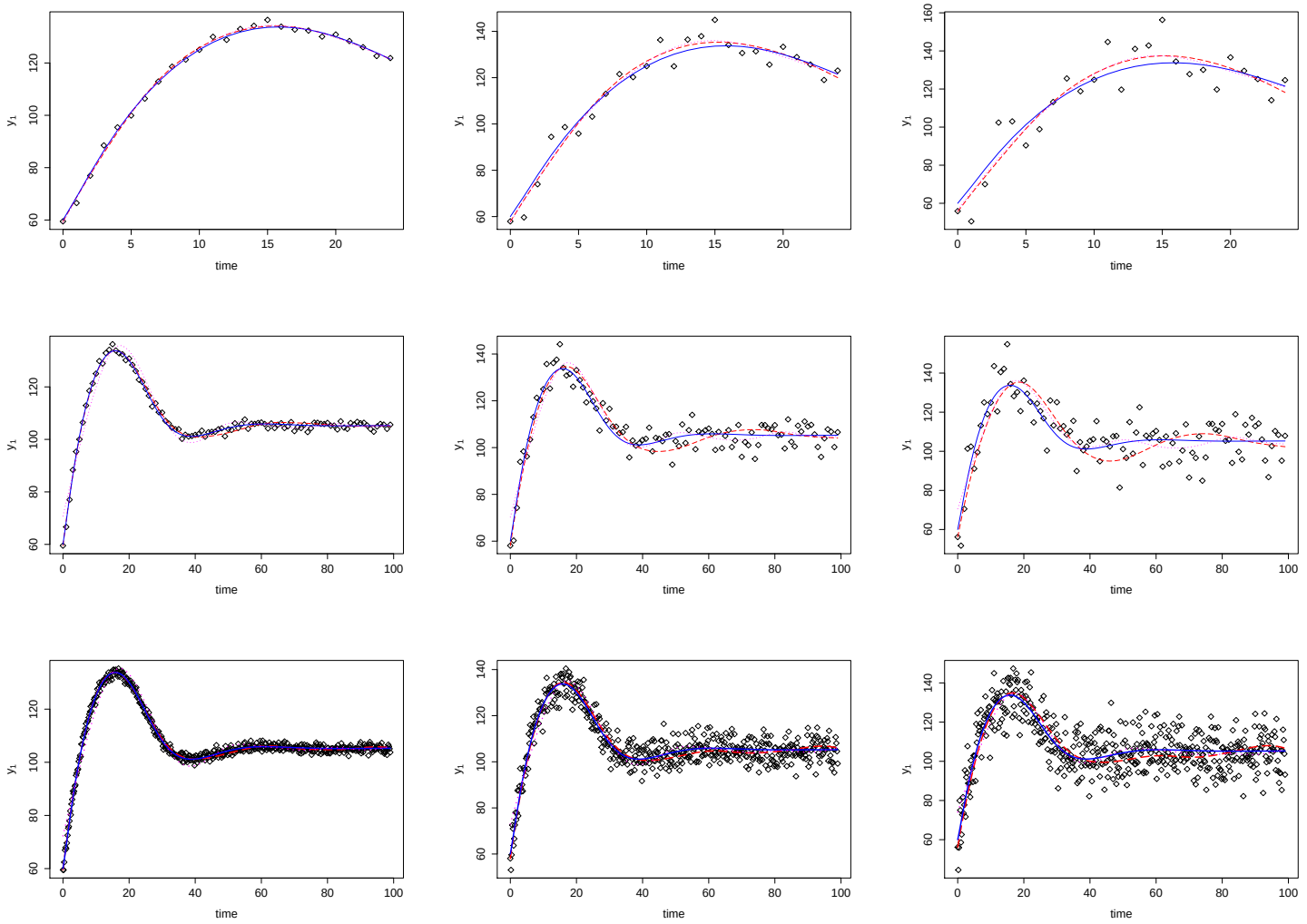

Figure 4.4: Observed noisy data (*dots*), smoothing spline (*dotted* line), true solution of the ODEs' system (*solid* line) and reconstructed solution (*long-dashed* line) for the first variable $\boldsymbol{x}_{\cdot 1}$. Different sample sizes ($n = 25, 100, 500$) from the top to the bottom and noise levels (low, medium and high) from left to right.

4.3.4 FitzHugh-Nagumo system for neuron electrical activity

We analyze a toy-data example available from the package *CollocInfer* [17]. The data (*FhNdata*) consist of 41 evenly-spaced observations in the timeframe $[0, 20]$ from the following ODEs model

$$\begin{cases} \boldsymbol{x}'_{\cdot 1} = c(\boldsymbol{x}_{\cdot 1} - \boldsymbol{x}^3_{\cdot 1}/3 + \boldsymbol{x}_{\cdot 2}) \\ \boldsymbol{x}'_{\cdot 2} = -\frac{1}{c}(\boldsymbol{x}_{\cdot 1} - a + b\boldsymbol{x}_{\cdot 2}) \\ x_{11} = \xi_1 \\ x_{12} = \xi_2 \end{cases} \tag{4.11}$$

known as the FitzHugh-Nagumo system [12, 23]. The set of equations describe the pulse transmission for neuronal activity. The parameters' values used to generate the data are $a = 0.2, d = 0.2, c = 3, \xi_1 = 0.5$; the simulated values are then perturbed with variances equal to 0.25 for both the variables $\boldsymbol{x}_{\cdot 1}$ and $\boldsymbol{x}_{\cdot 2}$. As it is written in Equation 4.11, the system is not linear in the parameters. We thus consider another representation

$$\begin{cases} \boldsymbol{x}'_{\cdot 1} = \beta_4(\boldsymbol{x}_{\cdot 1} - \boldsymbol{x}^3_{\cdot 1}/3 + \boldsymbol{x}_{\cdot 2}) \\ \boldsymbol{x}'_{\cdot 2} = \beta_1\boldsymbol{x}_{\cdot 1} + \beta_2 + \beta_3\boldsymbol{x}_{\cdot 2} \\ x_{11} = \xi_1 \\ x_{12} = \xi_2 \end{cases} \tag{4.12}$$

and we focus on the second differential equation and the subset of parameters $(\xi_2, \beta_1 = -1/c, \beta_2 = a/c, \beta_3 = -d/c)$. The ODE solution regression uses the following functions: $h_1(\boldsymbol{x}) = \boldsymbol{x}_{\cdot 1}$, $h_2(\boldsymbol{x}) = 1$ and $h_3(\boldsymbol{x}) = \boldsymbol{x}_{\cdot 2}$. We compare our results with the point estimates obtained by Vujačić et al. [28], remarking the fact that we only do inference on one of the two differential equations whereas they simultaneously estimate all the parameters of the model using information from both variables. In Figure 4.5, reconstructed curves from both steps of the procedure (*smooth* and *match*) are plotted: we can see, in the right plot, the penalized spline (*dotted* line) captures a biased initial condition in comparison to the ODE solution regression (*long-dashed* line), even if the associated mean square errors for the two curves, respectively $\text{MSE}_{g_1} = 0.042$ and $\text{MSE}_{x_1} = 0.033$, are actually close. The degree of smoothing for the two curves is also practically the same. The smoothing for the first variable (left plot of Figure 4.5) is satisfactory. As for the other parameters, we report their posterior means in Table 4.4: the algorithm recovers the true value for β_1 with appreciable accuracy; for β_2 and especially β_3, the algorithm returns slightly biased posterior means. Although, as stated before, there is no theoretical true correspondence between the estimated shared nuisance parameter and the variance of the noise added to the data, the posterior mean for σ_2^2 is lower than the one used to perturb the data, meaning that - potentially - the additional information from the ODE solution helps shrinking down the overall measure of uncertainty regarding the second variable $\boldsymbol{x}_{\cdot 2}$ that we are trying to model.

Table 4.4: Results for the *FhNdata*: posterior means (*posterior standard deviations* within brackets) from *SnM* in the second column and point estimates from Vujačić et al. [28] in the third column

Parameters	Post. Mean (*post. sd*)	Vujačić et al. [28]
$\xi_2 = 0.5$	0.696 (*0.244*)	0.569
$\beta_1 = -0.33$	-0.322 (*0.041*)	-0.333
$\beta_2 = 0.067$	0.091 (*0.025*)	0.106
$\beta_3 = -0.067$	-0.028 (*0.066*)	-0.047
σ_2^2	0.060 (*0.019*)	

Runtime: 10,000 MCMC iterations in 29.17 seconds.

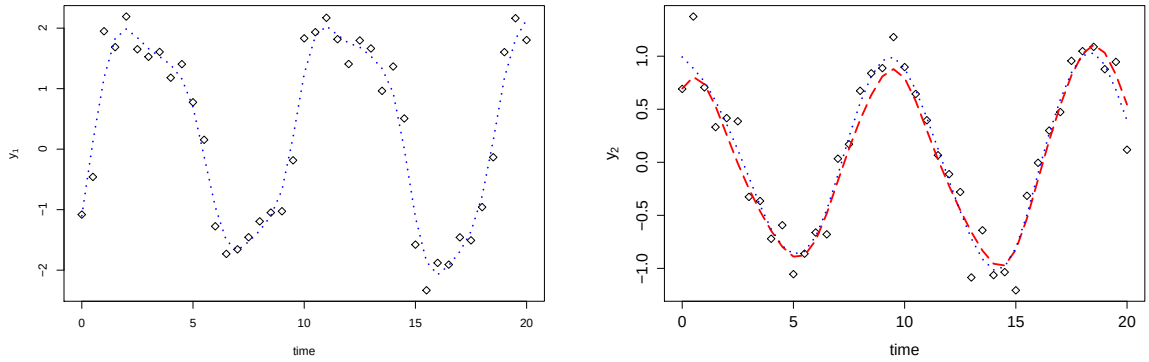

Figure 4.5: Observed noisy data (*dots*), smoothing spline (*dotted* line) and reconstructed solution (*long-dashed* line, only right plot) for the first variable (left plot) and the second variable $\boldsymbol{x}_{\cdot 2}$ (right plot)

4.4 Conclusions

We have proposed a Bayesian approach to indirectly solve an equation of an ODEs' system while doing inference on its parameters. The employed strategy is compartmentalized into two main stages: a first *smoothing* step, that serves as a reconstruction of the components of the process through penalized spline smoothing of the noisy observed data; a second *match* step, where the smoothed curves are numerically integrated and used as inputs for ridge regression. The two phases of the procedure are jointly governed by σ_k^2, a noise parameter - common to both steps - that measures the solution uncertainty of the k-th equation of the system we are indirectly solving. We evaluated the performance and reliability of the strategy through different ODEs' systems, starting from a simple one (with only one variable) and then moving to processes that had two or three variables and more complex time dynamics. We also tested the approach on a dataset previously analyzed by [28], to compare the results. The procedure we propose has the advantages of being fast, with a built-in quantification of uncertainty about the ODE solution obtained as a by-product of the inference. The 'tuning' parameters are minimal: the

number of knots and their placement have no substantial impact on the reliability of the smoothing step as long as the choice is reasonable with respect to the problem we are dealing with; different splines (i.e, *B-spline*, *thin plates*, etc.), that do not actually require such a choice, can be employed in the first step to address the issue. As far as the integration is concerned, we rely on an easy to implement - albeit 'rough' - trapezoidal rule that uses the observed time points as the grid to evaluate the integral: a better approximation can be achieved by employing a finer grid, at the cost of increased computational times. Other types of penalization, instead of the ridge, could be explored for the regression step of the strategy.

For future developments, we aim to be able to estimate all the parameters of the system while indirectly solving together and simultaneously the ordinary differential equations, instead of focusing on one of them. As trivial as it may seems to extend the approach, careful consideration is needed before moving toward this direction. For example, a first idea would be to independently run the procedure for each equation but, in that case, we would not be truly using the information at our disposal about the relationships between the variables. Another issue would be, when considering all the equations together, which of the two reconstructed curves (the smoothed spline and the regression solution) to use at each iteration of the MCMC procedure given that the measure of uncertainty considers them both. An interesting extension of the model could consider the regressing functions $h_j(\boldsymbol{x})$ as not known in advance and to be estimated along with the other parameters of the model. We already briefly explored this aspect using a Bayesian smoothed spline regression at the *match* step of the procedure: we obtained a satisfactory reconstruction of the curve but at the expense of losing interpretability of the parameter vector $\boldsymbol{\beta}$, meaning that further investigation on this topic is needed. Finally, as seen in Section 4.2.3, an interesting aspect would be to investigate deeper (and potentially quantify, theoretically) the consequences of the shared nuisance parameter σ_1^2. In this direction, an option could be to express the two levels of noise from the two sources of information ($\boldsymbol{y}_{\cdot 1}$ and $\boldsymbol{y}_1^\star$) as a proportion of the total shared nuisance parameter and to decide which curve to use, the smoothed one or the system's solution, based on these fractions.

Bibliography

[1] Barber, David. On solving Ordinary Differential Equations using Gaussian Processes. *arXiv preprint arXiv:1408.3807*, 2014.

[2] Barber, David and Wang, Yali. Gaussian Processes for Bayesian Estimation in Ordinary Differential Equations. In Jebara, Tony and Xing, Eric P., editors, *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1485–1493. JMLR Workshop and Conference Proceedings, 2014. URL http://jmlr.org/proceedings/papers/v32/barber14.pdf.

[3] Bonhoeffer, Sebastian, May, Robert M, Shaw, George M, and Nowak, Martin A. Virus dynamics and drug therapy. *Proceedings of the National Academy of Sciences*, 94(13):6971–6976, 1997.

[4] Brunel, Nicolas JB et al. Parameter estimation of ODE's via nonparametric estimators. *Electronic Journal of Statistics*, 2:1242–1267, 2008.

[5] Calderhead, Ben, Girolami, Mark, and Lawrence, Neil D. Accelerating Bayesian inference over nonlinear differential equations with Gaussian processes. In *Advances in neural information processing systems*, pages 217–224, 2009.

[6] Campbell, David and Steele, Russell J. Smooth functional tempering for nonlinear differential equation models. *Statistics and Computing*, 22(2):429–443, 2012.

[7] Chkrebtii, Oksana A, Campbell, David A, Calderhead, Ben, and Girolami, Mark A. Bayesian Solution Uncertainty Quantification for Differential Equations. *arXiv preprint arXiv:1306.2365*, 2013.

[8] Conrad, Patrick R, Girolami, Mark, Särkkä, Simo, Stuart, Andrew, and Zygalakis, Konstantinos. Probability measures for numerical solutions of differential equations. *arXiv preprint arXiv:1506.04592*, 2015.

[9] Dattner, Itai and Klaassen, Chris AJ. Estimation in Systems of Ordinary Differential Equations Linear in the Parameters. *arXiv preprint arXiv:1305.4126*, 2013.

[10] Dondelinger, F., Filippone, M., Rogers, S., and Husmeier, D. ODE parameter inference using adaptive gradient matching with Gaussian processes. In *Sixteenth International Conference on Artificial Intelligence and Statistics*, 2013. URL http://eprints.gla.ac.uk/77857/.

[11] Edelstein-Keshet, Leah. *Mathematical models in biology*, volume 46. Siam, 1988.

