THE UNIVERSITY
of EDINBURGH

Statistical modeling of oscillating biological networks for structure inference and experimental design

Daniel Trejo Baños

Doctor of Philosophy

Institute for Adaptive and Neural Computation
School of Informatics
University of Edinburgh

2015

Abstract

Oscillations lie at the core of many biological processes, from the cell cycle, to circadian oscillations and developmental processes. They are essential to enable organisms to adapt to varying conditions in environmental cycles, from day/night to seasonal. Transcriptional regulatory networks are one of the mechanisms behind these biological oscillations. One of the main problems of computational systems biology is elucidating the interaction between biological components. A common mathematical abstraction is to represent these interactions as networks whose nodes are the reactive species and the interactions are edges. There is abundant literature dealing with the reconstruction of the network structure from steady-state gene expression measurements; still, there are lots of advancements to be made because of the complex nature of biological systems. Experimental design is another obstacle to overcome; we wish to perform experiments that help us best define the network structure according to our current knowledge of the system.

In the first chapters of this thesis we will focus on reconstructing the network structure of biological oscillators by explicitly leveraging the cyclical nature of the transcriptional signals. We present a method for reconstructing network interactions tailored to this special but important class of genetic circuits. The method is based on projecting the signal onto a set of oscillatory basis functions. We build a Bayesian hierarchical model within a frequency domain linear model in order to enforce sparsity and incorporate prior knowledge about the network structure. Experiments on real and simulated data show that the method can lead to substantial improvements over competing approaches if the oscillatory assumption is met, and remains competitive also in cases it is not.

Having defined a model for gene expression in oscillatory systems, we also consider the problem of designing informative experiments for elucidating the dynamics and better identify the model. We demonstrate our approach on a benchmark scenario in plant biology, the circadian clock network of *Arabidopsis thaliana*, and discuss the different value of three types of commonly used experiments in terms of aiding the reconstruction of the network.

Finally we provide the architecture and design of a software implementation to plug in statistical methods of gene expression inference and network reconstruction into a biological data integration platform.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Daniel Trejo Baños*)

Acknowledgements

This research was funded by Microsoft Research through it's ph.D. scholarship program. I wish to deeply thank Microsoft and the University of Edinburgh for giving me this opportunity.

I am also very thankful to my supervisors Guido Sanguinetti and Andrew J. Millar, for their superb guidance and advice. I have learned so much from them and I will always look up to them as to how conduct myself as a scientist and as a leader.

I wish to thank my colleagues Botond Cseke, Vân Anh Huynh-thu, Ronald Begg, Gabriele Schweikert Andrew Zammit Mangion, Shahzad Asif and Daniel Seaton for all the useful discussions and help provided. I also want to mention all the other group members: Andrea Ocone, Tom Mayo, Alina Selega, Anastasias Georgulas, Dimitrios Milos, Andreas Capouranis and Yuanhua Huang. I feel honored of working among them in this wonderful institution. I wish to specially thank my dear friend David Schnoerr and my beloved girlfriend Milena Petkovic for their useful comments about this thesis.

To my parents Tomas Trejo Hernandez and Maria de Lourdes Baños Esquivel, and to my brother Alejandro Trejo Baños. To my angel Milena Petkovic. To my family and friends in Edinburgh and Mexico. Thank you for your love and support.

Contents

Chapter 1

Introduction

The field of computational biology currently is a Petri dish for the development of ideas and techniques in various disciplines. Even a quick overview of the top journals covering the subjects of biology, bioinformatics, systems biology and biostatistics will unveil a wide assortment of advancements and interdisciplinary exchange. Still, many of the challenges of the field remain unsolved given the inherent complexity of the subject (Nussinov, 2015).

In this thesis we tackle one of its oldest questions, how, given a set of measurements over gene expression, can we elucidate the intricate regulating interactions? The list of studies addressing this subject is extensive. Most of the current methods share a common framework, trying to infer the genetic regulatory network through steady state measurements, for example see Haury et al. (2012); Huynh-Thu et al. (2010); Margolin et al. (2006). Shorter is the list of methods taking into account expression profiles over a time series like in Bonneau et al. (2006); Huynh-Thu and Sanguinetti (2015). Here we will focus on biological systems with oscillatory behavior in particular. In these systems there is no steady state.

In biological systems, cycles are pervasive. From day/night to seasonal cycles, living beings developed oscillators to adapt to cyclical environmental conditions and control sequences of internal biological functions (Bell-Pedersen et al., 2005). One could try to adapt traditional methods to infer network structure from observations or just completely ignore the oscillating behavior. But our contention is that matching this prior knowledge about the regular oscillatory behavior will help us in our endeavor. By approximating the network dynamics through a linear model, we can operate fairly straightforwardly in the natural representation for cycles, the frequency domain. Of course, this approximation constrains the

type of systems to which we will be able to apply our method. However, we believe that this novel approach can potentially give us insights in these systems with oscillatory properties.

We embed our frequency domain approximation into a Bayesian statistical inference problem. Bayesian modeling will allow us to account for the inherent uncertainty related to the linear approximation and experimental noise. As an added bonus, we will be able to relate various sources of information by proposing a hierarchy of variables and models in which the network structure sits on top.

The resulting method, the DFT-based Spike and Slab prior (DSS) for network inference, showed good characteristics and higher accuracy than state of the art methods when the system under study presents regular oscillatory behavior. As this kind of behavior is not unique to circadian clocks, it could find some applicability in other fields. If the system fulfills the conditions of regular oscillations, observations over individual elements and scarce information about relationships between elements, then it is a candidate for being studied with this approach.

For example, an interesting application would be to study food webs with oscillations, these are ecological systems whose populations vary obeying cycles with approximately the same frequency. A potential application for these systems is to reconstruct complex ecological networks from species population measurements (Stone and He, 2007).

Some biological experiments are generally expensive and time consuming. We ideally wish to execute only those experiments with more promising outcomes for extracting information. The *scientific method* involves a model (hypothesis to be tested) and an experiment to test it.

State of the art work in experimental design for systems biology requires to simulate a mathematical model, along with some randomness added to the output or to the model parameters (Lindley, 1956; Kreutz and Timmer, 2009; Chu and Corey, 2012; Chaloner and Verdinelli, 1995; Liepe et al., 2013). This may require a lot of computational power thus an additional benefit of our method shall be to find a less expensive alternative.

We will make use of another property of our method to propose an angle of attack to the experimental design problem. Our network inference framework (and by extension model selection over the set of linearized dynamics) is formulated as a regression problem over a set of coefficients for some given basis functions. By employing the algebraic characteristics of the multivariate normal distribution,

we are able to directly sample from the solution of the linear system. We employ some of the tools of the Bayesian arsenal in order to design useful experiments.

Finally we develop a software implementation that can be incorporated into a bigger, biological data-integration server developed by Zielinski et al. (2014).

Thus the outline of this thesis follows the previously established narrative:

- *Chapter one* is divided in two parts. First we contextualize the biological concepts used in this work. The second part provides a quick overview of computational biology, with focus on statistics and machine learning.

- *Chapter two* presents the building blocks of the developed methods used in the rest of the thesis, from the linear approximation to the system dynamics, to the frequency-domain basis functions projection to the Bayesian inference framework.

- *Chapter three* is based on research done in the paper published in Trejo Banos et al. (2015a), where we will explain the *DFT-based Spike and Slab (DSS) method* for inferring network structure of biological oscillators. We use a frequency-domain approximation to the network dynamics with Bayesian multivariate linear regression for parameter inference.

- *Chapter four* accounts for the paper presented in Trejo Banos et al. (2015b). Here we adapt Bayesian *D-optimal design* for utility assessment and experimental design over the DSS framework.

- *Chapter five* draws conclusions about the advances made, limitations and potential improvements over the DSS framework, and its experimental design extension.

- *Appendix* presents the design architecture for developing specialized software that applies the developed inference method. In addition, the software includes another state of the art method for network inference, developed by Huynh-Thu and Sanguinetti (2015). This software is designed for integration to the Biodare repository (Zielinski et al., 2014).

1.1 Biological background

Living beings are composed lipids, carbohydrates, proteins and nucleic acids and other organic molecules. Most biological processes are ruled by biochemical reac-

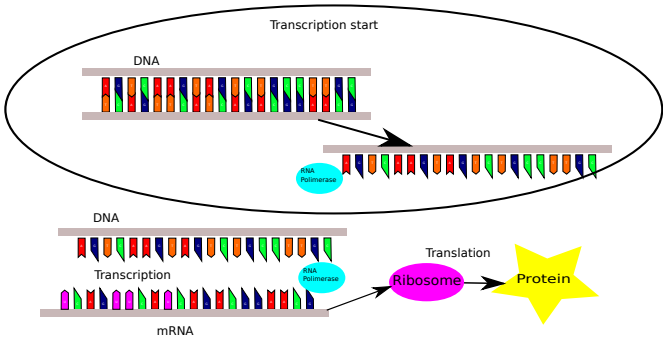


Figure 1.1: Central dogma of molecular biology. DNA is transcribed into mRNA and translated into proteins.

tions involving these components. For organisms to reach the levels of complexity that we observe in even the simplest living beings, information has to be passed from generation to generation.