[12] FitzHugh, Richard. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical journal*, 1(6):445, 1961.

[13] González, Javier, Vujačić, Ivan, and Wit, Ernst. Inferring latent gene regulatory network kinetics. *Statistical applications in genetics and molecular biology*, 12 (1):109–127, 2013.

[14] González, Javier, Vujačić, Ivan, and Wit, Ernst. Reproducing kernel Hilbert space based estimation of systems of ordinary differential equations. *Pattern Recognition Letters*, 45:26–32, 2014.

[15] Gugushvili, Shota, Klaassen, Chris AJ, et al. $\sqrt{n}$-consistent parameter estimation for systems of ordinary differential equations: bypassing numerical integration via smoothing. *Bernoulli*, 18(3):1061–1098, 2012.

[16] Honkela, Antti, Girardot, Charles, Gustafson, E Hilary, Liu, Ya-Hsin, Furlong, Eileen EM, Lawrence, Neil D, and Rattray, Magnus. Model-based method for transcription factor target identification with limited data. *Proceedings of the National Academy of Sciences*, 107(17):7793–7798, 2010.

[17] Hooker, G, Xiao, L, and Ramsay, J. Collocinfer: collocation inference for dynamic systems. *R package version 0.1.0*, 2010.

[18] Liang, Hua and Wu, Hulin. Parameter estimation for differential equation models using a framework of measurement error in regression models. *Journal of the American Statistical Association*, 2012.

[19] Macdonald, Benn, Higham, Catherine, and Husmeier, Dirk. Controversy in mechanistic modelling with Gaussian processes. In *Journal of Machine Learning Research: Workshop and Conference Proceedings*, volume 37, pages 1539–1547. Microtome Publishing, 2015.

[20] Madár, János, Abonyi, János, Roubos, Hans, and Szeifert, Ferenc. Incorporating Prior Knowledge in a Cubic Spline Approximation Application to the Identification of Reaction Kinetic Models. *Industrial & engineering chemistry research*, 42(17):4043–4049, 2003.

[21] McKendrick, A. G. and Pai, M. Kesava. The Rate of Multiplication of Microorganisms: A Mathematical Study. *Proceedings of the Royal Society of Edinburgh*, 31:649–653, 1 1912. ISSN 0370-1646. doi: 10.1017/S0370164600025426.

[22] Morrissey, Edward R, Juárez, Miguel A, Denby, Katherine J, and Burroughs, Nigel J. Inferring the time-invariant topology of a nonlinear sparse gene regulatory network using fully Bayesian spline autoregression. *Biostatistics*, 12(4): 682–694, 2011.

[23] Nagumo, Jinichi, Arimoto, Suguru, and Yoshizawa, Shuji. An active pulse transmission line simulating nerve axon. *Proceedings of the IRE*, 50(10):2061–2070, 1962.

[24] Ramsay, Jim O, Hooker, G, Campbell, D, and Cao, J. Parameter estimation for differential equations: a generalized smoothing approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(5):741–796, 2007.

[25] Robinson, James C. *An introduction to ordinary differential equations*. Cambridge University Press, 2004.

[26] Titsias, Michalis K, Honkela, Antti, Lawrence, Neil D, and Rattray, Magnus. Identifying targets of multiple co-regulating transcription factors from expression time-series by Bayesian model comparison. *BMC systems biology*, 6(1):1, 2012.

[27] Varah, JM. A spline least squares method for numerical parameter estimation in differential equations. *SIAM Journal on Scientific and Statistical Computing*, 3(1):28–46, 1982.

[28] Vujačić, Ivan, Dattner, Itai, González, Javier, and Wit, Ernst. Time-course window estimator for ordinary differential equations linear in the parameters. *Statistics and Computing*, 6:1057–1070, 2015.

[29] Wood, Simon. *Generalized additive models: an introduction with R*. CRC Press, 2006.

[30] Xue, Hongqi, Miao, Hongyu, and Wu, Hulin. Sieve estimation of constant and time-varying coefficients in nonlinear ordinary differential equation models by considering both numerical error and measurement error. *Annals of statistics*, 38(4):2351, 2010.

Appendix A

Appendix

"SPATIO-TEMPORAL MODEL FOR MULTIPLE CHIP-SEQ EXPERIMENTS": supplementary material.

1 Posterior distributions and full conditionals

The full conditional of X_{mt} depends on the position of the node (m,t) w.r.t. to the graph. The probabilities for the sampling scheme are:

$$P(X_{11}=s|X_{-(mt)},\mathbf{Y},\mathbf{\Theta}) \propto q_{s,X_{12}}\ w_{s,X_{21}}\ \prod_{r=1}^{R_t} P(Y_{mtr}|X_{mt}=s)$$

$$P(X_{1T}=s|X_{-(mt)},\mathbf{Y},\mathbf{\Theta}) \propto w_{X_{1,T-1},s}\ q_{s,X_{2T}}\ \prod_{r=1}^{R_t} P(Y_{mtr}|X_{mt}=s)$$

$$P(X_{1t}=s|X_{-(mt)},\mathbf{Y},\mathbf{\Theta}) \propto w_{X_{1,t-1},s}\ w_{s,X_{1,t+1}}\ q_{s,X_{2,t}}\ \prod_{r=1}^{R_t} P(Y_{mtr}|X_{mt}=s)$$

$$P(X_{M1}=s|X_{-(mt)},\mathbf{Y},\mathbf{\Theta}) \propto q_{X_{M-1,1},s}\ w_{s,X_{M2}}\ \prod_{r=1}^{R_t} P(Y_{mtr}|X_{mt}=s)$$

$$P(X_{MT}=s|X_{-(mt)},\mathbf{Y},\mathbf{\Theta}) \propto q_{X_{M-1,T},s}\ w_{X_{M,T-1},s}\ \prod_{r=1}^{R_t} P(Y_{mtr}|X_{mt}=s)$$

$$P(X_{Mt}=s|X_{-(mt)},\mathbf{Y},\mathbf{\Theta}) \propto q_{X_{M-1,t},s}\ w_{X_{M,t-1},s}\ w_{s,X_{M,t+1}}\ \prod_{r=1}^{R_t} P(Y_{mtr}|X_{mt}=s)$$

$$P(X_{m1} = s|X_{-(mt)}, \mathbf{Y}, \mathbf{\Theta}) \propto q_{X_{m-1,1},s}\ q_{s,X_{m+1,1}}\ w_{s,X_{m2}} \prod_{r=1}^{R_t} P(Y_{mtr}|X_{mt} = s)$$

$$P(X_{mT} = s|X_{-(mt)}, \mathbf{Y}, \mathbf{\Theta}) \propto q_{X_{m-1,T},s}\ q_{s,X_{m+1,T}}\ w_{X_{m,T-1},s} \prod_{r=1}^{R_t} P(Y_{mtr}|X_{mt} = s)$$

$$P(X_{mt} = s|X_{-(mt)}, \mathbf{Y}, \mathbf{\Theta}) \propto q_{X_{m-1,t},s}\ q_{s,X_{m+1,t}}\ w_{X_{m,t-1},s}\ w_{X_{m,t+1},s} \prod_{r=1}^{R_t} P(Y_{mtr}|X_{mt} = s)$$

with $s\,\epsilon\,\{0,1\}$, $m = 2,\ldots,M-1$, $t = 2,\ldots,T-1$ and $P(Y_{mtr}|X_{mt} = s)$ is the likelihood of the mixture model.

The inflation parameters π_{tr} and the Poisson distribution parameters μ_{tr} are updated through a Gibbs sampler, using Beta and Gamma conjugate priors:

$$\begin{aligned}
\pi_{tr} &\sim Beta(A_{\pi_{tr}},\ B_{\pi_{tr}}) \\
\lambda^B_{tr} &\sim Gamma(A_{\lambda^B_{tr}}, B_{\lambda^B_{tr}}) \\
\lambda^S_{tr} &\sim Gamma(A_{\lambda^S_{tr}}, B_{\lambda^S_{tr}})
\end{aligned}$$

with $A_{\pi_{tr}}$, $B_{\pi_{tr}}$, $A_{\lambda^B_{tr}}$, $B_{\lambda^B_{tr}}$, $A_{\lambda^S_{tr}}$, $B_{\lambda^S_{tr}}$ all being hyperparameters chosen to represent flat prior distributions. These priors distributions lead to the following posteriors:

$$\begin{aligned}
\pi_{tr} &\sim Beta\left(A_{\pi_{tr}} + \sum_{m=1}^{M} \mathbb{1}(X_{mt} = 0, Z_{mtr} = 1),\ B_{\pi_{tr}} + \sum_{m=1}^{M} \mathbb{1}(X_{mt} = 0, Z_{mtr} = 0)\right) \\
\lambda^B_{tr} &\sim Gamma\left(A_{\lambda^B_{tr}} + \sum_{m=1}^{M} y_{mtr}\mathbb{1}(X_{mt} = 0, Z_{mtr} = 1), B_{\lambda^B_{tr}} + \sum_{m=1}^{M} \mathbb{1}(X_{mt} = 0, Z_{mtr} = 1)\right) \\
\lambda^S_{tr} &\sim Gamma\left(A_{\lambda^S_{tr}} + \sum_{m=1}^{M} y_{mtr}\mathbb{1}(X_{mt} = 1), B_{\lambda^S_{tr}} + \sum_{m=1}^{M} \mathbb{1}(X_{mt} = 1)\right)
\end{aligned}$$

To estimate the transition probabilities and background/signal parameters for the Negative Binomial distribution, we use a Metropolis-Hastings procedure.

For the mean and dispersion parameters, the priors involved are:

$$\begin{aligned}
\mu^B_{tr} &\sim Gamma\left(A_{\mu^B_{tr}}, B_{\mu^B_{tr}}\right),\ \ \mu^S_{tr} \sim Gamma\left(A_{\mu^S_{tr}}, B_{\mu^S_{tr}}\right) \\
\phi^B_{tr} &\sim Gamma\left(A_{\phi^B_{tr}}, B_{\phi^B_{tr}}\right),\ \ \phi^S_{tr} \sim Gamma\left(A_{\phi^S_{tr}}, B_{\phi^S_{tr}}\right)
\end{aligned}$$

with $A_{\mu_{tr}^B}, B_{\mu_{tr}^B}, A_{\mu_{tr}^S}, B_{\mu_{tr}^S}, A_{\phi_{tr}^B}, B_{\phi_{tr}^B}, A_{\phi_{tr}^S}, B_{\phi_{tr}^S}$ all being hyperparameters chosen to represent flat prior distributions. We obtain the following posterior distributions:

$$\begin{aligned}
\mu_{tr}^B &\sim f(\mu_{tr}^B|\phi_{tr}^B, \mathbf{Y}, \mathbf{X}, \mathbf{Z}) \\
&= \left(\frac{\mu_{tr}^B}{\mu_{tr}^B + \phi_{tr}^B}\right)^{\sum_{m=1}^M y_{mtr}\mathbb{1}(X_{mt}=0, Z_{mtr}=1)} \left(\frac{\phi_{tr}^B}{\mu_{tr}^B + \phi_{tr}^B}\right)^{\phi_{tr}^B \sum_{m=1}^M \mathbb{1}(X_{mt}=0, Z_{mtr}=1)} \\
&\times \left(\mu_{tr}^B\right)^{A_{\mu_{tr}^B}-1} exp\left(-B_{\mu_{tr}^B}\mu_{tr}^B\right)
\end{aligned}$$

$$\begin{aligned}
\mu_{tr}^S &\sim f(\mu_{tr}^S|\phi_{tr}^S, \mathbf{Y}, \mathbf{X}, \mathbf{Z}) \\
&= \left(\frac{\mu_{tr}^S}{\mu_{tr}^S + \phi_{tr}^S}\right)^{\sum_{m=1}^M y_{mtr}\mathbb{1}(X_{mt}=1)} \left(\frac{\phi_{tr}^S}{\mu_{tr}^S + \phi_{tr}^B}\right)^{\phi_{tr}^S \sum_{m=1}^M \mathbb{1}(X_{mt}=1)} \\
&\times \left(\mu_{tr}^S\right)^{A_{\mu_{tr}^S}-1} exp\left(-B_{\mu_{tr}^S}\mu_{tr}^S\right)
\end{aligned}$$

$$\begin{aligned}
\phi_{tr}^B &\sim f(\phi_{tr}^B|\mu_{tr}^B, \mathbf{Y}, \mathbf{X}, \mathbf{Z}) \\
&= \prod_{m=1}^M \left[\frac{\Gamma(y_{mtr}+\phi_{tr}^B)}{\Gamma(\phi_{tr}^B)\Gamma(y_{mtr}+1)} \left(\frac{\mu_{tr}^B}{\phi_{tr}^B + \mu_{tr}^B}\right)^{y_{mtr}} \left(\frac{\phi_{tr}^B}{\phi_{tr}^B + \mu_{tr}^B}\right)^{\phi_{tr}^B}\right]^{\mathbb{1}(X_{mt}=0, Z_{mtr}=1)} \\
&\times \left(\phi_{tr}^B\right)^{A_{\phi_{tr}^B}-1} exp\left(-B_{\phi_{tr}^B}\phi_{tr}^B\right)
\end{aligned}$$

$$\begin{aligned}
\phi_{tr}^S &\sim f(\phi_{tr}^S|\mu_{tr}^S, \mathbf{Y}, \mathbf{X}, \mathbf{Z}) \\
&= \prod_{m=1}^M \left[\frac{\Gamma(y_{mtr}+\phi_{tr}^S)}{\Gamma(\phi_{tr}^S)\Gamma(y_{mtr}+1)} \left(\frac{\mu_{tr}^S}{\phi_{tr}^S + \mu_{tr}^S}\right)^{y_{mtr}} \left(\frac{\phi_{tr}^S}{\phi_{tr}^S + \mu_{tr}^S}\right)^{\phi_{tr}^S}\right]^{\mathbb{1}(X_{mt}=1)} \\
&\times \left(\phi_{tr}^S\right)^{A_{\phi_{tr}^S}-1} exp\left(-B_{\phi_{tr}^S}\phi_{tr}^S\right)
\end{aligned}$$

The Metropolis-Hastings procedure uses a Truncated Normal Distribution as a proposal to generate new values:

$$
\begin{aligned}
{}^{\star}\mu_{tr}^{B} &\sim N\left({}^{k}\mu_{tr}^{B},\, \sigma^2_{\mu_{tr}^{B}}\right)\\
{}^{\star}\mu_{tr}^{S} &\sim N\left({}^{k}\mu_{tr}^{S},\, \sigma^2_{\mu_{tr}^{S}}\right)\\
{}^{\star}\phi_{tr}^{B} &\sim N\left({}^{k}\phi_{tr}^{B},\, \sigma^2_{\phi_{tr}^{B}}\right)\\
{}^{\star}\phi_{tr}^{S} &\sim N\left({}^{k}\phi_{tr}^{S},\, \sigma^2_{\phi_{tr}^{S}}\right)
\end{aligned}
$$

where ${}^{k}\mu_{tr}^{B},\ {}^{k}\mu_{tr}^{S},\ {}^{k}\phi_{tr}^{B},\ {}^{k}\phi_{tr}^{S}$ are the current value at k-th iteration and the variances are tuning parameters. This choice of a proposal ensures that candidate values are always positive; the ratio of the normalizing constants has to be considered into the formula for the acceptance ratios.