This information is carried by *Polynucleotides,* which are molecules that guide the formation of exact copies of their own sequence. These sequences are based on paring of sub-units known as *nucleotides.* The DNA is a nucleic acid that is composed of four nucleotides *Adenine, Thymine, Guanine* and *Cytosine.* The *genetic code* is given by the combination of these nucleotides ordered in pairs (T-A) and (C-G) to form a double helix. In the organisms known as *prokaryotes* the DNA floats along other cell components within the cell membrane. In *eukaryotes*, the DNA is contained in the *nucleus.*

The *central dogma* of molecular biology explains the flow of information from DNA to the rest of the cell. This dogma states

> The coded genetic information is transcribed into messenger RNA (mRNA); each mRNA contains the program for synthesis of a protein through the process of translation

and it is illustrated in Fig.1.1.

A *gene* is a region of DNA that encodes one protein (or small group of proteins). An organism has the same genetic information in all its cells. And yet, there are many different kind of cells. This is because the type of cell and its functions will be determined by the level at which genes are expressed. Gene expression occurs through *transcription* and *translation (Alberts et al., 2002).*

Transcription *initiation* occurs when a molecule of *RNA polymerase* (RNAP) binds to a sequence of nucleotides, called *transcription start site.* Once bound, RNAP starts synthesizing RNA during the process of *elongation.* RNA is a single-

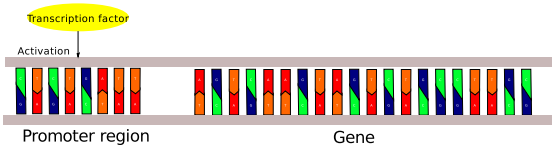


Figure 1.2: Representation of transcriptional regulation. The transcription factor (yellow) binds to the promoter region usually found upstream from the coding region of the gene (transcription start site).

stranded nucleic acid composed of the complementary base of the corresponding nucleotides in the DNA strand. RNAP assembles RNA during elongation, at the end of transcription (*termination)* a preliminary *messenger RNA (mRNA)* molecule is produced. This molecule contains *exons,* which are the coding regions of the gene, and non-coding *introns,* which are removed by a process known as *splicing.* The end result is a mRNA molecule ready to be bound and translated into protein (Alberts et al., 2002; Klipp, 2005).

A *transcription factor* is a protein that regulates gene expression by binding in a region upstream from the transcription start site of a gene; this region is called *promoter region.* The size of a promoter region can vary from hundreds, to thousands of base pairs counted from the transcription start site. When a transcription factor binds to this region, it promotes or inhibits the transcription rate of the corresponding gene. Fig.1.2 illustrates the process of gene regulation by the binding of transcription factors in the promoter region of a gene (Alberts et al., 2002; Klipp, 2005).

Once transcribed, the mRNA molecules are transported to the *cytosol*[1] where they bounded by the *ribosomes.* The latter are complexes of RNA and proteins where the mRNA nucleotides are coupled together with complementary bases, these complementary bases are grouped in short sequences. These short sequences, called *tRNA,* are bound to an aminoacid sequence; as these tRNA molecules bind to the mRNA the aminoacids assemble in chains. This chain-forming process is known as *protein synthesis.* This process of protein synthesis from mRNA is called *translation.* When the produced protein is a transcription factor or a component of one, we have genes intertwined in regulatory interactions. These genes form a so-called *genetic regulatory network* (GRN) (Alberts et al., 2002; Klipp, 2005).

[1]Interior of the cell not held by organels.

Some reactions important for gene regulation happen after translation. One of these kind of reactions is *phosphorylation.* It can be viewed as an "ON-OFF" switch for protein reactions. A special molecule called *phosphate* is catalyzed by proteins called *kinases.* The phosphate can then bind to specific aminoacids of the targeted protein chain [2]. This binding alters the shape of the protein and thus modifying its activity. This modification can be reversed by removal of the phosphate. This removal is performed by *protein phosphatase (Alberts et al., 2002).* Another reaction to take into account for the present thesis occurs when proteins are bound to each other by weak noncovalent interactions forming *protein complexes.*

1.2 Biological oscillators

Nature is ruled by cycles, starting from our planet that spins around it's axis and traces an orbit around the sun. Depending on the geography and location, conditions between day and night, and between seasons can change dramatically. There are even longer-run weather and geological cycles to which living organisms must adapt in order to survive.

By anticipating these variations, living beings are able to employ their resources more efficiently and prepare for adverse conditions. Many organisms possess an inner biological process that keeps track of these periodical changes over time. Additionally, many organisms require to execute biological functions in sequence; a clock-like oscillator is also needed for keeping track of these sequences. We will explain briefly two of the most common biological oscillatory process: *circadian clocks* are oscillators that rule the day-night response of organisms (Bell-Pedersen et al., 2005; Dodd et al., 2005; McClung, 2011) and the *cell cycle*, which governs growth and reproduction of cells (Stillman, 2013; Eser et al., 2014; Chen et al., 2004).

In Fig.1.3 we illustrate the circadian clock. This clock is present in most eukaryotes[3] and is in charge of regulating the response to day and night cycles. This is especially poignant for plants as they need sunlight for photosynthesis[4]. The main hypothesis of circadian studies is that circadian rhythms increase fitness and thus chances of survival, see for example Dodd et al. (2005). It is also

[2] specifically the aminoacids *serine*, *threonine*, or *tyrosine*

[3] Organisms whose cells have a nucleus containing the genetic information.

[4] Process by which plants transform solar energy into chemical energy

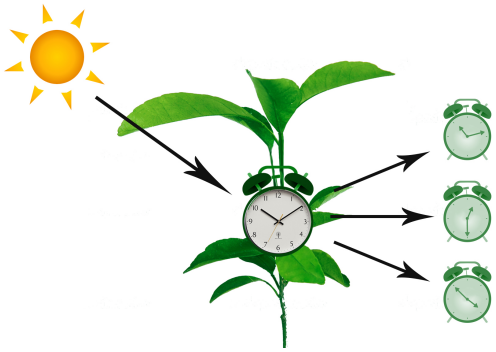


Figure 1.3: Illustration of the circadian clock. The sunlight is synchronizing signal for the clock. The clock itself is an internal biological process that is in charge of adjusting and coordinating many other independent biological processes and clocks.

known that malfunctions in the cellular time keeping mechanisms are frequently associated with diseases, further motivating the study of these systems (Bell-Pedersen et al., 2005).

Even the simplest plants, like *A. thaliana,* have an internal clock for day/night responses. Depending on the organism, between 10% to 100% of the genome may be regulated by circadian oscillations. These circadian regulated regions of the genome are in charge of important functions, for example flowering time (Shim and Imaizumi, 2015) and leaf growth (Dornbusch et al., 2014). As such it is evident that the understanding of this process is of vital importance for many fields related to biology, agriculture and medicine.

The main characteristics of a circadian clock are:

- A regular period of approximately (if not, exactly) 24 hours.
- Temperature compensation, that is, changes in temperature lead to changes in phase whereas the frequency remains the same.
- It is the result of a biological process inside the cell. It is self-sustained but it is entrained by rhythmic light and/or temperature signals.

When a GRN has feedback or feed-forward loops, oscillatory behavior can arise. In Fig.1.4 we present just one example of a genetic regulatory network with a loop. Transcription factors A and B are activated by extraneous inputs (environmental signals for example). These transcription factors bind to the promoter region of the target gene. This gene produces mRNA which is translated into a protein. This protein activates TF B and at the same time binds to