The new values ${}^{\star}\mu_{tr}^{B},\ {}^{\star}\mu_{tr}^{S}$ are accepted with probabilities $min\left\{1, D_{\mu_{tr}^{B}}\right\}$ and $min\left\{1, D_{\mu_{tr}^{S}}\right\}$, while the new values ${}^{\star}\phi_{tr}^{B},\ {}^{\star}\phi_{tr}^{S}$ are accepted with probabilities $min\left\{1, D_{\phi_{tr}^{B}}\right\}$ and $min\left\{1, D_{\phi_{tr}^{S}}\right\}$, where

$$
\begin{aligned}
D_{\mu_{tr}^{B}} &= \frac{p({}^{\star}\mu_{tr}^{B}|\phi_{tr}^{B})q({}^{k}\mu_{tr}^{B}|{}^{\star}\mu_{tr}^{B})}{p({}^{k}\mu_{tr}^{B}|\phi_{tr}^{B})q({}^{\star}\mu_{tr}^{B}|{}^{k}\mu_{tr}^{B})}\\
&= \left(\frac{{}^{\star}\mu_{tr}^{B}}{{}^{\star}\mu_{tr}^{B}+\phi_{tr}^{B}} \Big/ \frac{{}^{k}\mu_{tr}^{B}}{{}^{k}\mu_{tr}^{B}+\phi_{tr}^{B}}\right)^{\sum_{m=1}^{M} y_{mtr}\mathbb{1}(X_{mt}=0,Z_{mtr}=1)} \left(\frac{{}^{k}\mu_{tr}^{B}+\phi_{tr}^{B}}{{}^{\star}\mu_{tr}^{B}+\phi_{tr}^{B}}\right)^{\phi_{tr}^{B}\sum_{m=1}^{M}\mathbb{1}(X_{mt}=0,Z_{mtr}=1)}\\
&\times \left(\frac{{}^{\star}\mu_{tr}^{B}}{{}^{k}\mu_{tr}^{B}}\right)^{A_{\mu_{tr}^{B}}-1} exp\left(-B_{\mu_{tr}^{B}}({}^{\star}\mu_{tr}^{B} - {}^{k}\mu_{tr}^{B})\right) E_{\mu_{tr}^{B}}
\end{aligned}
$$

$$
\begin{aligned}
D_{\mu_{tr}^{S}} &= \frac{p({}^{\star}\mu_{tr}^{S}|\phi_{tr}^{S})q({}^{k}\mu_{tr}^{S}|{}^{\star}\mu_{tr}^{S})}{p({}^{k}\mu_{tr}^{S}|\phi_{tr}^{S})q({}^{\star}\mu_{tr}^{S}|{}^{k}\mu_{tr}^{S})}\\
&= \left(\frac{{}^{\star}\mu_{tr}^{S}}{{}^{\star}\mu_{tr}^{S}+\phi_{tr}^{S}} \Big/ \frac{{}^{k}\mu_{tr}^{S}}{{}^{k}\mu_{tr}^{S}+\phi_{tr}^{S}}\right)^{\sum_{m=1}^{M} y_{mtr}\mathbb{1}(X_{mt}=1)} \left(\frac{{}^{k}\mu_{tr}^{S}+\phi_{tr}^{S}}{{}^{\star}\mu_{tr}^{S}+\phi_{tr}^{S}}\right)^{\phi_{tr}^{S}\sum_{m=1}^{M}\mathbb{1}(X_{mt}=1)}\\
&\times \left(\frac{{}^{\star}\mu_{tr}^{S}}{{}^{k}\mu_{tr}^{S}}\right)^{A_{\mu_{tr}^{S}}-1} exp\left(-B_{\mu_{tr}^{S}}({}^{\star}\mu_{tr}^{S} - {}^{k}\mu_{tr}^{S})\right) E_{\mu_{tr}^{S}}
\end{aligned}
$$

$$D_{\phi_{tr}^B} = \frac{p(^\star\phi_{tr}^B|\mu_{tr}^B)q(^k\phi_{tr}^B|^\star\phi_{tr}^B)}{p(^k\phi_{tr}^B|\mu_{tr}^B)q(^\star\phi_{tr}^B|^k\phi_{tr}^B)}$$
$$= \prod_{m=1}^{M}\left[\frac{\Gamma(y_{mtr}+^\star\phi_{tr}^B)\Gamma(^k\phi_{tr}^B)}{\Gamma(y_{mtr}+^k\phi_{tr}^B)\Gamma(^\star\phi_{tr}^B)}\left(\frac{\mu_{tr}^B+^k\phi_{tr}^B}{\mu_{tr}^B+^\star\phi_{tr}^B}\right)^{y_{mtr}}\left(\frac{^\star\phi_{tr}^B}{^\star\phi_{tr}^B+\mu_{tr}^B}\right)^{^\star\phi_{tr}^B}\right]^{\sum_{m=1}^{M}\mathbb{1}(X_{mt}=0,Z_{mtr}=1)}$$
$$\times\left[\left(\frac{^k\phi_{tr}^B}{^k\phi_{tr}^B+\mu_{tr}^B}\right)^{-^k\phi_{tr}^B}\right]^{\sum_{m=1}^{M}\mathbb{1}(X_{mt}=0,Z_{mtr}=1)}\left(\frac{^\star\phi_{tr}^B}{^k\phi_{tr}^B}\right)^{A_{\phi_{tr}^B}-1}exp\left(-B_{\phi_{tr}^B}(^\star\phi_{tr}^B-^k\phi_{tr}^B)\right)E_{\phi_{tr}^B}$$

$$D_{\phi_{tr}^S} = \frac{p(^\star\phi_{tr}^S|\mu_{tr}^S)q(^k\phi_{tr}^S|^\star\phi_{tr}^S)}{p(^k\phi_{tr}^S|\mu_{tr}^S)q(^\star\phi_{tr}^S|^k\phi_{tr}^S)}$$
$$= \prod_{m=1}^{M}\left[\frac{\Gamma(y_{mtr}+^\star\phi_{tr}^S)\Gamma(^k\phi_{tr}^S)}{\Gamma(y_{mtr}+^k\phi_{tr}^S)\Gamma(^\star\phi_{tr}^S)}\left(\frac{\mu_{tr}^S+^k\phi_{tr}^S}{\mu_{tr}^S+^\star\phi_{tr}^S}\right)^{y_{mtr}}\left(\frac{^\star\phi_{tr}^S}{^\star\phi_{tr}^S+\mu_{tr}^S}\right)^{^\star\phi_{tr}^S}\left(\frac{^k\phi_{tr}^S}{^k\phi_{tr}^S+\mu_{tr}^S}\right)^{-^k\phi_{tr}^S}\right]^{\sum_{m=1}^{M}\mathbb{1}(X_{mt}=1)}$$
$$\times\left(\frac{^\star\phi_{tr}^S}{^k\phi_{tr}^S}\right)^{A_{\phi_{tr}^S}-1}exp\left(-B_{\phi_{tr}^S}(^\star\phi_{tr}^S-^k\phi_{tr}^S)\right)E_{\phi_{tr}^S}$$

and

$$E_{\mu_{tr}^B} = \frac{\Phi(^k\mu_{tr}^B/\sigma_{\mu_{tr}^B})}{\Phi(^\star\mu_{tr}^B/\sigma_{\mu_{tr}^B})}$$
$$E_{\mu_{tr}^S} = \frac{\Phi(^k\mu_{tr}^S/\sigma_{\mu_{tr}^S})}{\Phi(^\star\mu_{tr}^S/\sigma_{\mu_{tr}^S})}$$
$$E_{\phi_{tr}^B} = \frac{\Phi(^k\phi_{tr}^B/\sigma_{\phi_{tr}^B})}{\Phi(^\star\phi_{tr}^B/\sigma_{\phi_{tr}^B})}$$
$$E_{\phi_{tr}^S} = \frac{\Phi(^k\phi_{tr}^S/\sigma_{\phi_{tr}^S})}{\Phi(^\star\phi_{tr}^S/\sigma_{\phi_{tr}^S})}$$

with $\Phi(\cdot)$ the distribution function of a standard normal distribution.

2 Simulation settings

We assess the performance of our model *stMRF* and compare it to other two existing methods, *iSeq* and *MRF*, in a simulated environment with the characteristics showed in Table 2. For all the scenarios with fix the number of replicates $R = 3$ and the number of regions $M = 2000$. We use a Zero-Inflated Negative Binomial to model the background component and a Negative Binomial for the signal of the mixture. The parameters ϕ^B, ϕ^S and μ^B have been kept constant for all the scenarios aforementioned and equal to the quantities in Table 3.

Table 1: Summary of true values used in the simulated environment (T=2).

Time-points	Zero-Inflation	NB means signal (S)	Binding propensity		Scenario
			q_1	w_1	
$T = 2$	$\pi_{1r} = (0.5, 0.5, 0.6)$	$\mu_{1r} = (10, 10, 11)$	0.3	0.6	(1a)
			0.3	0.8	(2a)
		$\mu_{2r} = (10, 11, 10)$	0.7	0.4	(3a)
			0.7	0.8	(4a)
	$\pi_{2r} = (0.4, 0.5, 0.5)$	$\mu_{1r} = (20, 19, 21)$	0.3	0.6	(5a)
			0.3	0.8	(6a)
		$\mu_{2r} = (20, 20, 21)$	0.7	0.4	(7a)
			0.7	0.8	(8a)
	$\pi_{1r} = (0.8, 0.9, 0.8)$	$\mu_{1r} = (10, 10, 11)$	0.3	0.6	(9a)
			0.3	0.8	(10a)
		$\mu_{2r} = (10, 11, 10)$	0.7	0.4	(11a)
			0.7	0.8	(12a)
	$\pi_{2r} = (0.8, 0.9, 0.9)$	$\mu_{1r} = (20, 19, 21)$	0.3	0.6	(13a)
			0.3	0.8	(14a)
		$\mu_{2r} = (20, 20, 21)$	0.7	0.4	(15a)
			0.7	0.8	(16a)

Table 2: Summary of true values used in the simulated environment (T=4).

Time-points	Zero-Inflation	NB means signal (S)	Binding propensity		Scenario
			q_1	w_1	
$T = 4$	$\pi_{1r} = (0.5, 0.5, 0.6)$	$\mu_{1r} = (10, 10, 11)$	0.3	0.6	(1b)
		$\mu_{2r} = (10, 11, 10)$	0.3	0.8	(2b)
	$\pi_{2r} = (0.4, 0.5, 0.5)$	$\mu_{3r} = (10, 11, 10)$	0.7	0.4	(3b)
		$\mu_{4r} = (10, 11, 10)$	0.7	0.8	(4b)
	$\pi_{3r} = (0.6, 0.5, 0.5)$	$\mu_{1r} = (20, 19, 21)$	0.3	0.6	(5b)
		$\mu_{2r} = (20, 20, 21)$	0.3	0.8	(6b)
	$\pi_{4r} = (0.5, 0.4, 0.5)$	$\mu_{3r} = (19, 19, 21)$	0.7	0.4	(7b)
		$\mu_{4r} = (20, 19, 20)$	0.7	0.8	(8b)
	$\pi_{1r} = (0.9, 0.8, 0.9)$	$\mu_{1r} = (10, 10, 11)$	0.3	0.6	(9b)
		$\mu_{2r} = (10, 11, 10)$	0.3	0.8	(10b)
	$\pi_{2r} = (0.8, 0.9, 0.9)$	$\mu_{3r} = (10, 11, 10)$	0.7	0.4	(11b)
		$\mu_{4r} = (10, 11, 10)$	0.7	0.8	(12b)
	$\pi_{3r} = (0.9, 0.8, 0.9)$	$\mu_{1r} = (20, 19, 21)$	0.3	0.6	(13b)
		$\mu_{2r} = (20, 20, 21)$	0.3	0.8	(14b)
	$\pi_{4r} = (0.8, 0.9, 0.8)$	$\mu_{3r} = (19, 19, 21)$	0.7	0.4	(15b)
		$\mu_{4r} = (20, 19, 20)$	0.7	0.8	(16b)

Table 3: Fixed true values for the parameters of the background and signal densities for all scenarios.

Time-points	NB means background (B)	NB dispersion parameters	
		Background (B)	Signal (S)
$T = 2$	$\mu_{1r} = (2, 1, 1)$	$\phi_{1r} = (1.5, 1.5, 1.5)$	$\phi_{1r} = (1.5, 1.5, 1.5)$
	$\mu_{2r} = (1, 2, 2)$	$\phi_{2r} = (1.5, 1.5, 1.5)$	$\phi_{2r} = (1.5, 1.5, 1.5)$
$T = 4$	$\mu_{1r} = (2, 1, 1)$	$\phi_{1r} = (1.5, 1.5, 1.5)$	$\phi_{1r} = (1.5, 1.5, 1.5)$
	$\mu_{2r} = (1, 2, 2)$	$\phi_{2r} = (1.5, 1.5, 1.5)$	$\phi_{2r} = (1.5, 1.5, 1.5)$
	$\mu_{3r} = (3, 2, 3)$	$\phi_{3r} = (1.5, 1.5, 1.5)$	$\phi_{3r} = (1.5, 1.5, 1.5)$
	$\mu_{4r} = (1, 1, 1)$	$\phi_{4r} = (1.5, 1.5, 1.5)$	$\phi_{4r} = (1.5, 1.5, 1.5)$

3 Tables & Figures

Table 4: Posterior means of the measurement model parameters (μ, ϕ) and inflation proportion (π). Posterior standard deviations are provided between brackets.