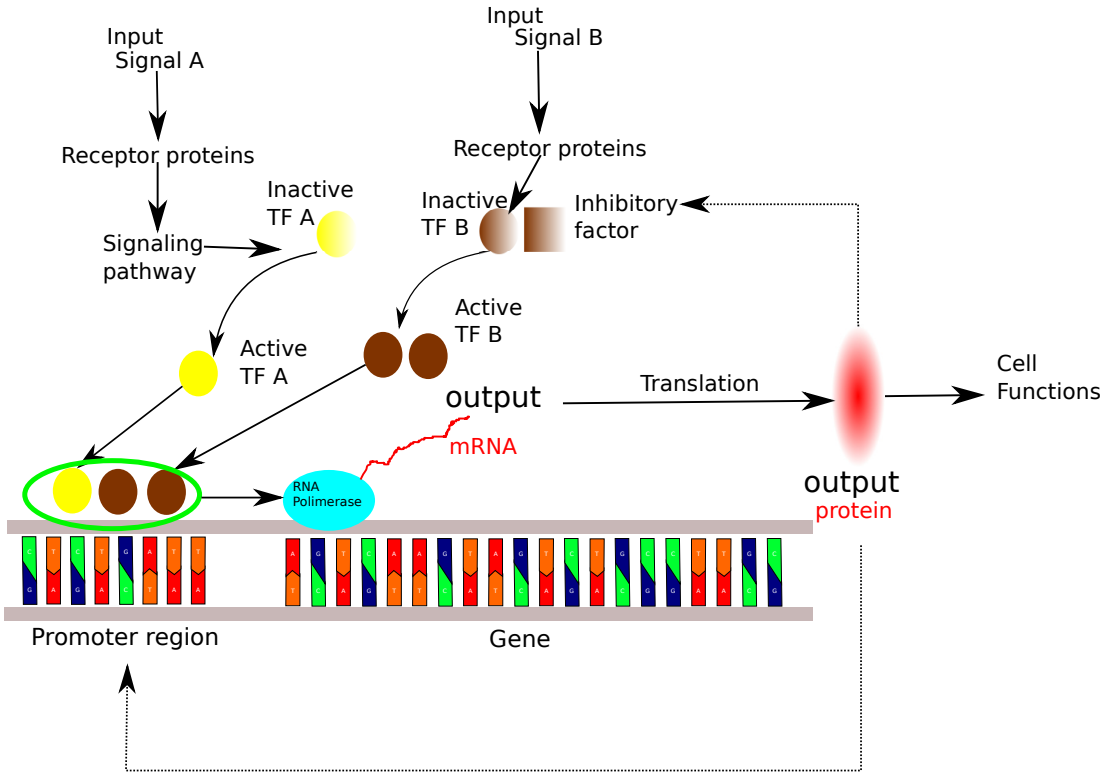


Figure 1.4: Example of genetic regulatory network with a loop. Transcription factors A and B are activated by extraneous inputs (environmental signals for example). These transcription factors bind to the promoter region of the target gene. This gene produces mRNA which is translated into a protein. This protein activates TF B and at the same time binds to its promoter region regulating (usually by inhibition) its own expression

Figure 1.5: Illustration of one of the clock mechanism. The genes interact through their promoter regions by producing transcription factors. These networks with feedback loops can be seen as the "gears" of the clock. The promoter regions are the connection points.

its promoter region regulating (usually by inhibition) its own expression. Many molecular genetic studies seem to indicate that regulatory feedback loops constitute the central mechanism of the clock. In Fig.1.5, we illustrate this idea, the "gears" of the clock are given by GRN with oscillatory behavior, the "teeth" or connections between components are the promoter regions of each gene.

In this thesis we will first focus on the circadian clock of *A. thaliana.* This organism has a relatively small genome of approximately 135 million base pairs. This clock has been widely studied but its exact mechanism remains to be deciphered. By observing the clock behavior three main transcription-translation feedback loops have been identified, though not fully characterized (McClung, 2011):

- A central loop consisting of Pseudo-Response Regulator (PRR), Timing Of Cab expression 1 (TOC1) and two transcription factors Clock Associated 1 (CCA1) and Late Elongated Hypocotyl (LHY). Experiments changing the concentration of these components result in the clock changing its phase, thus confirming them as clock components (Pokhilko et al., 2012; Vandepoele et al., 2009; McClung, 2011).

- A morning loop consisting of LHY and CCA1 regulating two PRR-type components, PRR7 and PRR9. In turn PRR7 and PRR9 along with PRR5 repress the expression of LHY and CCA1 thus forming a feedback loop (Pokhilko et al., 2012; Vandepoele et al., 2009; McClung, 2011).

- An evening loop that includes Gigantea (GI) and 'TOC1, along with LHY and the Evening Complex (EC) formed by LUX, Early Flowering 3 (ELF3) and Early Flowering 4 (ELF4) (Pokhilko et al., 2012).

Another example of oscillatory behavior in eukaryotes is the *cell cycl*e, which will also be briefly addressed in this thesis. By the cell cycle we mean the process by which a mother cell divides into daughter cells. Generally speaking, it involves a *synthesis phase* (S phase) and a *mitosis phase* (M phase). During the S phase the genome and other cell components replicate, and then divide during the M phase. The *growth phase* (G phase) is an intermediate phase between M and S, during which the cell grows until a certain signal is sent. This signal is given by the synthesis and destruction of certain types of proteins known as *cyclins.* The cyclins bind to *cyclin dependent kinases* (CDK) which initiates the biological process of the cell cycle phases. The CDK regulate proteins through phosphorylation.

Thus under regular environmental conditions, the cell can grow and reproduce at regular intervals. The *S. cerivisae* (yeast) cell cycle has been well studied and many components of the genetic regulatory network of the cycle have been identified (Eser et al., 2014).

1.3 Techniques and experimental design for measuring gene expression

In this section we clarify the distinction between the *experimental techniques* used to observe gene expression and gene regulation, and the *experimental design,* which aims at controlling experimental variables to study the biological system. These two procedures are not independent though. Usually the techniques used to measure gene expression greatly determine the experimental setting. For this, we will give a brief description of the most relevant measurement techniques and then an explanation of some common experimental designs for genetic regulatory studies.

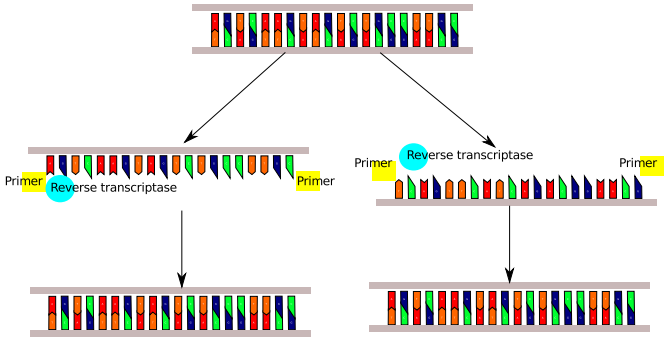


Figure 1.6: DNA amplification using PCR. The DNA strand is sheared and the primers (yellow) bind to the complementary bases. Reverse transcriptase (blue) starts adding a complementary base to the single strands. The final result is two copies of the original DNA strand.

1.3.1 DNA amplification and gene expression measurement using PCR

The measurement of components in a single cell is desirable, but it is still an emergent technology (de Souza, 2012). As such, we rely on older but proven technique for the analysis of DNA. One of such techniques, *polymerase chain reaction* (PCR), requires the generation of millions of identical DNA copies via a process called *amplification.*

The setting for this technique is as follows: suppose we need to replicate a region of length of 500 base pairs (bp) that is flanked by a 20 bp known sequence. We start by having available a copy of the region of interest, many artificially produced copies of two 20 bp length flanking fragments, and billions of nucleotides. The flanking fragments are called *primers,* and they are necessary for a *DNA polymerase* molecule to add a complementary copy to a single strand of DNA using the spare nucleotides (Klipp, 2005; Logan et al., 2009).

With the latter ingredients, PCR is illustrated in Fig.1.6 and comprises the following steps:

1. *Denaturation:* the DNA molecule is heated to split it into two single strands.

2. *Priming:* the solution cools down, the primers in the single strands anneal to complimentary sequences from the spare primers. The annealing is known as *hybridization.*

3. *Extension:* DNA polymerase produces double stranded DNA.

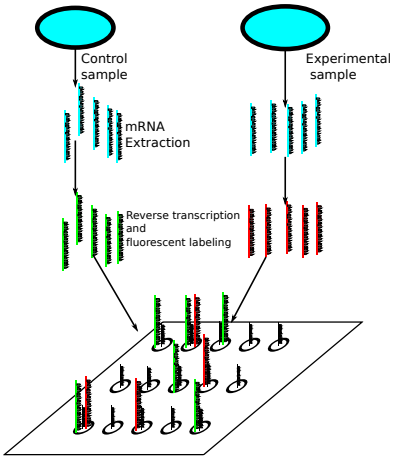


Figure 1.7: Microarray assay. In a two sample assay the mRNA from the control sample and experimental sample is extracted (blue). Then a flurescent liquid is applied and the dyed mRNA strands are hybridized with the probes laid in the array.

Each iteration produces two copies of the DNA sequence, thus allowing an exponential increase in the amount of DNA.

If we combine this technique with *reverse transcriptase* (produces a DNA copy from RNA) we can amplify mRNA. Using a technique known as *real-time PCR,* we apply a fluorescent dye to track the amount of product in each PCR cycle, thus having a quantitative assessment of the amount of mRNA as a measure of gene expression. Real time PCR provides a broad dynamic range (ability to detect samples with high and low copy number) but generally the amount of mRNA present is limited to one type; and thus limiting the capacity to detect expression changes in more than one gene (low throughput) (Klipp, 2005; Logan et al., 2009).