Scenario			$\hat{\pi}$	$\hat{\mu}^B$	$\hat{\mu}^S$	$\hat{\phi}^B$	$\hat{\phi}^S$
(1a)	t=1	r=1	0.72 (*0.15*)	0.82 (*0.18*)	10.14 (*0.39*)	1.16 (*0.23*)	1.54 (*0.12*)
		r=2	0.43 (*0.11*)	1.07 (*0.21*)	11.25 (*0.41*)	1.35 (*0.26*)	1.50 (*0.12*)
		r=3	0.37 (*0.03*)	1.10 (*0.12*)	10.99 (*0.41*)	4.18 (*0.66*)	1.55 (*0.13*)
	t=2	r=1	0.62 (*0.13*)	0.84 (*0.16*)	10.32 (*0.40*)	1.21 (*0.24*)	1.50 (*0.13*)
		r=2	0.57 (*0.05*)	1.91 (*0.17*)	10.10 (*0.41*)	1.47 (*0.20*)	1.37 (*0.12*)
		r=3	0.52 (*0.05*)	2.05 (*0.21*)	10.03 (*0.39*)	1.44 (*0.20*)	1.41 (*0.13*)
(2a)	t=1	r=1	0.47 (*0.06*)	1.13 (*0.14*)	10.18 (*0.40*)	2.18 (*0.40*)	1.42 (*0.12*)
		r=2	0.44 (*0.06*)	0.87 (*0.13*)	10.82 (*0.39*)	1.91 (*0.37*)	1.51 (*0.13*)
		r=3	0.68 (*0.12*)	0.81 (*0.13*)	11.19 (*0.42*)	0.61 (*0.15*)	1.52 (*0.12*)
	t=2	r=1	0.68 (*0.14*)	0.86 (*0.18*)	10.82 (*0.40*)	0.58 (*0.14*)	1.52 (*0.12*)
		r=2	0.57 (*0.04*)	1.97 (*0.14*)	9.61 (*0.38*)	1.47 (*0.20*)	1.55 (*0.13*)
		r=3	0.53 (*0.05*)	2.00 (*0.18*)	9.89 (*0.39*)	1.44 (*0.20*)	1.41 (*0.12*)
(3a)	t=1	r=1	0.59 (*0.14*)	1.04 (*0.23*)	9.91 (*0.37*)	1.17 (*0.21*)	1.48 (*0.08*)
		r=2	0.40 (*0.06*)	1.03 (*0.14*)	11.52 (*0.43*)	1.94 (*0.34*)	1.44 (*0.08*)
		r=3	0.59 (*0.10*)	0.93 (*0.15*)	11.28 (*0.39*)	1.19 (*0.26*)	1.69 (*0.09*)
	t=2	r=1	0.58 (*0.16*)	0.83 (*0.20*)	10.12 (*0.39*)	1.07 (*0.22*)	1.42 (*0.07*)
		r=2	0.53 (*0.04*)	2.14 (*0.18*)	10.30 (*0.41*)	1.58 (*0.21*)	1.45 (*0.08*)
		r=3	0.57 (*0.06*)	1.82 (*0.19*)	9.81 (*0.38*)	1.18 (*0.16*)	1.48 (*0.08*)
(4a)	t=1	r=1	0.43 (*0.05*)	1.09 (*0.13*)	10.34 (*0.27*)	2.69 (*1.17*)	1.57 (*0.09*)
		r=2	0.41 (*0.07*)	0.95 (*0.16*)	11.34 (*0.31*)	1.62 (*0.81*)	1.62 (*0.08*)
		r=3	0.62 (*0.07*)	0.93 (*0.12*)	10.78 (*0.30*)	1.00 (*0.30*)	1.43 (*0.07*)
	t=2	r=1	0.45 (*0.06*)	1.20 (*0.17*)	10.18 (*0.28*)	2.63 (*1.27*)	1.46 (*0.07*)
		r=2	0.57 (*0.04*)	2.06 (*0.15*)	9.82 (*0.28*)	1.47 (*0.45*)	1.44 (*0.08*)
		r=3	0.52 (*0.06*)	2.11 (*0.21*)	10.39 (*0.30*)	1.44 (*0.50*)	1.57 (*0.08*)

Table 5: Posterior means of the measurement model parameters (μ, ϕ) and inflation proportion (π). Posterior standard deviations are provided between brackets.

Scenario			$\hat{\pi}$	$\hat{\mu}^B$	$\hat{\mu}^S$	$\hat{\phi}^B$	$\hat{\phi}^S$
(5a)	t=1	r=1	0.55 (*0.04*)	1.82 (*0.16*)	19.72 (*0.52*)	1.29 (*0.31*)	1.54 (*0.07*)
		r=2	0.40 (*0.08*)	0.96 (*0.18*)	20.45 (*0.54*)	1.71 (*1.07*)	1.49 (*0.07*)
		r=3	0.67 (*0.16*)	0.75 (*0.17*)	19.14 (*0.51*)	0.75 (*0.41*)	1.43 (*0.06*)
	t=2	r=1	0.45 (*0.03*)	2.32 (*0.14*)	19.96 (*0.56*)	2.20 (*0.51*)	1.37 (*0.06*)
		r=2	0.51 (*0.05*)	1.18 (*0.12*)	21.37 (*0.56*)	3.40 (*1.31*)	1.51 (*0.07*)
		r=3	0.51 (*0.04*)	2.07 (*0.17*)	21.11 (*0.54*)	1.37 (*0.34*)	1.48 (*0.08*)
(6a)	t=1	r=1	0.55 (*0.06*)	1.96 (*0.20*)	20.54 (*0.57*)	0.97 (*0.24*)	1.56 (*0.08*)
		r=2	0.59 (*0.13*)	0.76 (*0.17*)	19.53 (*0.50*)	0.78 (*0.47*)	1.47 (*0.07*)
		r=3	0.52 (*0.06*)	0.97 (*0.11*)	20.19 (*0.55*)	1.43 (*0.49*)	1.49 (*0.07*)
	t=2	r=1	0.54 (*0.07*)	1.87 (*0.22*)	19.92 (*0.54*)	1.09 (*0.37*)	1.53 (*0.07*)
		r=2	0.82 (*0.13*)	0.74 (*0.14*)	20.09 (*0.51*)	1.47 (*0.46*)	1.52 (*0.07*)
		r=3	0.50 (*0.04*)	2.08 (*0.18*)	21.46 (*0.59*)	1.44 (*0.51*)	1.38 (*0.06*)
(7a)	t=1	r=1	0.48 (*0.03*)	2.03 (*0.14*)	19.69 (*0.51*)	2.64 (*0.80*)	1.50 (*0.07*)
		r=2	0.45 (*0.10*)	0.96 (*0.18*)	20.31 (*0.55*)	1.35 (*0.66*)	1.49 (*0.08*)
		r=3	0.51 (*0.07*)	1.03 (*0.14*)	18.81 (*0.48*)	1.93 (*0.89*)	1.49 (*0.07*)
	t=2	r=1	0.42 (*0.03*)	2.35 (*0.15*)	18.96 (*0.52*)	2.34 (*0.66*)	1.45 (*0.07*)
		r=2	0.58 (*0.08*)	0.95 (*0.12*)	20.58 (*0.51*)	2.39 (*1.14*)	1.55 (*0.07*)
		r=3	0.53 (*0.05*)	1.94 (*0.19*)	20.54 (*0.56*)	1.13 (*0.28*)	1.48 (*0.07*)
(8a)	t=1	r=1	0.49 (*0.03*)	2.06 (*0.13*)	20.82 (*0.54*)	2.20 (*0.57*)	1.54 (*0.07*)
		r=2	0.41 (*0.07*)	0.95 (*0.17*)	19.84 (*0.55*)	1.70 (*1.15*)	1.53 (*0.07*)
		r=3	0.57 (*0.12*)	0.92 (*0.17*)	18.07 (*0.48*)	1.31 (*0.72*)	1.52 (*0.07*)
	t=2	r=1	0.49 (*0.04*)	2.15 (*0.20*)	21.08 (*0.57*)	1.49 (*0.42*)	1.38 (*0.07*)
		r=2	0.62 (*0.09*)	0.91 (*0.12*)	21.35 (*0.56*)	1.86 (*0.86*)	1.50 (*0.07*)
		r=3	0.51 (*0.05*)	1.78 (*0.15*)	21.09 (*0.52*)	1.88 (*0.61*)	1.51 (*0.07*)

Table 6: Posterior means of the measurement model parameters (μ, ϕ) and inflation proportion (π). Posterior standard deviations are provided between brackets.

Scenario			$\hat{\pi}$	$\hat{\mu}^B$	$\hat{\mu}^S$	$\hat{\phi}^B$	$\hat{\phi}^S$
(9a)	t=1	r=1	0.82 (*0.07*)	1.10 (*0.10*)	10.24 (*0.28*)	2.22 (*0.98*)	1.54 (*0.08*)
		r=2	0.83 (*0.08*)	1.01 (*0.10*)	10.88 (*0.31*)	1.38 (*0.46*)	1.57 (*0.08*)
		r=3	0.79 (*0.09*)	1.00 (*0.12*)	11.33 (*0.33*)	1.56 (*0.69*)	1.55 (*0.09*)
	t=2	r=1	0.84 (*0.06*)	1.11 (*0.09*)	10.20 (*0.29*)	2.26 (*0.89*)	1.41 (*0.08*)
		r=2	0.94 (*0.04*)	1.84 (*0.10*)	9.82 (*0.29*)	1.44 (*0.25*)	1.36 (*0.07*)
		r=3	0.89 (*0.05*)	1.98 (*0.12*)	10.59 (*0.30*)	1.56 (*0.34*)	1.48 (*0.08*)
(10a)	t=1	r=1	0.83 (*0.07*)	1.12 (*0.10*)	9.57 (*0.26*)	2.53 (*0.99*)	1.55 (*0.08*)
		r=2	0.85 (*0.07*)	1.00 (*0.09*)	11.08 (*0.32*)	1.52 (*0.51*)	1.49 (*0.07*)
		r=3	0.82 (*0.09*)	1.01 (*0.12*)	11.03 (*0.29*)	1.35 (*0.51*)	1.45 (*0.07*)
	t=2	r=1	0.89 (*0.07*)	1.06 (*0.10*)	10.09 (*0.28*)	1.31 (*0.41*)	1.47 (*0.08*)
		r=2	0.93 (*0.04*)	1.97 (*0.11*)	9.98 (*0.29*)	1.41 (*0.24*)	1.46 (*0.08*)
		r=3	0.92 (*0.05*)	2.06 (*0.13*)	10.41 (*0.29*)	1.44 (*0.33*)	1.53 (*0.08*)
(11a)	t=1	r=1	0.91 (*0.06*)	0.93 (*0.08*)	10.68 (*0.30*)	2.72 (*1.16*)	1.65 (*0.09*)
		r=2	0.78 (*0.12*)	0.99 (*0.15*)	11.55 (*0.33*)	1.76 (*1.16*)	1.58 (*0.09*)
		r=3	0.73 (*0.10*)	1.06 (*0.14*)	10.61 (*0.31*)	2.08 (*1.02*)	1.47 (*0.08*)
	t=2	r=1	0.85 (*0.07*)	1.02 (*0.09*)	10.26 (*0.28*)	1.94 (*0.68*)	1.57 (*0.09*)
		r=2	0.92 (*0.04*)	1.87 (*0.10*)	9.60 (*0.29*)	1.37 (*0.23*)	1.40 (*0.07*)
		r=3	0.90 (*0.04*)	2.13 (*0.13*)	10.27 (*0.30*)	1.68 (*0.34*)	1.43 (*0.08*)
(12a)	t=1	r=1	0.85 (*0.07*)	1.11 (*0.11*)	10.53 (*0.30*)	1.87 (*0.62*)	1.46 (*0.08*)
		r=2	0.80 (*0.09*)	1.00 (*0.11*)	11.35 (*0.31*)	1.46 (*0.54*)	1.58 (*0.09*)
		r=3	0.74 (*0.09*)	1.02 (*0.12*)	11.19 (*0.33*)	1.99 (*0.91*)	1.41 (*0.08*)
	t=2	r=1	0.87 (*0.06*)	1.17 (*0.10*)	10.60 (*0.30*)	1.90 (*0.59*)	1.47 (*0.08*)
		r=2	0.94 (*0.04*)	1.88 (*0.12*)	9.80 (*0.25*)	1.45 (*0.28*)	1.58 (*0.08*)
		r=3	0.89 (*0.04*)	1.96 (*0.11*)	10.29 (*0.27*)	1.78 (*0.35*)	1.55 (*0.08*)

Table 7: Posterior means of the measurement model parameters (μ, ϕ) and inflation proportion (π). Posterior standard deviations are provided between brackets.

Scenario			$\hat{\pi}$	$\hat{\mu}^B$	$\hat{\mu}^S$	$\hat{\phi}^B$	$\hat{\phi}^S$
(13a)	t=1	r=1	0.83 (*0.05*)	2.13 (*0.14*)	20.37 (*0.58*)	1.52 (*0.33*)	1.53 (*0.08*)
		r=2	0.85 (*0.09*)	0.91 (*0.09*)	20.18 (*0.59*)	1.33 (*0.44*)	1.48 (*0.07*)
		r=3	0.81 (*0.09*)	0.94 (*0.11*)	18.99 (*0.51*)	1.69 (*0.73*)	1.54 (*0.07*)
	t=2	r=1	0.94 (*0.04*)	1.88 (*0.10*)	19.90 (*0.56*)	1.22 (*0.20*)	1.46 (*0.07*)
		r=2	0.91 (*0.06*)	0.98 (*0.08*)	21.28 (*0.53*)	1.51 (*0.37*)	1.55 (*0.07*)
		r=3	0.90 (*0.04*)	2.22 (*0.12*)	21.62 (*0.61*)	1.35 (*0.20*)	1.36 (*0.07*)
(14a)	t=1	r=1	0.86 (*0.04*)	2.15 (*0.12*)	19.80 (*0.53*)	1.84 (*0.34*)	1.49 (*0.07*)
		r=2	0.79 (*0.08*)	1.02 (*0.12*)	19.68 (*0.52*)	1.58 (*0.65*)	1.60 (*0.07*)
		r=3	0.88 (*0.06*)	0.86 (*0.07*)	19.16 (*0.53*)	1.35 (*0.33*)	1.45 (*0.06*)
	t=2	r=1	0.92 (*0.05*)	2.13 (*0.13*)	20.13 (*0.53*)	1.21 (*0.23*)	1.43 (*0.07*)
		r=2	0.92 (*0.05*)	0.97 (*0.07*)	20.57 (*0.58*)	1.66 (*0.42*)	1.43 (*0.07*)
		r=3	0.94 (*0.03*)	1.93 (*0.09*)	20.79 (*0.54*)	1.45 (*0.22*)	1.61 (*0.08*)
(15a)	t=1	r=1	0.87 (*0.05*)	2.03 (*0.13*)	20.37 (*0.54*)	1.68 (*0.36*)	1.56 (*0.08*)
		r=2	0.67 (*0.05*)	1.18 (*0.09*)	20.55 (*0.60*)	3.22 (*1.36*)	1.38 (*0.06*)
		r=3	0.89 (*0.08*)	0.86 (*0.10*)	19.31 (*0.51*)	1.46 (*0.63*)	1.49 (*0.07*)
	t=2	r=1	0.93 (*0.03*)	1.84 (*0.09*)	20.66 (*0.54*)	1.33 (*0.21*)	1.51 (*0.08*)
		r=2	0.83 (*0.06*)	1.11 (*0.09*)	20.48 (*0.58*)	2.23 (*0.80*)	1.42 (*0.06*)
		r=3	0.90 (*0.04*)	1.96 (*0.11*)	21.78 (*0.63*)	1.71 (*0.35*)	1.47 (*0.07*)
(16a)	t=1	r=1	0.86 (*0.04*)	2.07 (*0.12*)	19.58 (*0.51*)	1.59 (*0.30*)	1.45 (*0.07*)
		r=2	0.82 (*0.08*)	1.05 (*0.11*)	19.75 (*0.54*)	1.55 (*0.51*)	1.53 (*0.08*)
		r=3	0.73 (*0.06*)	1.03 (*0.09*)	18.93 (*0.50*)	3.32 (*1.20*)	1.48 (*0.07*)
	t=2	r=1	0.91 (*0.04*)	2.07 (*0.13*)	19.74 (*0.51*)	1.56 (*0.28*)	1.44 (*0.07*)
		r=2	0.87 (*0.06*)	0.98 (*0.09*)	20.71 (*0.53*)	2.21 (*0.78*)	1.52 (*0.07*)
		r=3	0.88 (*0.04*)	1.97 (*0.11*)	21.06 (*0.60*)	1.73 (*0.36*)	1.42 (*0.07*)

Table 8: Posterior means of the measurement model parameters (μ, ϕ) and inflation proportion (π). Posterior standard deviations are provided between brackets.