1.3.2 Gene expression analysis using microarray

A microarray experiment starts by building an *array* of spots of amplified single stranded-DNA into a glass slide or nylon membrane. Then mRNA is extracted from samples of the experiment being conducted. The mRNA is transcribed into a *complementary DNA* (cDNA) molecule using reverse transcriptase.

The cDNA molecules are labeled with fluorescent dye and incubated into the array. These single stranded cDNA molecules hybridize with its complementary DNA strands placed in the array. Thus the brightness of the spots in the array quantifies the amount of mRNA present as illustrated in Fig.1.7. This allows

us to perform high-throughput analysis of gene expression (Klipp, 2005). It is a costly technique and generally it is only possible to obtain few time-point samples for a single experiment.

1.3.3 Measuring gene expression using luciferase reporting assay

The luciferase reporting assay technique uses the biological process of *bioluminescence.* In this process *luciferin* is converted into *oxiluciferin* in a reaction catalyzed by the enzyme *luciferase* (produced by fireflies). This reaction is highly efficient and produces light.

We start by cloning the regulatory region under study upstream of a luciferase-producing gene. The resulting sequence containing both the regulatory region and the luciferase gene is inserted into a cell and allowed to grow. Then a low-light imaging is used to monitor the plant growth over many days. This light signal is another way of measuring gene expression (Fan and Wood, 2007; Van Leeuwen et al., 2000).

This technique is cheap and very sensitive to small changes in transcription and offers better signal-to-noise ratio than fluorescence-based methods. It's main drawbacks are that the dimmer light signal is more difficult to measure and it is a low throughput technique, which means that fewer samples are available for experiment.(Fan and Wood, 2007; Van Leeuwen et al., 2000).

1.3.4 Chromatin immunprecipitation (ChIP) for identifying binding sites

The ChIP technique aims at identifying regions of DNA associated with a certain protein by using a probe to link it (*cross-linking*) with the DNA-enveloping *chromatin.* This process is illustrated in Fig.1.8 and can be summed up in the following steps (Collas, 2009):

1. The proteins are cross-linked to the chromatin envelop of the DNA region of interest.

2. The DNA (along with its enveloping chromatin and proteins) is broken up.

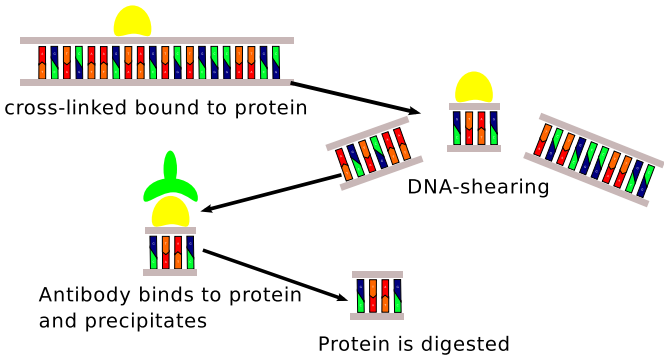


Figure 1.8: Chromatin immuno-precipitation steps. On top the proteins (yellow) are crosslinked to the chromatin envelop of the DNA. Then the DNA is broken up. An antibody (green) is added to the solution and precipitates the protein-DNA. Finally the protein is digested..

3. An antibody (protein that specifically binds to a protein) for the protein of interest is added to the solution. As a result of this process the antibody-bound proteins will precipitate. This is known as *immunoprecipitation.*

4. The DNA is cleaned from chromatin and protein.

Regions with more appearances will be those bound by the protein. If we apply this technique to a micro-array, it is known as ChIP-on-chip. Ideally it would allow to perform ChIP for multiple genes along the genome. It is severely limited by the amount of the genome that can be represented by an array and the signal-to-noise ratio (Gottardo, 2009). Additionally, depending on the organism, suitable antibodies may be unavailable. For example plant antibodies are specially difficult to come by.

1.3.5 Gene expression data

It is important to know the techniques used for measuring gene expression in order to properly handle the generated data. We will describe the data generated by Affymetrix microarray technologies. This technique is used in Orlando et al. (2008) for measuring gene expression in yeast, measurements which will be used in section 3.4.4.

Gene expression measurements as collected by Affymetrix micro array consist of image files of the microarray probes. These images are processed into cel files, which according to the manufacturer: “stores the results of the intensity

calculations on the pixel values of the DAT file. This includes an intensity value, standard deviation of the intensity, the number of pixels used to calculate the intensity value, a flag to indicate an outlier as calculated by the algorithm and a user defined flag indicating the feature should be excluded from future analysis" [5].

These cel files are then processed and normalized against the control sample using different software packages. Once this pre-processing is done we are left with gene expression readings, usually we transform them using the logarithm base 2 of the readings to identify fold changes in the data.

In Tu et al. (2002), the authors identify two sources of experimental noise for micro array experiments, which are:

- Sample preparation noise
- Hybridization noise

From empirical studies in Tu et al. (2002), the sample preparation noise " is dominated by an expression-independent constant and is in general much smaller than the hybridization noise". The authors also propose that "the genes labeled by the Affymetrix call as present, the dependence of the hybridization noise strength on the expression indicates a Poisson-like noise". Thus the signal to noise ratio[6] as a function of the number of (hybridization) events N is given by

$$SNR = \frac{\mathrm{N}}{\sqrt{\mathrm{N}}}$$

As N increases the noise distribution (Poisson) can be approximated by a Gaussian distribution (see 2.2.1 for characterization of this distribution).

Thus, we arrive to one of our modeling compromises, we assume a Gaussian approximation to the underlying noise distribution. This assumption can be applied to other techniques like bioluminiscence (which generally has a higher signal to noise ration than microarray) , or real time pcr which will possess different signal to noise ratio.

[5] Affymetrix web site `http://media.affymetrix.com/support/developer/powertools/changelog/gcos-agcc/cel.html` accessed on 10-01-2016

[6] The higher the signal to noise ratio, less noise is present in the signal.

1.3.6 Experimental design

For performing experiments in biology first we identify the components of the system under study. Then under controlled conditions, we perform measurements and try to deduce the functioning of the system from the observations. For biological systems we have to identify all the variables that interact with each other, from environmental signals to species in chemical reactions *(Kreutz and Timmer, 2009)*. By modifying these control variables of the experiment, we can observe the response of the system under different conditions and compare them with the response of organisms under "normal" circumstances.

For example, for studying the *A. thaliana* circadian clock, it is common to measure gene expression of clock related genes under varying day/night conditions. By observing how the components behave in these circumstances we can deduce the role that the light plays their gene expression.

We may also want to observe the behavior of the system by modifying one of its components *(Kreutz and Timmer, 2009)*. For this, mutant populations are created with desired characteristics. These mutant populations are then analyzed and compared with the unmodified populations. Some examples of these kind of experiments are

- Gene knockouts: one or more genes are prevented from expressing through various techniques. Depending on the amount of genes knocked out, they are called single mutant, double mutant and so on.

- Gene knockdown: instead of knocking out gene expression completely, it is inhibited.

- Gene overexpression, the counter part to gene knock-down, the gene expression of the target gene is increased.

We have now covered the basic biological concepts that are necessary to understand the problem of identifying the elements and relations of an oscillating genetic regulatory network. In order to make sense of the measurements made, we use a model of the system, in the next section we introduce the concepts of modeling in systems biology.

1.4 Modeling and computational biology

In *Systems biology* a wide assortment of mathematical tools are available in order to draw an abstract representation of a biological system. It typically relies on the integration of experimentation, data processing and modeling. It's characterized by making hypothesis about a system through a *model*. With this model we make predictions and we test them on experimental observations in order to iteratively refine or discard models.

Once we have a model we can test different scenarios. Generally there is a large number of scenarios that are needed to simulate and analyze. It is in these circumstances that the power of computational tools, to evaluate multiple conditions becomes essential. Still, it is of vital importance to recognize the limitations of modeling in general and of the chosen modeling framework in particular.

In Kitano (2002), the author identified four fundamental properties that a systems biology model should aim to study:

- *System structures:* identify which components are interacting and how they are connected.
- *System dynamics*: try to observe and predict the behavior of the system over time.
- *Control methods:* find which mechanism can be used to control the malfunction of a cellular process.
- *Design method:* how to build biological systems that present a desired behavior.

Ideally we would be able to identify all the components of a biological network, then draw hypothesis about the mechanistic process underlying their interactions and then proceed to apply control and design methods. More often than not, the inherent complexity of the system requires simultaneous analysis of these properties (Kitano, 2002).

This problem is compounded if we consider that there are many ways of describing a biological process. The same process can be investigated using different experimental techniques and different models. As stated in Klipp (2005), "modeling has to reflect the essential properties of the system". This is not an easy task as we have to first identify which are these properties.

Even if we possessed a reasonable mathematical model based on our current knowledge, we need to make it coherent with the observed behavior of the system. Here is where the *identifiability* problem arises: how it is possible to uniquely parametrize a model in order to explain the experimental measurements made? (Kreutz and Timmer, 2009)

Most biological processes are robust i.e., they will endure perturbations to some of their components without significantly changing their behavior (such as the previously discussed circadian rhythms). The question arises, how do we include this robustness into our model? Robustness increases the difficulty of identification, so that even for the most simple systems, a huge number of possible models and parametrizations may exist. In this case the amount of data needed to correctly identify a system can be very large in comparison to the amount of data available.

Optimally we would like to use all of the available sources of information in order to solve the identification problem. It is thus important to reconcile the data obtained through varied experimental techniques. The term *data integration,* in a biological context, refers to the use of different sources of information for the study of a system. The development of data integration techniques and methodologies has become one of the main focus of bioinformatics and system biology in recent years (Gomez-Cabrero et al., 2014).

An important characteristic in choosing a modeling framework is the scope of the model. Currently there is a trade off between detail and size biological models can offer. Very coarse models such as petri nets and boolean network can reproduce certain characteristics of dynamical systems such as steady states and oscillations, but their coarseness limit their biological feasibility (Karlebach and Shamir, 2008). On the other hand, the same processes can be modeled through more fine grained mechanistic models, but their complexity and high number of parameters may render them impractical to systems of more than a few elements.

In this thesis we develop a mathematical model of gene expression at the level of transcription that fits somewhere in between the coarseness of boolean (or logical) networks and the detail of mechanistic models, allowing for systems of medium size (the method has been tested with systems up to 20 components). A wide array of mathematical modeling techniques has been applied to this problem before. We used a linear approximation to the network dynamics. This kind of approximation is possible for any non-linear system and offers a trade-off between

identifiability and biological plausibility. Given the complex nature of the problem we hoped our method, while inducing some biases in the parametrisation, may provide reliable answers for structure learning. We proceed to overview some of the most relevant for our work. Then we will introduce our model approach of reformulating the problem in the frequency domain; this in order to apply statistical modeling methods for structure and parameter identification.

1.4.1 *Ordinary Differential Equations* (ODE)

In ODE models, the rate of change of the concentrations of mRNA or proteins is described. They are represented by functions of transcription, translation or other individual processes. These functions are usually non-linear and are called *rate equations.*

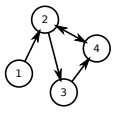
These models assume the elements are "well mixed", this means that species are present in large abundance thus space is homogeneous and there is no space dependency. Even though concentrations are a positive integer, they are approximated as real numbers. The final assumption is that interactions have instantaneous effects. Under these assumptions we can formulate a deterministic description of the system in continuous time (Lawrence et al., 2009).

The main issue of these models is choosing or determining the rate functions. These functions relate the concentrations of the other process components through sets of parameters known as *kinetic constants.* Herein lies its main difficulty, as it is generally not possible to measure these constants and we lack of knowledge about most of the molecules involved and their interaction (Klipp, 2005; Lawrence et al., 2009).

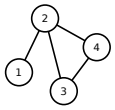
1.4.2 Graph representations

A graph is a mathematical object and it is represented by a tuple $\langle V, E \rangle$. In our biological context the vertices V represent the genes and the edges E represent interactions from gene j to gene i or viceversa, as a tuple (i, j). When (i, j) equals (j, i) the graph is *undirected*. Another way to represent a graph is through its *adjacency matrix,* in Fig.1.9 we show a graph with four vertices and four edges. On top we have a directed graph with its corresponding adjacency matrix, an undirected graph is shown at the bottom.

In an adjacency matrix, the vertices are sorted in rows and columns, and each



$$\begin{matrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{matrix}$$



$$\begin{matrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{matrix}$$

Figure 1.9: Graph representation of a four node network. Vertices are numbered from one to four. The upper and lower panel show a directed and undirected graph respectively, with the corresponding adjacency matrices.

element of the matrix will represent an edge. If the value of an element equals 1 the edge exists and 0 if not.

Even though a graph representation may not provide information about the dynamics of the network, it can be used to analyze certain properties such as feedback loops, redundancies and network complexity (Pavlopoulos et al., 2011).

1.4.3 Stochastic methods

The assumptions made for ODE modeling do not take into account that the biochemical process of gene expression is neither continuous nor deterministic. Because the particles are discrete and diffuse in the cytosol the chemical reactions are random events, and thus inherently stochastic.

The probability of a well mixed and diluted system[7] being in a certain state at given time is represented by a *chemical master equation.* Generally there are no analytical solutions available for this equation. However the system behavior can be simulated through a *stochastic simulation algorithm* (Gillespie, 1977).

As with ODE, one big obstacle to overcome is its reliance on kinetic parameters that are generally unknown. Additionally it requires a bigger computational overhead than deterministic simulations. To overcome this problem *Stochastic differential equations (SDE)* are frequently used as a continuous approximation. They describe systems with a low to intermediate abundance of species[8], while deterministic simulations on the contrary case. For example in Ocone et al. (2013)

[7]This means that we can obviate the spatial components of the system as all chemical particles are accessible.

[8]For low abundances stochastic simulation may be feasible

a SDE-based model is used to model regulatory network dynamics.

1.5 Statistical inference

In addition to the previously discussed intrinsic stochasticity of the chemical reaction, additional sources of uncertainty (measurement error, unobservable quantities) are present (Klipp, 2005). As such, the field requires us to make sense of data taking these uncertainties into account. For this purpose we use probabilities. Probabilities are "the mathematical language for quantifying uncertainty" (Wasserman, 2013). Here we will introduce the essential concepts for this thesis.

Having the space of possible experimental outcomes, each outcome is called *realization.* A set of realizations is called an *event.* A *random variable* $x \in \mathbb{R}$ is a variable whose value is subject to an experimental outcome[9]. If x is discrete, the *probability* of x taking certain value is denoted by $P(x)$. The *probability mass function* is a function that describes the probability of discrete variable x taking certain value. If the random variable x is continuous then we define the *probability density function,* which has the following properties

$$\begin{aligned} p(a < x < b) &= \int_a^b p(x)\,dx, \\ \int_{-\infty}^{\infty} p(x)\,dx &= 1, \\ p(x) &\geq 0. \end{aligned}$$

In this thesis we will use the term probability distribution to refer to probability functions over either discrete or continuous random variables and will be denoted as $p(x)$.

Statistical inference, as succinctly stated by (Wasserman, 2013) is "the process of using data to infer the distribution that generated it". There are two opposing views about what a probability conveys and consequently, about how to perform inference. On one hand, among the main contentions of the *frequentist* point of view, is the assumption that probabilities are objective properties of the real world. In frequentist statistics model parameters are fixed, but potentially unknown.

[9]This can be extended to non-numeric domains, and are called *random elements.* Nevertheless, the term random variable is used indistinctly.

On the other hand, for Bayesian inference probabilities are assumed to describe degrees of belief. By adopting this point of view we can produce a probability distribution over the models parameters given the available data.

Bayesian inference assigns a *prior distribution* to a parameter vector θ. This distribution $p(\theta)$ amounts to our prior knowledge about these parameters. The distribution of the data given a model parametrized by θ is known as *likelihood.* The prior is combined with the likelihood by applying Bayes theorem and as a result we obtain the distribution of the parameters given the observations, known as *posterior distribution.* A more detailed explanation of Bayesian inference will be given in Section 2.2.

The literature discussing the characteristics, advantages and disadvantages of Bayesian methods is extensive, for example see (Wasserman, 2013; MacKay, 2003; Bishop, 2001; Barber, 2012), just to cite a few. Among its main disadvantages are the difficulties and computational overhead when dealing with high dimensional problems. Nevertheless, in this thesis we will adopt a Bayesian inference framework as it offers a principled way to combine prior beliefs about the model with the observed data (Wasserman, 2013).

1.6 Discussion

In this chapter we briefly reviewed the biological and statistical background for this thesis. Due to the great extent of literature on systems biology and statistical inference, we are not able to cover in detail all the mentioned concepts. However, we hopefully provided enough information to contextualize our research and the significant difficulties that we face understanding and modeling biological systems.

It is in this context of system biology that we will develop a model for approximating the complex regulatory interactions between regulatory network elements. Then we will apply a statistical framework in order to infer the model parameters from experimental data. We will then demonstrate the accuracy and efficiency of our method by testing it in simulated and real data. Finally we will compare our method with other state of the art network inference methods.

As mentioned, a model provides a way of simulating experiments; by working in a probabilistic setting we can use inference to help us design the experiments most useful for identifying our system. Thus the objectives of this thesis can be

summed up as deriving a statistical inference method that:

- Can learn the structure of oscillating genetic regulatory network and that accounts for network dynamics.
- Can integrate multiple sources of information.
- Accounts for experimental and modeling uncertainties.
- Leverages prior information about oscillatory systems.
- Help us choose the most useful future experiments based on the current information about the system.