Scenario			$\hat{\pi}$	$\hat{\mu}^B$	$\hat{\mu}^S$	$\hat{\phi}^B$	$\hat{\phi}^S$
(1b)	t=1	r=1	0.46 (*0.03*)	2.24 (*0.17*)	10.33 (*0.30*)	2.28 (*0.68*)	1.56 (*0.08*)
		r=2	0.59 (*0.10*)	0.81 (*0.15*)	9.93 (*0.29*)	1.17 (*0.52*)	1.55 (*0.08*)
		r=3	0.62 (*0.11*)	0.97 (*0.16*)	11.33 (*0.31*)	1.85 (*1.32*)	1.55 (*0.09*)
	t=2	r=1	0.55 (*0.15*)	0.68 (*0.19*)	9.90 (*0.27*)	1.07 (*0.80*)	1.56 (*0.08*)
		r=2	0.47 (*0.03*)	2.08 (*0.15*)	10.52 (*0.29*)	2.05 (*0.70*)	1.48 (*0.08*)
		r=3	0.49 (*0.04*)	2.02 (*0.17*)	9.83 (*0.30*)	1.67 (*0.50*)	1.34 (*0.08*)
	t=3	r=1	0.57 (*0.03*)	3.12 (*0.18*)	9.44 (*0.27*)	1.62 (*0.31*)	1.43 (*0.08*)
		r=2	0.58 (*0.11*)	1.69(*0.26*)	11.23 (*0.33*)	1.08 (*0.39*)	1.41 (*0.08*)
		r=3	0.58 (*0.05*)	2.32 (*0.23*)	9.35 (*0.26*)	1.02 (*0.29*)	1.58 (*0.09*)
	t=4	r=1	0.58 (*0.14*)	0.90 (*0.21*)	9.72 (*0.26*)	1.39 (*0.93*)	1.50 (*0.08*)
		r=2	0.40 (*0.06*)	1.06 (*0.16*)	10.62 (*0.30*)	2.21 (*1.23*)	1.59 (*0.08*)
		r=3	0.53 (*0.07*)	0.95 (*0.12*)	10.09 (*0.30*)	1.73 (*0.73*)	1.40 (*0.07*)
(2b)	t=1	r=1	0.51 (*0.04*)	2.00 (*0.17*)	10.06 (*0.29*)	1.70 (*0.53*)	1.52 (*0.08*)
		r=2	0.57 (*0.09*)	0.84 (*0.14*)	10.03 (*0.30*)	1.03 (*0.50*)	1.56 (*0.09*)
		r=3	0.71 (*0.10*)	0.92 (*0.13*)	10.84 (*0.30*)	1.29 (*0.53*)	1.45 (*0.08*)
	t=2	r=1	0.53 (*0.11*)	0.82 (*0.17*)	9.80 (*0.26*)	0.81 (*0.38*)	1.52 (*0.09*)
		r=2	0.49 (*0.05*)	2.02 (*0.18*)	10.99 (*0.30*)	1.65 (*0.55*)	1.48 (*0.08*)
		r=3	0.47 (*0.04*)	2.02 (*0.18*)	10.25 (*0.28*)	1.95 (*0.58*)	1.50 (*0.07*)
	t=3	r=1	0.59 (*0.03*)	3.04 (*0.17*)	9.93 (*0.30*)	1.84 (*0.35*)	1.48 (*0.07*)
		r=2	0.53 (*0.04*)	1.94 (*0.16*)	11.14 (*0.32*)	1.75 (*0.53*)	1.46 (*0.07*)
		r=3	0.47 (*0.02*)	3.27 (*0.18*)	9.87 (*0.25*)	1.91 (*0.37*)	1.47 (*0.08*)
	t=4	r=1	0.47 (*0.06*)	0.86 (*0.10*)	10.35 (*0.28*)	2.04 (*0.95*)	1.65 (*0.09*)
		r=2	0.66 (*0.15*)	0.73 (*0.17*)	10.77 (*0.31*)	0.66 (*0.35*)	1.43 (*0.07*)
		r=3	0.60 (*0.08*)	0.92 (*0.13*)	10.44 (*0.28*)	1.38 (*0.55*)	1.50 (*0.08*)

Table 9: Posterior means of the measurement model parameters (μ, ϕ) and inflation proportion (π). Posterior standard deviations are provided between brackets.

Scenario			$\hat{\pi}$	$\hat{\mu}^B$	$\hat{\mu}^S$	$\hat{\phi}^B$	$\hat{\phi}^S$
(3b)	t=1	r=1	0.47 (*0.03*)	2.01 (*0.11*)	10.32 (*0.30*)	2.28 (*0.61*)	1.59 (*0.08*)
		r=2	0.59 (*0.11*)	0.70 (*0.13*)	9.50 (*0.20*)	1.72 (*0.32*)	1.51 (*0.08*)
		r=3	0.60 (*0.12*)	1.02 (*0.12*)	11.10 (*0.33*)	1.90 (*1.42*)	1.55 (*0.10*)
	t=2	r=1	0.45 (*0.15*)	0.60 (*0.19*)	9.23 (*0.30*)	1.04 (*0.80*)	1.51 (*0.02*)
		r=2	0.49 (*0.05*)	2.01 (*0.13*)	9.30 (*0.30*)	2.55 (*0.75*)	1.80 (*0.02*)
		r=3	0.48 (*0.05*)	2.09 (*0.13*)	9.91 (*0.30*)	1.57 (*0.70*)	1.20 (*0.03*)
	t=3	r=1	0.57 (*0.05*)	2.10 (*0.18*)	10.44 (*0.27*)	1.52 (*0.31*)	1.52 (*0.04*)
		r=2	0.58 (*0.12*)	1.31(*0.16*)	10.11 (*0.33*)	1.38 (*0.25*)	1.51 (*0.01*)
		r=3	0.51 (*0.09*)	2.34 (*0.23*)	11.38 (*0.26*)	1.22 (*0.25*)	1.28 (*0.03*)
	t=4	r=1	0.51 (*0.11*)	1.04 (*0.21*)	9.97 (*0.26*)	1.21 (*0.87*)	1.50 (*0.07*)
		r=2	0.43 (*0.03*)	1.16 (*0.13*)	10.62 (*0.58*)	2.21 (*1.13*)	1.59 (*0.07*)
		r=3	0.54 (*0.03*)	1.05 (*0.09*)	10.09 (*0.41*)	1.73 (*0.44*)	1.40 (*0.07*)
(4b)	t=1	r=1	0.48 (*0.03*)	2.15 (*0.15*)	9.49 (*0.27*)	2.08 (*0.57*)	1.67 (*0.09*)
		r=2	0.61 (*0.14*)	0.77 (*0.17*)	10.30 (*0.29*)	1.41 (*0.93*)	1.51 (*0.08*)
		r=3	0.58 (*0.07*)	1.11 (*0.13*)	10.68 (*0.30*)	2.27 (*1.02*)	1.64 (*0.09*)
	t=2	r=1	0.41 (*0.05*)	1.04 (*0.13*)	9.78 (*0.29*)	2.56 (*1.33*)	1.51 (*0.08*)
		r=2	0.47 (*0.04*)	2.07 (*0.18*)	10.73 (*0.30*)	1.87 (*0.55*)	1.49 (*0.08*)
		r=3	0.55 (*0.06*)	1.86 (*0.20*)	9.97 (*0.29*)	1.33 (*0.46*)	1.49 (*0.08*)
	t=3	r=1	0.59 (*0.03*)	3.11 (*0.17*)	9.70 (*0.28*)	1.69 (*0.29*)	1.47 (*0.09*)
		r=2	0.50 (*0.04*)	2.36 (*0.18*)	11.03 (*0.34*)	1.63 (*0.44*)	1.45 (*0.07*)
		r=3	0.48 (*0.03*)	2.97 (*0.22*)	10.30 (*0.29*)	1.41 (*0.28*)	1.57 (*0.08*)
	t=4	r=1	0.51 (*0.08*)	0.98 (*0.14*)	9.95 (*0.29*)	1.57 (*0.72*)	1.43 (*0.08*)
		r=2	0.60 (*0.13*)	0.65 (*0.14*)	11.26 (*0.31*)	0.66 (*0.32*)	1.62 (*0.09*)
		r=3	0.68 (*0.14*)	0.73 (*0.17*)	10.40 (*0.29*)	0.77 (*0.52*)	1.38 (*0.07*)

Table 10: Posterior means of the measurement model parameters (μ, ϕ) and inflation proportion (π). Posterior standard deviations are provided between brackets.

Scenario			$\hat{\pi}$	$\hat{\mu}^B$	$\hat{\mu}^S$	$\hat{\phi}^B$	$\hat{\phi}^S$
(5b)	t=1	r=1	0.48 (*0.03*)	2.15 (*0.15*)	9.49 (*0.27*)	2.08 (*0.57*)	1.67 (*0.09*)
		r=2	0.61 (*0.14*)	0.77 (*0.17*)	10.30 (*0.29*)	1.41 (*0.93*)	1.51 (*0.08*)
		r=3	0.58 (*0.07*)	1.11 (*0.13*)	10.68 (*0.30*)	2.27 (*1.02*)	1.64 (*0.09*)
	t=2	r=1	0.41 (*0.05*)	1.04 (*0.13*)	9.78 (*0.29*)	2.56 (*1.33*)	1.51 (*0.08*)
		r=2	0.47 (*0.04*)	2.07 (*0.18*)	10.73 (*0.30*)	1.87 (*0.55*)	1.49 (*0.08*)
		r=3	0.55 (*0.06*)	1.86 (*0.20*)	9.97 (*0.29*)	1.33 (*0.46*)	1.49 (*0.08*)
	t=3	r=1	0.59 (*0.03*)	3.11 (*0.17*)	9.70 (*0.28*)	1.69 (*0.29*)	1.57 (*0.09*)
		r=2	0.50 (*0.04*)	2.36 (*0.18*)	11.03 (*0.34*)	1.63 (*0.44*)	1.45 (*0.07*)
		r=3	0.48 (*0.03*)	2.97 (*0.22*)	10.30 (*0.29*)	1.41 (*0.28*)	1.57 (*0.08*)
	t=4	r=1	0.51 (*0.08*)	0.98 (*0.14*)	9.95 (*0.29*)	1.57 (*0.72*)	1.43 (*0.08*)
		r=2	0.60 (*0.13*)	0.65 (*0.14*)	11.26 (*0.31*)	0.66 (*0.32*)	1.62 (*0.09*)
		r=3	0.68 (*0.14*)	0.73 (*0.17*)	10.40 (*0.29*)	0.77 (*0.52*)	1.38 (*0.07*)
(6b)	t=1	r=1	0.51 (*0.05*)	1.90 (*0.19*)	19.42 (*0.53*)	1.14 (*0.30*)	1.46 (*0.09*)
		r=2	0.58 (*0.08*)	0.88 (*0.13*)	20.08 (*0.55*)	1.20 (*0.48*)	1.35 (*0.08*)
		r=3	0.79 (*0.11*)	0.80 (*0.12*)	20.51 (*0.50*)	1.01 (*0.43*)	1.58 (*0.09*)
	t=2	r=1	0.37 (*0.06*)	1.12 (*0.16*)	20.13 (*0.54*)	3.24 (*1.83*)	1.51 (*0.08*)
		r=2	0.51 (*0.05*)	1.88 (*0.20*)	19.93 (*0.52*)	1.65 (*0.56*)	1.55 (*0.08*)
		r=3	0.50 (*0.05*)	1.89 (*0.19*)	21.00 (*0.56*)	1.32 (*0.40*)	1.54 (*0.08*)
	t=3	r=1	0.61 (*0.03*)	2.89 (*0.16*)	19.65 (*0.54*)	1.71 (*0.30*)	1.52 (*0.09*)
		r=2	0.55 (*0.05*)	1.88 (*0.18*)	18.81 (*0.52*)	1.34 (*0.35*)	1.52 (*0.07*)
		r=3	0.50 (*0.03*)	2.86 (*0.21*)	20.36 (*0.57*)	1.34 (*0.26*)	1.42 (*0.08*)
	t=4	r=1	0.52 (*0.07*)	1.01 (*0.13*)	19.81 (*0.53*)	1.90 (*0.90*)	1.54 (*0.08*)
		r=2	0.27 (*0.15*)	1.04 (*0.26*)	19.27 (*0.51*)	1.74 (*1.03*)	1.48 (*0.09*)
		r=3	0.56 (*0.06*)	1.01 (*0.12*)	19.51 (*0.54*)	1.52 (*0.59*)	1.39 (*0.07*)

Table 11: Posterior means of the measurement model parameters (μ, ϕ) and inflation proportion (π). Posterior standard deviations are provided between brackets.

Scenario			$\hat{\pi}$	$\hat{\mu}^B$	$\hat{\mu}^S$	$\hat{\phi}^B$	$\hat{\phi}^S$
(7b)	t=1	r=1	0.48 (*0.03*)	2.08 (*0.16*)	19.99 (*0.58*)	1.59 (*0.38*)	1.44 (*0.07*)
		r=2	0.45 (*0.05*)	1.07 (*0.11*)	18.54 (*0.55*)	2.07 (*0.97*)	1.46 (*0.08*)
		r=3	0.59 (*0.07*)	1.04 (*0.13*)	20.63 (*0.58*)	2.34 (*1.33*)	1.61 (*0.09*)
	t=2	r=1	0.42 (*0.05*)	1.01 (*0.13*)	19.78 (*0.53*)	1.74 (*0.72*)	1.55 (*0.08*)
		r=2	0.63 (*0.11*)	1.62 (*0.28*)	20.50 (*0.52*)	0.80 (*0.29*)	1.54 (*0.06*)
		r=3	0.61 (*0.08*)	1.60 (*0.22*)	20.64 (*0.54*)	0.90 (*0.28*)	1.49 (*0.07*)
	t=3	r=1	0.57 (*0.03*)	3.32 (*0.17*)	18.56 (*0.53*)	1.80 (*0.30*)	1.37 (*0.06*)
		r=2	0.54 (*0.04*)	1.86 (*0.15*)	18.90 (*0.53*)	1.49 (*0.42*)	1.47 (*0.08*)
		r=3	0.53 (*0.03*)	2.77 (*0.18*)	21.47 (*0.57*)	1.62 (*0.35*)	1.47 (*0.08*)
	t=4	r=1	0.50 (*0.08*)	0.95 (*0.15*)	20.41 (*0.56*)	1.79 (*1.14*)	1.47 (*0.07*)
		r=2	0.43 (*0.07*)	0.98 (*0.16*)	18.66 (*0.52*)	1.47 (*0.76*)	1.48 (*0.08*)
		r=3	0.50 (*0.07*)	0.99 (*0.14*)	18.94 (*0.53*)	1.87 (*0.91*)	1.50 (*0.07*)
(8b)	t=1	r=1	0.54 (*0.05*)	1.81 (*0.17*)	19.56 (*0.49*)	1.15 (*0.28*)	1.68 (*0.08*)
		r=2	0.61 (*0.12*)	0.83 (*0.15*)	18.54 (*0.54*)	1.24 (*0.58*)	1.41 (*0.07*)
		r=3	0.67 (*0.10*)	0.83 (*0.13*)	20.97 (*0.58*)	1.30 (*0.55*)	1.44 (*0.07*)
	t=2	r=1	0.74 (*0.18*)	0.59 (*0.19*)	20.21 (*0.59*)	0.53 (*0.38*)	1.34 (*0.06*)
		r=2	0.49 (*0.03*)	2.09 (*0.15*)	19.41 (*0.54*)	1.93 (*0.49*)	1.47 (*0.07*)
		r=3	0.59 (*0.06*)	1.71 (*0.19*)	21.22 (*0.57*)	1.00 (*0.27*)	1.60 (*0.08*)
	t=3	r=1	0.65 (*0.04*)	2.76 (*0.19*)	18.46 (*0.51*)	1.24 (*0.21*)	1.51 (*0.07*)
		r=2	0.60 (*0.08*)	1.62 (*0.19*)	19.43 (*0.52*)	1.22 (*0.37*)	1.45 (*0.07*)
		r=3	0.47 (*0.03*)	2.86 (*0.19*)	20.50 (*0.55*)	1.49 (*0.32*)	1.50 (*0.06*)
	t=4	r=1	0.51 (*0.06*)	1.01 (*0.13*)	19.24 (*0.51*)	2.63 (*1.42*)	1.52 (*0.07*)
		r=2	0.46 (*0.09*)	0.91 (*0.16*)	18.70 (*0.51*)	1.60 (*0.91*)	1.57 (*0.07*)
		r=3	0.64 (*0.13*)	0.83 (*0.17*)	20.39 (*0.59*)	1.03 (*0.59*)	1.36 (*0.06*)

Table 12: Posterior means of the measurement model parameters (μ, ϕ) and inflation proportion (π). Posterior standard deviations are provided between brackets.