Our main modeling assumptions can be summed up as:

- The system behavior can be approximated by a system of linear ordinary differential equations (see 2.1).
- The noise distribution of the gene expression measurements can be approximated using a normal distribution (see 3.2).

Thus the performance of the method will greatly depend on the characteristics of the system and the noise present in the experimental measurements.

Chapter 2

Methods

Biological processes are physical systems with many components interacting through chemical reactions. Mathematical modeling offers general purpose tools to hypothesize, experiment and draw conclusions from biological experiments. Having a model about a given biological system usually is not enough. We are only able to measure some of the systems components in an experimental setting and we need to match the observations with the mathematical abstractions used. This process of *identification* requires us to estimate the parameters for a model given the observed experimental outcomes.

In this chapter, a *linear time invariant* (LTI) model for gene expression is developed. Next we review a main tool used in this thesis, the *Discrete Fourier Transform* (DFT) which allows us to convert differentiation and integration in the time domain into a set of matrix algebra equations in the frequency domain (Pintelon and Schoukens, 2012).

By transforming the time domain data into the frequency domain, we are able to translate the network inference problem into a regression problem. We first introduce the DFT basis, its advantages and pitfalls for representing a sampled continuous signal in discrete frequency domain. We then explore some useful properties of this transformation and its relation with linear time invariant systems. Finally, we give a brief introduction to Bayesian inference as a way of parameter estimation. These techniques will come together in the next chapter as we develop a Bayesian hierarchical model for structure learning and parameter estimation in a frequency-domain LTI model of genetic regulatory networks.

2.1 Gene expression as a Linear Time Invariant system

The mathematical framework of *Ordinary Differential Equations* (ODE) aims at modeling the instantaneous change of the network components' as a function of the concentrations of the other components. These models have been used extensively and provide a detailed description of the network dynamics (Dalchau, 2012); usually through non-linear functions of the components' concentrations. Its main drawback resides in the precise knowledge that is needed about the network components and the kinetic parameters of the reactions (Karlebach and Shamir, 2008).

In ODE modeling the *state* is a representation of the system at a given time. All possible states of a system are contained in the *state space* and the dimension is equal to the number of variables of the system. A solution to a system of ODE given its parameters and initial conditions is represented by a *trajectory* in state space. A *steady state* is a point in state space where the solution to the system of ODEs is constant in time. An equilibrium point is a point at which the system will remain at if started there. An ODE representation of the concentration of the x_i components has the following form

$$\begin{aligned} \frac{d}{dt}x_1 &= f_1(x_1,..x_n) \\ &\vdots \quad\quad \vdots \\ \frac{d}{dt}x_n &= f_n(x_1,..x_n) \end{aligned} \tag{2.1}$$

where $\frac{d}{dt}$ is the derivative w.r.t to time and $f_1 \dots f_n$ are some nonlinear functions of the other components concentrations.

The framework of *linearization*, aims at obtaining qualitative insights into regulatory networks. In these models, we approximate the system behavior around a *equilibrium point* through linear functions. We are going to approximate a systems response as a function of its deviation from a equilibrium point $\tilde{x_i}$. We denote $\delta_i(t)$ as this deviation from the equilibrium, such that

$$x_i(t) = \tilde{x_i} + \delta_i(t).$$

The ODE system given by Eq.(2.1) as a function of $\delta_i(t)$ is

$$\frac{d}{dt}\delta_i = f_i(\tilde{x_1} + \delta_i, .., \tilde{x_n} + \delta_i).$$

Now let's write down everything in vector notation, having $\mathbf{x}(t) = [x_1(t), ..x_n(t)]^T$ be the vector of n gene expression levels, the corresponding n-dimensional equilibrium point $\tilde{\mathbf{x}} = [\tilde{x_1}, .., \tilde{x_n}]^T$ and the vector of deviation from equilibrium $\Delta(t)$; such that $\mathbf{x}(t) = \tilde{\mathbf{x}} + \Delta(t)$, thus the vector $\mathbf{f}(\mathbf{x}) = [f_1(x_1, ..x_n) \dots f_n(x_1, ..x_n)]^T$ denotes the nonlinear functions.

We wish to approximate $\frac{d}{dt}\Delta$ by a Taylor expansion up to the second term around $\tilde{\mathbf{x}} = \mathbf{0}$, which yields

$$\frac{d}{dt}\Delta = \mathbf{f}(\Delta) \approx \mathbf{f}(\mathbf{0}) + \Delta \mathbf{J}_{\tilde{\mathbf{x}}}(\mathbf{f})\,\Delta + \mathcal{O}\left(\Delta^2\right)$$

Where $\mathbf{f}(\mathbf{0}) = \mathbf{0}$ and $\mathbf{J}_{\tilde{\mathbf{x}}}(\mathbf{f})$ is called Jacobian matrix and is given by

$$\mathbf{J}_{\tilde{\mathbf{x}}} = \{a_{ij}\} = \begin{bmatrix} \frac{\partial}{x_1} f_1\Big|_{\tilde{x_i}} & \cdots & \frac{\partial}{x_1} f_n\Big|_{\tilde{x_i}} \\ \vdots & \ddots & \vdots \\ \frac{\partial}{x_n} f_n\Big|_{\tilde{x_i}} & \cdots & \frac{\partial}{x_n} f_n\Big|_{\tilde{x_i}} \end{bmatrix}$$

the term $\mathcal{O}\left(\Delta^2\right)$ is a function of a third order tensor [1] called Hessian[2], thus we have a set of n second derivative matrices, each one of dimension $n \times n$. Given the computational complexity that involves including the second term in our method, we compromise to the first order linearization

$$\frac{d}{dt}\mathbf{x} = \mathbf{J}_{\tilde{\mathbf{x}}}\mathbf{x} \tag{2.2}$$

The linearized system of eq.2.2 predicts the local behavior of the nonlinear system around an equilibrium point if the equilibrium point is hyperbolic,that is, if the Jacobian has no imaginary eigenvalues. The Hartman-Grobman theorem states that, if the equilibrium point is hyperbolic, then there exists a continuous map with a continuous inverse that transforms every trajectory of the nonlinear system to a trajectory of the linearized system. If the equilibrium point is non-hyperbolic the behavior of the system around this equilibrium then would be very dependent on the higher-order terms in the Taylor's series expansion . Even though in nature systems are non linear and the Hartman-Grobman conditions usually are not met, this is a widely used approach in systems biology, that allows us to deal with systems of different sizes while retaining some qualitative

[1] Array of matrices.

[2] Important to notice that $\mathbf{f}$ is a vector field; thus it's Hessian is a third order tensor, an array of matrices, each one being a matrix of second order partial derivatives for the scalar functions $f_i(x_1, ..x_n)$.

properties of the system (Polynikis et al., 2009; De Jong et al., 2004; Bonneau et al., 2006; Dalchau, 2012; Morrissey et al., 2011).

Internal oscillations in a system occur when the solutions are closed curves around a steady state. For many biological oscillators such as circadian rhythms, these oscillations have regular periods (24 hour periods for circadian oscillations for example). Our main hypothesis is that we can model these internal oscillations through a "driven" linear system, that is, a system whose oscillations are driven by a extraneous force (light in case of circadian rhythms and cyclins in case of cell cycle). This approximation takes the form of a linear system of ODE with constant parameters, which is known as *Linear Time Invariant Systems* (LTI) with periodic input. A solution to an LTI with periodic input will yield a set of output signals with the same frequency as the given input (Callier and Desoer, 2012). These solutions are also known as steady states in LTI literature, we will refrain to use the term in order to avoid confusion with it's common definition.

In the setting of a LTI system, the resulting model does not offer the details that complex models based on non-linear ODE or *Stochastic Differential Equations* (SDE) can yield. On the other hand, LTIs are considerably easier to evaluate and parametrize while often providing a reasonable approximation to the observed behavior. By choosing an LTI model representation we sacrifice accuracy and physiological interpretability, but this is compensated by the quantity of models and parameters that can be evaluated, especially when the interactions between components are unknown, additionally it matches our prior observations of regular sustained oscillations with (almost) the same frequency as the environmental signals driving these processes (Dalchau, 2012; Karlebach and Shamir, 2008).

2.1.1 ODE model

For a set of N genes in a gene-expression measuring experiment, q will denote the experiment number and the variable $x_i^q(t)$ is the expression level i.e. the mRNA concentrations of gene $1 < i < N$ at time t in experiment q. We will express the rate of change of $x_i^q(t)$ as a function of the molecular level of the *regulatory genes* x_j^q such that gene j regulates gene i, a *basal expression rate* b_i^q, and the gene's mRNA *decay rate* λ_i.

A main assumption is that we can approximate the effect of the unobserved

regulators by the observed gene expression levels. We define gene regulation by including the *interaction parameters* $\alpha_{ij} \in \mathbb{R}$ which will be positive in case of an activating interaction and negative in case of repression. By allowing α_{ij} to be zero, we can expand the set of putative regulatory genes to cover all genes j such that $j \neq i$.

For completeness we add a set of L *external inputs* u_l^q such that $1 < l < L$ and their corresponding set of interaction coefficients $c_{il} \in \mathbb{R}$. The inputs can represent either environmental signals, gene expression levels from elements outside the network, or known Transcription Factor concentrations. With all these considerations the ODE model reads

$$\frac{d}{dt}x_i^q(t) = \sum_{j \neq i}^{N} \alpha_{ij}x_j^q(t) - \lambda_i x_i^q(t) + b_i^q + \sum_{l=1}^{L} c_{il}u_l^q(t)\,,\ i = 1,..,N. \tag{2.3}$$

Eq.(2.3) is a continuous LTI. The properties of linearity and time in-variance are evident from the fact that the parameters $\{\alpha, \lambda, b, c\}$ are constants. Additionally, it is well known that for any oscillatory function or signal, the output of the system is another oscillatory signal scaled in phase and amplitude, but the frequency remains constant (Pintelon and Schoukens, 2012).

Given the LTI model, the problem of network inference can be approached by noticing that the interaction parameters α correspond to the weights assigned to edges in a gene regulatory network. For recovering the network structure we need to compute the interaction parameters α and c. Given these interaction parameters we aim to recover the embedded network topology.

2.1.2 Discrete Fourier Transform

The Discrete Fourier Transform (DFT) is the projection of a discrete signal, denoted as $x[m]$, of length M into a discrete set of coefficients. In our case the discrete signal corresponds to measurements over the gene expression levels at equally spaced time points. The DFT coefficients are computed for a set of sinusoidal basis functions and are defined as

$$X[k] = \sum_{m=0}^{M-1} x[m]e^{-i2\pi mk/M} \tag{2.4}$$

with the basis functions $e^{-i2\pi/M}$ known as M-th *root of unity.* From now on we will denote X the DFT of the sampled signal.

From a mathematical point of view, the DFT is a discrete approximation to the continuous Fourier spectrum of a signal. Let us assume that:

1. The signal's bandwidth (range of frequencies) is less than half the sampling frequency. This amounts to having more than one sample per oscillation period. This is crucial in order to resolve the oscillation frequency

2. The signal is sampled over an integer number of periods

then the DFT is an exact reconstruction of the signal's spectrum (Pintelon and Schoukens, 2012). Any sampling scheme that intends to analyze the frequency spectrum of a signal should aim to fulfill those two conditions (as it is often the case in experiments studying biological oscillators).

From a computational point of view, the DFT's main advantage is that it can be obtained efficiently by a family of algorithms called *Fast Fourier Transform*, see (Cooley and Tukey, 1965). With these algorithms it is possible to obtain the DFT in $\mathcal{O}(M \log_2 M)$ instead of the $\mathcal{O}\left(M^2\right)$ operations that the explicit calculation requires (Cooley and Tukey, 1965).

The DFT computed by the FFT will be our main tool for transformating into the frequency domain. From the DFT definition in (2.4) some useful properties are observed

- *Linearity* is the most relevant property for the present work. Having a sum of two discrete signals x and y, the DFT obeys the equality

$$DFT(ax + by) = aX + bY$$

  with constants a, b, and $X = DFT(x)$, $Y = DFT(y)$.

- The *time derivative* of a signal amounts to a multiplication in frequency domain

$$\frac{d}{dt}x(t) = \frac{i2\pi}{T}kX[k]$$

  this property holds in those uniformly sampled point from which the DFT of x is computed. The derivation of this property is based on a trigonometric interpolation. We can express the continuous time signal $x(t)$ as a function of the IDFT of X by simply substituting $m = (M/T)t$, where T is the

signal's period and the term M/T is known as the *sampling rate*. Thus formulating a interpolation for $x(t)$ at M uniformly spaced points using the IDFT, we have

$$x(t) = \sum_{k=0}^{M-1} e^{i2\pi kt/T} X[k]. \tag{2.5}$$

- The final DFT-related property we introduce is the reconstruction of a discrete signal $x[m]$ from a set of DFT coefficients through the Inverse Discrete Fourier Transform, IDFT, which is defined as:

$$x[m] = \sum_{k=0}^{M-1} e^{i2\pi mk/M} X[k]. \tag{2.6}$$

2.1.2.1 Real representation of the complex DFT coefficients

For data manipulation it is convenient to represent the DFT coefficients as a vector of real valued elements. If the signal $x(t)$ is real, only the first $(M-1)/2$ coefficients of the DFT given by the FFT will be non redundant as $X[k] = \bar{X}[k]$ with $\bar{X}$ being the complex conjugate of X and for $k > M/2$.

Let $X_{M/2}$ denote the first $(M-1)/2$ coefficients of X, we define the *real composite* vector $X_R \in \mathbb{R}^{M-1}$, by stacking the real and imaginary parts of $X_{M/2}$. This representation is called the *Real Discrete Fourier Transform* (RDFT)*(Ersoy, 1985)* and in vector notation reads

$$X^R = \begin{bmatrix} \Re X_{M/2} \\ \Im X_{M/2} \end{bmatrix}.$$

2.1.3 LTI formulation in the frequency domain

The time derivative in the DFT is straightforward to compute, by differentiating Eq.(2.5) w.r.t. t and evaluating over sample points $t = mL/M$

$$\left.\frac{d}{dt}x(t)\right|_{t=mL/M} = \sum_{k=0}^{M-1} \frac{i2\pi}{T} kX[k] e^{i2\pi kt/T}.$$

Thus the DFT coefficients of the time domain derivative, are related to the DFT coefficients X by a factor of $\frac{i2\pi}{T}k$. The most important conclusion is that, in frequency domain, time differentiation is a *multiplicative operator*.

The time derivative can be represented in matrix form by the block matrix $\mathbf{D}$, whose upper-right and bottom-left blocks correspond to diagonal matrices with

the frequency components as elements

$$\mathbf{D} = \left[\begin{array}{c|c} \mathbf{0} & \boldsymbol{\omega} \\ \hline -\boldsymbol{\omega} & \mathbf{0} \end{array}\right]$$

where $\boldsymbol{\omega}$ is the diagonal matrix with elements $\frac{2\pi}{T}k$. Using this matrix, time differentiation of the spectra is defined as the product $\mathbf{D}X^R$.

Going back to the LTI Eq.(2.3), let's define the matrix $\mathbf{X}^q$ as the matrix whose columns represent the RDFT of the expression level samples of a set of N genes for an experiment q. That is, $\mathbf{X}^q = \left[X_i^R\right]_i$ with $1 < i < N$ and $X_i^R = RDFT\left(x_i[M]\right)$. Analogously, $\mathbf{U}^q$ will represent the RDFT of the system inputs. We denote by $\dot{\mathbf{X}}^q$ the time derivative of the spectra, which can be computed by the matrix product $\mathbf{DX}$. Then the matrix form for the LTI system is given by

$$\dot{\mathbf{X}}^q = \mathbf{X}^q\mathbf{A}^T + \mathbf{U}^q\mathbf{C}^T. \tag{2.7}$$

With Eq.(2.7) we have turned the system of differential equations into a matrix equation. The interaction coefficients and decay rates are grouped in the matrix $\mathbf{A}$, with off diagonal entries given by the α parameters and the λ parameters located along the main diagonal. Similarly, the input coefficients are grouped in matrix $\mathbf{C}$. The basal expression rates are located in the first entry of matrix $\mathbf{X}^q$ and correspond to the signal's mean value.

Now the challenge will be to estimate parameters $\mathbf{A}, \mathbf{C}$ given the gene expression and input signals. We pursue a Bayesian approach to parameter estimation in order to fulfill our modeling objectives from section 1.6.

2.2 Bayesian statistics

The problem of parameter estimation given a set of observations remains an important problem for modeling a physical process. By estimating the parameters of our model we can make predictions and validate the model assumptions.

The observed data usually comes from experiments with uncertainties from the measuring instruments, experimental conditions and intrinsic fluctuations. A direct fitting of the model parameters by the solution of a system of equations would not account for these uncertainties; therefore statistical methods have been developed to deal with these issues. In these models we make use of *probability distributions* to model assumptions over the data.