Scenario			$\hat{\pi}$	$\hat{\mu}^B$	$\hat{\mu}^S$	$\hat{\phi}^B$	$\hat{\phi}^S$
(9b)	t=1	r=1	0.95 (*0.03*)	1.89 (*0.09*)	9.59 (*0.26*)	1.54 (*0.24*)	1.42 (*0.08*)
		r=2	0.79 (*0.09*)	1.03 (*0.12*)	9.73 (*0.28*)	2.22 (*1.09*)	1.56 (*0.08*)
		r=3	0.94 (*0.04*)	0.87 (*0.06*)	11.14 (*0.33*)	1.66 (*0.40*)	1.39 (*0.07*)
	t=2	r=1	0.76 (*0.07*)	1.10 (*0.11*)	10.03 (*0.30*)	1.93 (*0.78*)	1.40 (*0.08*)
		r=2	0.87 (*0.04*)	2.17 (*0.12*)	11.32 (*0.33*)	1.61 (*0.34*)	1.53 (*0.08*)
		r=3	0.90 (*0.04*)	2.07 (*0.11*)	9.71 (*0.29*)	1.54 (*0.29*)	1.51 (*0.08*)
	t=3	r=1	0.92 (*0.04*)	2.82 (*0.16*)	9.97 (*0.27*)	1.37 (*0.21*)	1.53 (*0.08*)
		r=2	0.85 (*0.06*)	1.88 (*0.13*)	10.93 (*0.30*)	1.39 (*0.36*)	1.56 (*0.09*)
		r=3	0.89 (*0.05*)	2.60 (*0.17*)	9.65 (*0.29*)	1.12 (*0.20*)	1.52 (*0.08*)
	t=4	r=1	0.75 (*0.07*)	1.00 (*0.11*)	10.37 (*0.29*)	2.11 (*1.76*)	1.61 (*0.08*)
		r=2	0.90 (*0.07*)	1.03 (*0.08*)	10.50 (*0.27*)	1.72 (*0.58*)	1.66 (*0.09*)
		r=3	0.85 (*0.08*)	0.99 (*0.10*)	10.07 (*0.28*)	1.26 (*0.37*)	1.56 (*0.08*)
(10b)	t=1	r=1	0.94 (*0.04*)	2.01 (*0.11*)	9.67 (*0.28*)	1.39 (*0.25*)	1.53 (*0.08*)
		r=2	0.82 (*0.09*)	0.95 (*0.12*)	9.88 (*0.27*)	1.30 (*0.51*)	1.53 (*0.08*)
		r=3	0.83 (*0.07*)	1.15 (*0.10*)	10.93 (*0.31*)	1.68 (*0.50*)	1.36 (*0.07*)
	t=2	r=1	0.77 (*0.09*)	1.00 (*0.11*)	9.59 (*0.26*)	1.47 (*0.67*)	1.62 (*0.09*)
		r=2	0.88 (*0.04*)	2.05 (*0.12*)	11.60 (*0.32*)	1.67 (*0.35*)	1.44 (*0.07*)
		r=3	0.87 (*0.04*)	2.19 (*0.11*)	9.80 (*0.24*)	1.73 (*0.33*)	1.62 (*0.09*)
	t=3	r=1	0.85 (*0.03*)	3.13 (*0.13*)	9.97 (*0.28*)	1.99 (*0.28*)	1.50 (*0.09*)
		r=2	0.87 (*0.05*)	1.89 (*0.13*)	11.31 (*0.32*)	1.18 (*0.22*)	1.51 (*0.09*)
		r=3	0.77 (*0.03*)	2.96 (*0.14*)	10.25 (*0.30*)	1.60 (*0.25*)	1.46 (*0.08*)
	t=4	r=1	0.78 (*0.06*)	1.04 (*0.10*)	9.46 (*0.26*)	2.28 (*1.80*)	1.60 (*0.09*)
		r=2	0.83 (*0.08*)	1.07 (*0.10*)	11.13 (*0.40*)	3.11 (*1.53*)	1.56 (*0.09*)
		r=3	0.74 (*0.07*)	1.06 (*0.10*)	10.02 (*0.28*)	2.66 (*1.16*)	1.42 (*0.08*)

Table 13: Posterior means of the measurement model parameters (μ, ϕ) and inflation proportion (π). Posterior standard deviations are provided between brackets.

Scenario			$\hat{\pi}$	$\hat{\mu}^B$	$\hat{\mu}^S$	$\hat{\phi}^B$	$\hat{\phi}^S$
(11b)	t=1	r=1	0.91 (*0.04*)	1.90 (*0.11*)	9.90 (*0.26*)	1.40 (*0.24*)	1.55 (*0.09*)
		r=2	0.82 (*0.08*)	0.89 (*0.10*)	10.13 (*0.28*)	1.71 (*0.57*)	1.48 (*0.09*)
		r=3	0.93 (*0.06*)	0.98 (*0.09*)	11.11 (*0.33*)	1.55 (*0.48*)	1.50 (*0.09*)
	t=2	r=1	0.72 (*0.06*)	1.12 (*0.09*)	10.09 (*0.30*)	2.01 (*0.78*)	1.44 (*0.09*)
		r=2	0.92 (*0.04*)	1.78 (*0.10*)	10.98 (*0.33*)	1.43 (*0.27*)	1.56 (*0.08*)
		r=3	0.89 (*0.05*)	1.95 (*0.12*)	9.69 (*0.29*)	1.51 (*0.34*)	1.57 (*0.09*)
	t=3	r=1	0.91 (*0.03*)	3.19 (*0.14*)	9.38 (*0.27*)	1.52 (*0.24*)	1.56 (*0.08*)
		r=2	0.74 (*0.04*)	2.28 (*0.14*)	11.28 (*0.30*)	1.91 (*0.46*)	1.50 (*0.09*)
		r=3	0.76 (*0.03*)	3.16 (*0.15*)	9.88 (*0.29*)	1.94 (*0.32*)	1.49 (*0.09*)
	t=4	r=1	0.84 (*0.10*)	0.88 (*0.12*)	9.67 (*0.29*)	1.68 (*0.96*)	1.38 (*0.07*)
		r=2	0.83 (*0.09*)	1.07 (*0.12*)	10.87 (*0.27*)	2.34 (*1.18*)	1.48 (*0.08*)
		r=3	0.80 (*0.07*)	0.91 (*0.09*)	9.38 (*0.28*)	2.01 (*0.73*)	1.64 (*0.09*)
(12b)	t=1	r=1	0.88 (*0.04*)	2.02 (*0.11*)	9.88 (*0.29*)	1.59 (*0.32*)	1.32 (*0.07*)
		r=2	0.75 (*0.07*)	1.08 (*0.10*)	9.72 (*0.30*)	2.30 (*0.84*)	1.45 (*0.08*)
		r=3	0.86 (*0.06*)	1.03 (*0.08*)	11.07 (*0.33*)	2.24 (*0.72*)	1.42 (*0.09*)
	t=2	r=1	0.93 (*0.05*)	0.94 (*0.07*)	9.96 (*0.29*)	0.93 (*0.16*)	1.48 (*0.09*)
		r=2	0.85 (*0.04*)	2.14 (*0.13*)	10.75 (*0.30*)	1.85 (*0.37*)	1.56 (*0.08*)
		r=3	0.93 (*0.04*)	1.89 (*0.12*)	9.83 (*0.29*)	1.23 (*0.19*)	1.54 (*0.09*)
	t=3	r=1	0.90 (*0.04*)	2.96 (*0.15*)	9.92 (*0.27*)	1.50 (*0.25*)	1.52 (*0.09*)
		r=2	0.89 (*0.06*)	1.83 (*0.14*)	11.23 (*0.33*)	1.08 (*0.25*)	1.42 (*0.08*)
		r=3	0.82 (*0.04*)	2.97 (*0.17*)	9.65 (*0.28*)	1.32 (*0.21*)	1.54 (*0.09*)
	t=4	r=1	0.88 (*0.07*)	0.89 (*0.09*)	9.70 (*0.26*)	1.09 (*0.30*)	1.50 (*0.08*)
		r=2	0.82 (*0.06*)	1.18 (*0.09*)	11.03 (*0.32*)	2.39 (*0.80*)	1.43 (*0.07*)
		r=3	0.85 (*0.10*)	0.88 (*0.12*)	9.58 (*0.29*)	1.28 (*0.52*)	1.40 (*0.08*)

Table 14: Posterior means of the measurement model parameters (μ, ϕ) and inflation proportion (π). Posterior standard deviations are provided between brackets.

Scenario			$\hat{\pi}$	$\hat{\mu}^B$	$\hat{\mu}^S$	$\hat{\phi}^B$	$\hat{\phi}^S$
(13b)	t=1	r=1	0.95 (*0.10*)	1.89 (*0.11*)	20.99 (*0.55*)	1.44 (*0.24*)	1.53 (*0.07*)
		r=2	0.86 (*0.08*)	0.98 (*0.10*)	18.77 (*0.52*)	1.70 (*0.62*)	1.53 (*0.07*)
		r=3	0.92 (*0.06*)	0.96 (*0.08*)	21.02 (*0.55*)	1.75 (*0.56*)	1.57 (*0.08*)
	t=2	r=1	0.76 (*0.08*)	1.05 (*0.12*)	19.65 (*0.54*)	2.53 (*1.15*)	1.58 (*0.08*)
		r=2	0.89 (*0.04*)	2.19 (*0.11*)	20.18 (*0.54*)	1.63 (*0.28*)	1.50 (*0.07*)
		r=3	0.93 (*0.04*)	1.95 (*0.11*)	20.90 (*0.59*)	1.42 (*0.25*)	1.48 (*0.07*)
	t=3	r=1	0.92 (*0.04*)	2.84 (*0.15*)	18.93 (*0.54*)	1.28 (*0.20*)	1.46 (*0.07*)
		r=2	0.79 (*0.05*)	1.87 (*0.13*)	18.92 (*0.52*)	1.47 (*0.33*)	1.47 (*0.07*)
		r=3	0.84 (*0.05*)	2.75 (*0.18*)	21.86 (*0.64*)	1.36 (*0.28*)	1.47 (*0.07*)
	t=4	r=1	0.80 (*0.09*)	1.00 (*0.12*)	19.36 (*0.51*)	1.32 (*0.44*)	1.60 (*0.07*)
		r=2	0.91 (*0.06*)	0.93 (*0.07*)	19.12 (*0.50*)	1.77 (*0.60*)	1.50 (*0.07*)
		r=3	0.87 (*0.08*)	0.94 (*0.11*)	19.37 (*0.53*)	1.29 (*0.46*)	1.56 (*0.07*)
(14b)	t=1	r=1	0.89 (*0.04*)	2.08 (*0.11*)	20.06 (*0.55*)	1.98 (*0.42*)	1.53 (*0.08*)
		r=2	0.85 (*0.08*)	0.90 (*0.10*)	19.10 (*0.53*)	1.49 (*0.49*)	1.42 (*0.07*)
		r=3	0.93 (*0.06*)	1.05 (*0.09*)	20.65 (*0.57*)	1.53 (*0.42*)	1.50 (*0.08*)
	t=2	r=1	0.77 (*0.08*)	1.13 (*0.12*)	19.85 (*0.53*)	2.05 (*0.72*)	1.43 (*0.07*)
		r=2	0.92 (*0.04*)	1.90 (*0.11*)	19.37 (*0.50*)	1.35 (*0.21*)	1.58 (*0.07*)
		r=3	0.94 (*0.04*)	1.90 (*0.12*)	21.09 (*0.56*)	1.37 (*0.24*)	1.56 (*0.07*)
	t=3	r=1	0.91 (*0.03*)	2.86 (*0.14*)	18.48 (*0.51*)	1.37 (*0.20*)	1.46 (*0.07*)
		r=2	0.85 (*0.07*)	1.77 (*0.15*)	19.29 (*0.49*)	1.10 (*0.28*)	1.62 (*0.08*)
		r=3	0.75 (*0.03*)	2.88 (*0.14*)	21.21 (*0.58*)	1.73 (*0.29*)	1.38 (*0.06*)
	t=4	r=1	0.92 (*0.07*)	0.84 (*0.08*)	20.07 (*0.56*)	1.28 (*0.38*)	1.42 (*0.06*)
		r=2	0.88 (*0.07*)	0.98 (*0.08*)	19.33 (*0.51*)	1.66 (*0.51*)	1.56 (*0.07*)
		r=3	0.74 (*0.07*)	1.06 (*0.10*)	20.71 (*0.54*)	1.88 (*0.66*)	1.44 (*0.06*)

Table 15: Posterior means of the measurement model parameters (μ, ϕ) and inflation proportion (π). Posterior standard deviations are provided between brackets.

Scenario			$\hat{\pi}$	$\hat{\mu}^B$	$\hat{\mu}^S$	$\hat{\phi}^B$	$\hat{\phi}^S$
(15b)	t=1	r=1	0.88 (*0.04*)	2.14 (*0.10*)	20.92 (*0.57*)	1.86 (*0.33*)	1.44 (*0.06*)
		r=2	0.81 (*0.08*)	0.98 (*0.11*)	18.03 (*0.51*)	1.28 (*0.39*)	1.48 (*0.08*)
		r=3	0.84 (*0.07*)	1.05 (*0.10*)	20.79 (*0.58*)	1.56 (*0.47*)	1.51 (*0.08*)
	t=2	r=1	0.77 (*0.09*)	1.05 (*0.11*)	19.20 (*0.52*)	2.90 (*1.39*)	1.44 (*0.07*)
		r=2	0.85 (*0.04*)	2.10 (*0.12*)	20.02 (*0.52*)	2.00 (*0.39*)	1.54 (*0.07*)
		r=3	0.92 (*0.04*)	2.02 (*0.11*)	20.40 (*0.54*)	1.36 (*0.23*)	1.59 (*0.08*)
	t=3	r=1	0.89 (*0.03*)	3.12 (*0.15*)	18.45 (*0.48*)	1.49 (*0.22*)	1.58 (*0.07*)
		r=2	0.87 (*0.05*)	1.83 (*0.12*)	19.38 (*0.59*)	1.23 (*0.22*)	1.37 (*0.07*)
		r=3	0.82 (*0.03*)	2.93 (*0.15*)	21.37 (*0.60*)	1.45 (*0.24*)	1.44 (*0.07*)
	t=4	r=1	0.77 (*0.08*)	1.04 (*0.11*)	19.92 (*0.56*)	2.32 (*0.93*)	1.58 (*0.08*)
		r=2	0.89 (*0.06*)	1.08 (*0.09*)	19.15 (*0.54*)	1.73 (*0.47*)	1.53 (*0.07*)
		r=3	0.91 (*0.07*)	0.84 (*0.08*)	19.57 (*0.53*)	1.50 (*0.49*)	1.41 (*0.07*)
(16b)	t=1	r=1	0.91 (*0.04*)	1.85 (*0.10*)	20.27 (*0.53*)	1.61 (*0.29*)	1.53 (*0.07*)
		r=2	0.68 (*0.06*)	1.11 (*0.09*)	19.11 (*0.50*)	3.50 (*1.47*)	1.66 (*0.08*)
		r=3	0.88 (*0.08*)	1.09 (*0.10*)	20.30 (*0.53*)	1.32 (*0.35*)	1.47 (*0.08*)
	t=2	r=1	0.87 (*0.08*)	0.99 (*0.11*)	19.36 (*0.53*)	1.23 (*0.37*)	1.44 (*0.07*)
		r=2	0.95 (*0.03*)	1.89 (*0.09*)	19.41 (*0.50*)	1.29 (*0.19*)	1.69 (*0.09*)
		r=3	0.91 (*0.04*)	1.90 (*0.12*)	21.14 (*0.62*)	1.38 (*0.24*)	1.35 (*0.07*)
	t=3	r=1	0.89 (*0.03*)	3.10 (*0.15*)	18.37 (*0.51*)	1.45 (*0.21*)	1.51 (*0.07*)
		r=2	0.80 (*0.05*)	2.17 (*0.14*)	18.36(*0.53*)	1.71 (*0.39*)	1.43 (*0.08*)
		r=3	0.77 (*0.03*)	3.01 (*0.15*)	20.97 (*0.56*)	1.67 (*0.27*)	1.47 (*0.07*)
	t=4	r=1	0.85 (*0.08*)	0.97 (*0.10*)	19.98 (*0.55*)	1.55 (*0.51*)	1.55 (*0.08*)
		r=2	0.87 (*0.06*)	1.03 (*0.08*)	19.40 (*0.55*)	1.68 (*0.47*)	1.40 (*0.06*)
		r=3	0.87 (*0.09*)	0.89 (*0.10*)	19.40 (*0.55*)	1.30 (*0.51*)	1.48 (*0.07*)

Table 16: Posterior means and posterior standard deviations (between brackets) of the spatial and temporal transition probabilities.

Scenario	$\hat{q}_1$	$\hat{w}_1$	Scenario	$\hat{q}_1$	$\hat{w}_1$
(1a)	0.31 (*0.01*)	0.40 (*0.01*)	(1b)	0.28 (*0.01*)	0.62 (*0.01*)
(2a)	0.33 (*0.01*)	0.80 (*0.01*)	(2b)	0.27 (*0.01*)	0.82 (*0.01*)
(3a)	0.68 (*0.01*)	0.41 (*0.01*)	(3b)	0.69 (*0.01*)	0.39 (*0.01*)
(4a)	0.68 (*0.01*)	0.79 (*0.01*)	(4b)	0.73 (*0.01*)	0.81 (*0.01*)
(5a)	0.30 (*0.01*)	0.41 (*0.01*)	(5b)	0.28 (*0.01*)	0.61 (*0.01*)
(6a)	0.33 (*0.01*)	0.80 (*0.01*)	(6b)	0.27 (*0.01*)	0.82 (*0.01*)
(7a)	0.68 (*0.01*)	0.41 (*0.01*)	(7b)	0.70 (*0.01*)	0.37 (*0.01*)
(8a)	0.68 (*0.01*)	0.79 (*0.01*)	(8b)	0.72 (*0.01*)	0.81 (*0.01*)
(9a)	0.31 (*0.01*)	0.41 (*0.01*)	(9b)	0.28 (*0.01*)	0.62 (*0.01*)
(10a)	0.33 (*0.01*)	0.81 (*0.01*)	(10b)	0.26 (*0.01*)	0.82 (*0.01*)
(11a)	0.67 (*0.01*)	0.41 (*0.01*)	(11b)	0.70 (*0.01*)	0.36 (*0.01*)
(12a)	0.69 (*0.01*)	0.79 (*0.01*)	(12b)	0.73 (*0.01*)	0.82 (*0.01*)
(13a)	0.30 (*0.01*)	0.40 (*0.01*)	(13b)	0.28 (*0.01*)	0.62 (*0.01*)
(14a)	0.33 (*0.01*)	0.81 (*0.01*)	(14b)	0.27 (*0.01*)	0.81 (*0.01*)
(15a)	0.68 (*0.01*)	0.41 (*0.01*)	(15b)	0.70 (*0.01*)	0.37 (*0.01*)
(16a)	0.68 (*0.01*)	0.79 (*0.01*)	(16b)	0.72 (*0.01*)	0.81 (*0.01*)

Table 17: Summary of the False Non Discovery Rate (in percentage %) evaluated for the three models at a fixed 5% False Discovery Rate.

Scenario		$FNDR_{iSeq}$	$FNDR_{MRF}$	$FNDR_{stMRF}$
(1a)	$t=1$	30.0	2.3	0.2
	$t=2$	28.9	3.7	0.7
(2a)	$t=1$	31.0	3.4	0.5
	$t=2$	28.9	3.1	0.5
(3a)	$t=1$	25.7	3.2	0.4
	$t=2$	31.3	2.6	0.8
(4a)	$t=1$	33.0	2.8	0.6
	$t=2$	32.4	4.2	0.5
(5a)	$t=1$	26.8	0.0	0.0
	$t=2$	33.6	0.2	0.1
(6a)	$t=1$	23.5	0.0	0.0
	$t=2$	29.5	0.0	0.1
(7a)	$t=1$	27.8	0.1	0.1
	$t=2$	34.4	0.0	0.0
(8a)	$t=1$	31.0	0.0	0.0
	$t=2$	26.5	0.0	0.0
(9a)	$t=1$	36.8	2.8	0.0
	$t=2$	33.6	3.2	0.0
(10a)	$t=1$	35.0	3.3	1.8
	$t=2$	29.5	3.5	1.6
(11a)	$t=1$	48.1	2.4	1.5
	$t=2$	43.9	3.7	2.2
(12a)	$t=1$	45.2	3.8	1.9
	$t=2$	41.1	2.3	0.9
(13a)	$t=1$	28.6	0.4	0.2
	$t=2$	36.4	0.7	0.5
(14a)	$t=1$	29.2	0.1	0.1
	$t=2$	37.9	0.1	0.2
(15a)	$t=1$	28.5	0.5	0.3
	$t=2$	33.4	0.5	0.4
(16a)	$t=1$	31.6	0.6	0.2
	$t=2$	39.1	0.4	0.2

Table 18: Summary of the False Non Discovery Rate (in percentage %) evaluated for the three models at a fixed 5% False Discovery Rate.

Scenario		$FNDR_{iSeq}$	$FNDR_{MRF}$	$FNDR_{stMRF}$
(1b)	$t=1$	34.2	1.9	0.5
	$t=2$	28.9	4.7	1.2
	$t=3$	33.9	30.5	3.8
	$t=4$	31.6	0.6	0.1
(2b)	$t=1$	37.0	1.7	0.5
	$t=2$	31.8	3.5	0.5
	$t=3$	38.5	36.4	0.5
	$t=4$	28.9	0.6	0.3
(3b)	$t=1$	35.1	1.7	0.4
	$t=2$	30.1	2.8	0.3
	$t=3$	36.3	29.1	0.3
	$t=4$	28.9	0.9	0.2
(4b)	$t=1$	33.2	2.9	0.8
	$t=2$	29.7	6.6	0.3
	$t=3$	32.6	6.7	1.1
	$t=4$	33.0	0.1	0.1
(5b)	$t=1$	26.0	0.1	0.0
	$t=2$	27.6	0.3	0.1
	$t=3$	28.7	0.8	0.4
	$t=4$	23.3	0.0	0.0
(6b)	$t=1$	28.6	0.0	0.0
	$t=2$	22.9	0.0	0.0
	$t=3$	30.2	1.3	0.0
	$t=4$	29.6	0.0	0.0
(7b)	$t=1$	28.2	0.1	0.0
	$t=2$	25.5	0.1	0.0
	$t=3$	38.3	1.1	0.0
	$t=4$	27.4	0.1	0.0
(8b)	$t=1$	29.8	0.1	0.0
	$t=2$	40.5	0.0	0.0
	$t=3$	44.8	3.8	0.0
	$t=4$	42.7	0.0	0.0

Table 19: Summary of the False Non Discovery Rate (in percentage %) evaluated for the three models at a fixed 5% False Discovery Rate.

Scenario		$FNDR_{iSeq}$	$FNDR_{MRF}$	$FNDR_{stMRF}$
(9b)	$t=1$	44.8	2.6	0.3
	$t=2$	39.0	5.0	0.5
	$t=3$	41.6	16	0.6
	$t=4$	34.8	1.4	0.1
(10b)	$t=1$	36.6	3.5	1.1
	$t=2$	31.5	4.1	1.6
	$t=3$	43.7	18.1	3.0
	$t=4$	31.1	1.6	0.8
(11b)	$t=1$	42.1	2.8	1.8
	$t=2$	35.2	4.3	2.3
	$t=3$	43.5	32.6	13.4
	$t=4$	38.2	0.6	0.4
(12b)	$t=1$	35.71	3.9	1.7
	$t=2$	35.8	6.6	2.9
	$t=3$	39.2	19.6	3.8
	$t=4$	33.0	1.7	1.1
(13b)	$t=1$	36.4	0.3	0.3
	$t=2$	28.3	0.4	0.4
	$t=3$	40.7	1.2	0.9
	$t=4$	26.4	0.0	0.0
(14b)	$t=1$	34.6	0.3	0.1
	$t=2$	31.8	0.3	0.1
	$t=3$	41.2	0.9	0.4
	$t=4$	32.2	0.1	0.1
(15b)	$t=1$	35.9	0.5	0.3
	$t=2$	42.2	0.7	0.6
	$t=3$	42.1	1.7	1.0
	$t=4$	25.2	0.0	0.0
(16b)	$t=1$	33.4	0.3	0.3
	$t=2$	30.7	0.4	0.1
	$t=3$	40.6	1.8	0.2
	$t=4$	27.5	0.1	0.0

For the sake of brevity, we show only the plots of the chains of all the parameters for one scenario (16a). The other plots are similar and they share the same convergence behavior to the stationary distributions.

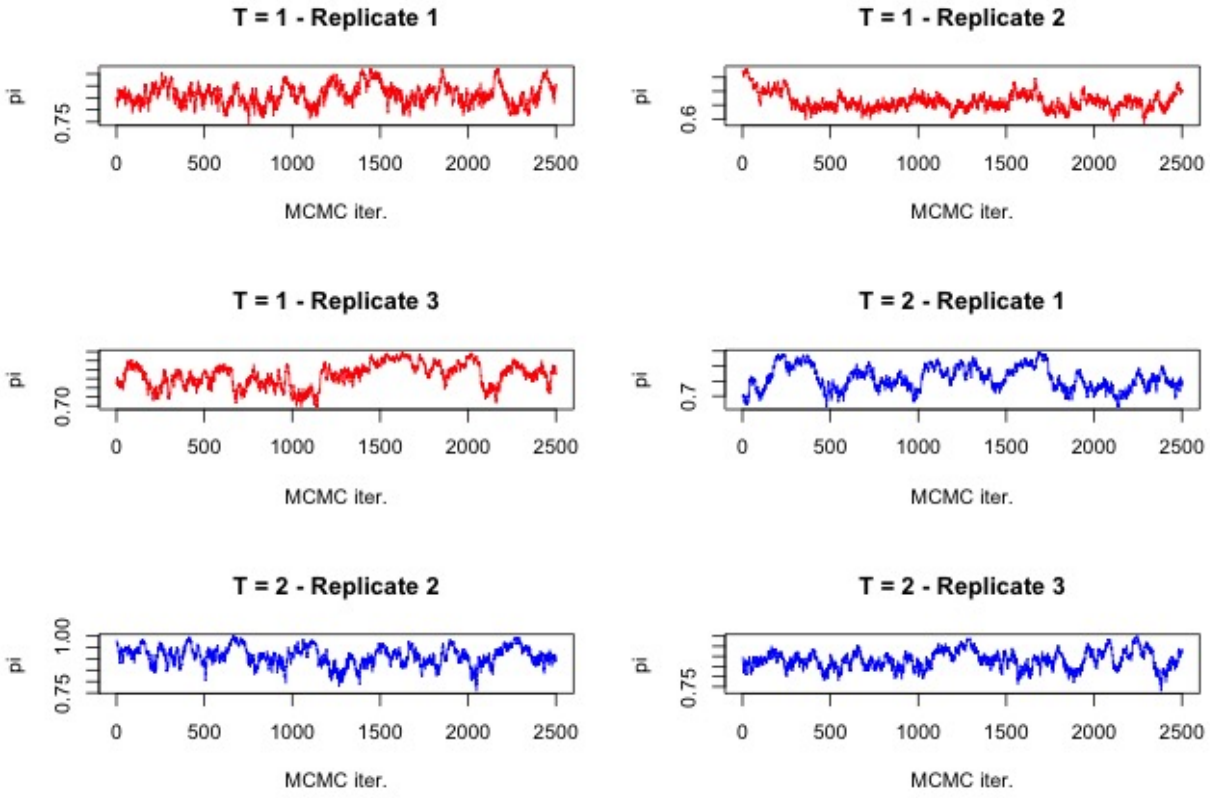

Figure 1: Plots of the chains of π for the scenario (16a).

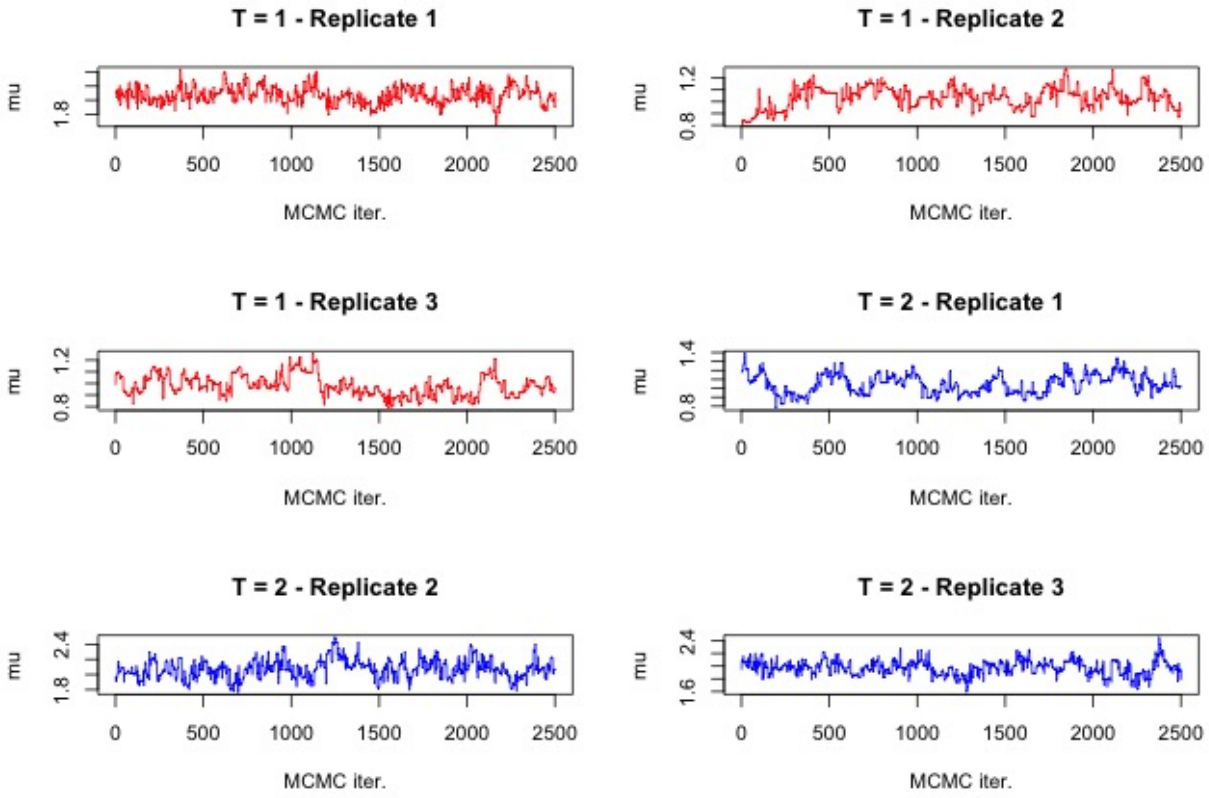

Figure 2: Plots of the chains of μ^B for the scenario (16a).

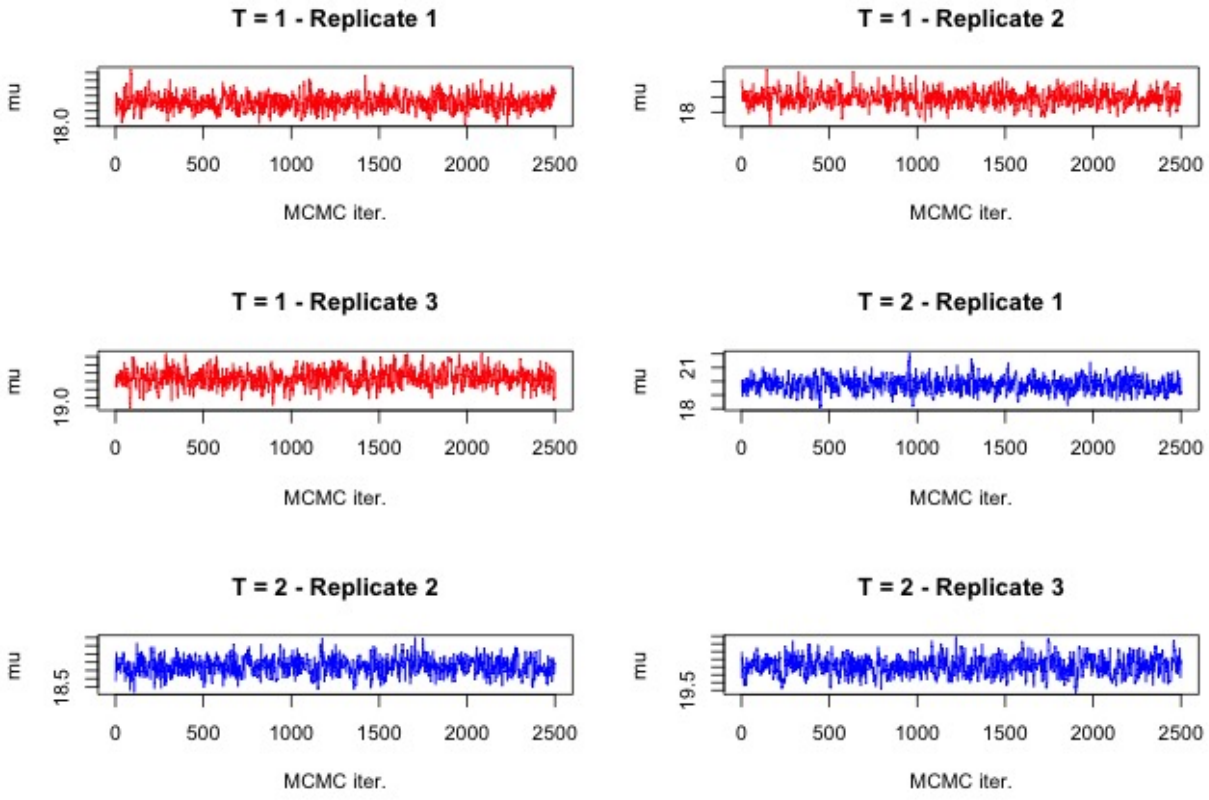

Figure 3: Plots of the chains of μ^S for the scenario (16a).

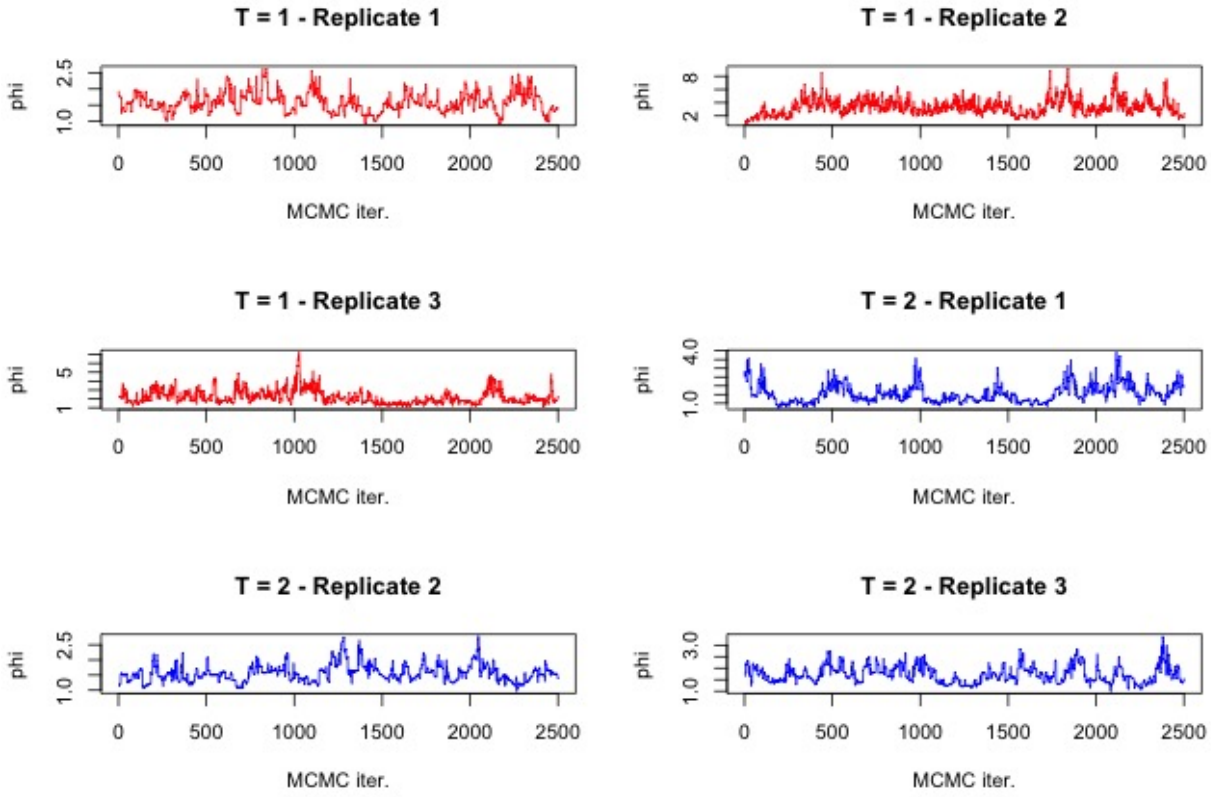

Figure 4: Plot of the chains of ϕ^B for the scenario (16a).

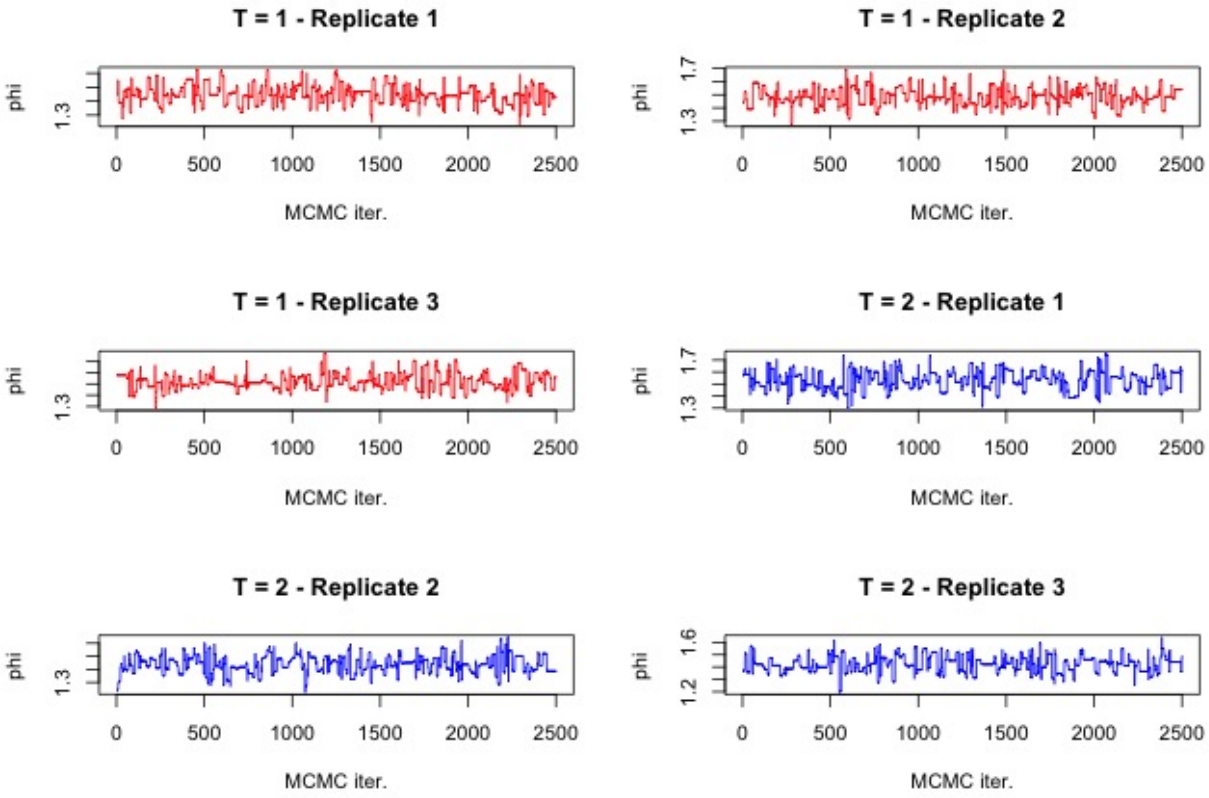

Figure 5: Plot of the chains of ϕ^S for the scenario (16a).

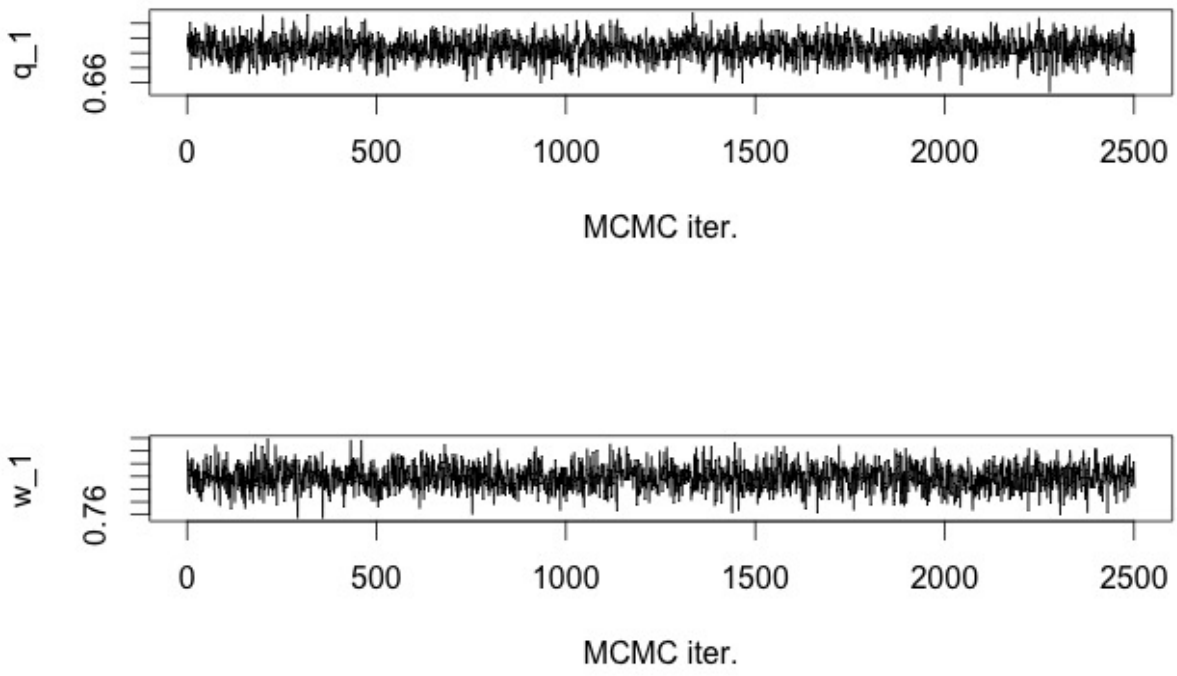

Figure 6: Plot of the chains of the transition probabilities q_1 and w_1 for the scenario (16a).