2.2.1 Exponential family distributions

Among the most commonly used distributions in statistical modeling we find the *exponential family*. This family of distributions include the normal, exponential, gamma, chi-squared, beta, Dirichlet, Bernoulli, categorical, Poisson, Wishart, Inverse Wishart distribution and others. Relevant distributions for this thesis, are

- *Bernoulli distribution*, it is a discrete distribution with a single parameter $0 < \alpha < 1$, $\alpha \in \mathbb{R}$. The distribution $p(v|\alpha)$ of the random variable $v \in \{0, 1\}$ conditioned on this α is given by

$$\text{Bernoulli}(\alpha) = \alpha^v (1-\alpha)^{1-v}.$$

- *Beta distribution*, determined by a pair of shape parameters $\alpha > 0$, $\alpha \in \mathbb{R}$ and $\beta > 0$, $\beta \in \mathbb{R}$. The random variable $v \in [0, 1]$ will have the distribution $p(v|\alpha, \beta)$ given by

$$\text{Beta}(\alpha, \beta) = \frac{v^{\alpha-1}(1-v)^{\beta-1}}{\text{B}(\alpha, \beta)}$$

  where $\text{B}(\alpha, \beta)$ is the *Beta function.*

- *Normal distribution*, parametrized by it's mean $\mu \in \mathbb{R}$ and variance $\sigma^2 > 0$. In this case the distribution $p\left(v|\mu, \sigma^2\right)$ of $v \in (-\infty, \infty)$ is

$$\mathcal{N}\left(\mu, \sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(v-\mu)^2}{2\sigma^2}}.$$

  Additionally, we can model a vector of N normally distributed random variables $\mathbf{v} \in \mathbb{R}^N$, by the *Multivariate Normal Distribution* with mean $\mathbf{m}$ and variance-covariance matrix Σ with pdf

$$\mathcal{N}(\mathbf{m}, \Sigma) = \frac{1}{\sqrt{2\pi^N \det(\Sigma)}} e^{-\frac{1}{2}(\mathbf{v}-\mathbf{m})^T \Sigma^{-1} (\mathbf{v}-\mathbf{m})}.$$

- *Gamma distribution*, parametrized by its scale $\alpha > 0$ and rate $\beta > 0$. Here the distribution $p(v|\alpha, \beta)$ of $v \in (0, \infty)$ is

$$\text{Gamma}(\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} v^{\alpha-1} e^{-\beta v}$$

  where $\Gamma(\alpha)$ is the *Gamma function.*

Likelihood	Conjugate prior
Bernoulli with $\theta = \alpha$	Beta (α_0, β_0)
Normal with $\theta = \mu$	$\mathcal{N}(\mu_0, \sigma_0)$
Normal with $\theta = \sigma^{-2}$	Gamma (α_0, β_0)
Multivariate normal with $\theta = \mathbf{m}$	$\mathcal{N}(\mathbf{m}_0, \Sigma_0)$
Multivariate normal with $\theta = \Sigma = \sigma^{-2}\mathbf{I}$	Gamma (α_0, β_0)

Table 2.1: Some common conjugate priors $p(\theta)$ for likelihood functions $p(v|\theta)$.

2.2.2 Bayesian inference

The main objective of the present thesis is to set the solution of the matrix Eq.(2.7) in a probabilistic framework, in order to account for both experimental and approximation-related uncertainties. To this end, Bayesian statistics are appropriate, as it quantifies the uncertainties over a set of parameters given some observations (Bishop, 2001).

In Bayesian statistics, the assumptions over a set of parameters θ are encoded in the *prior* probability distribution $p(\theta)$. The probability of the observations v given the parameters θ, which is commonly referred as *likelihood* or *likelihood function*, is denoted by $p(v|\theta)$. Then we can compute the conditional distribution $p(\theta|v)$, known as *posterior*, by applying Bayes' theorem

$$p(\theta|v) = \frac{p(v|\theta)\, p(\theta)}{p(v)} \tag{2.8}$$

from which $p(v)$ can also be computed by the prior and the likelihood by

$$p(v) = \int p(v|\theta)\, p(\theta)\, d\theta. \tag{2.9}$$

The integral in (2.9) is generally intractable. However there are special cases where the posterior can be computed analytically and in a fairly simple manner through *conjugacy*. When a prior and its corresponding posterior have the same form, we speak of a *conjugate prior* of the likelihood function, and the posterior can be straightforwardly computed in closed form. The resulting posterior will be a distribution over parameters θ', which are a function of the prior parameters θ. This property is observed for the exponential family distributions, for which some of the relevant conjugate priors are presented in table 2.1.

If we are interested in computing the probability of new observations over v given the previous observations, we can employ the same Bayesian machinery.

Lets define v^*as the set of new observations over v or a set of predictions about v. The *predictive distribution* is defined in terms of the likelihood of v^* and the posterior distribution given the previous observations, computed by Eq.(2.8) and reads

$$p(v^*|v) = \int p(v^*|\theta)\, p(\theta|v)\, d\theta.$$

2.2.3 Bayesian Linear regression

We make the ansatz to write variable y as a linear combination of a set of basis functions $\boldsymbol{\phi}_i \in \mathbb{R}^{1\times K}$ with parameter vector $\theta \in \mathbb{R}^{K\times 1}$ plus an error term ϵ. Suppose we observe y, then each observation over y will be denoted as y_i and will follow the relation

$$y_i = \boldsymbol{\phi}_i\theta + \epsilon_i$$

with ϵ_i representing a realization of a zero-mean Gaussian random variable ϵ with variance σ^2. This variable is commonly referred to *noise*, but can also be seen as everything that does not obey the linear relationship between y_i and $\boldsymbol{\phi_i}$. Adopting this point of view, we will refer to the ϵ as *residuals* for the rest of this thesis.

Under this probabilistic model the conditional probability of an observation of the target variable given the basis functions and the the *residual variance* σ^2 is

$$p\left(y_i|\boldsymbol{\phi}_i,\theta,\sigma^2\right) = \mathcal{N}\left(y_i - \boldsymbol{\phi}_i\theta, \sigma^2\right) \tag{2.10}$$

which corresponds to the likelihood term of Eq.(2.8).

Having a set of observations over y collected in the vector $\mathbf{y} = [y_1 \dots y_M]$ and basis functions $\boldsymbol{\Phi} = [\phi_1 \dots \phi_M]^T$ obeying the same distribution as Eq.(2.10), the likelihood will be

$$p\left(\mathbf{y}|\boldsymbol{\Phi},\theta,\sigma^2\right) = \prod_{i=1}^{M} \mathcal{N}\left(y_i - \boldsymbol{\phi}_i\theta, \sigma^2\right). \tag{2.11}$$

In this expression, the parameters θ and σ^{-2} are unknown, and estimating them is the main goal of linear regression.

One way to estimate model parameters θ is to take the logarithm of the likelihood function, then to maximize the likelihood. The obtained estimates are called the *maximum likelihood* estimates θ_{ML} for the parameters. This is obtained by solving

$$\frac{d}{d\theta} \ln p\left(\mathbf{y}|\Phi,\theta,\sigma^2\right) = 0$$

where $\frac{d}{d\theta}$ is the derivative w.r.t. θ . The solution to this problem is known and is equivalent to the least squares estimate (Bishop, 2001), which is given in closed form by

$$\theta_{ML} = \left(\Phi^T\Phi\right)^{-1}\Phi^T\mathbf{y}.$$

The vector θ_{ML} is an *unbiased estimate*[3] of θ (Bishop, 2001).

Alternatively, by Bayesian inference we employ Bayes theorem to infer the posterior distribution over parameters θ according to Eq.(2.8). To this end we must choose an appropriate prior $p(\theta)$ that will encode the information we possess over the set of parameters.

The conjugate prior for a normal distribution with unknown mean is also a normal distribution. We can thus assume a conjugate normal prior for parameters θ of the form

$$p(\theta) = \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_0) \tag{2.12}$$

consequently the posterior will be

$$p\left(\theta|\mathbf{y}, \Phi, \sigma^2, \mathbf{\Sigma}_0\right) = \mathcal{N}\left(\mathbf{m}', \mathbf{\Sigma}'\right) \tag{2.13}$$

where

$$\begin{aligned} \mathbf{m}^{'} &= \sigma^{-2}\mathbf{\Sigma}\Phi^T\mathbf{y} \\ \mathbf{\Sigma}' &= \mathbf{\Sigma}_0^{-1} + \sigma^{-2}\Phi^T\Phi. \end{aligned}$$

A common choice is $\mathbf{\Sigma}_0 = \rho^2\mathbf{I}$. The parameters can then be estimated by solving the equation

$$\frac{d}{d\theta}\ln p\left(\theta|\mathbf{y}, \Phi, \sigma^2, \mathbf{\Sigma}_0\right) = 0$$

which in this case corresponds to solving a regularized least squares problem (Bishop, 2001).

This solution would imply that we know the values for the residual variance σ^2 and prior variance ρ^2. To proceed in a fully Bayesian way, we would need to set a prior over these two parameters and integrate with respect to both. This problem becomes intractable, so a different approach is necessary in order to infer the parameter values.

Additionally, more sophisticated priors for $\mathbf{\Sigma}_0$ require the formulation of a probability distribution that may not be completely expressed by the common

[3] An estimator over a parameter is called unbiased if its expected value is equal to the true value of the parameter.

exponential family distributions. With this objective in mind, we treat $\mathbf{\Sigma}_0$ as another random variable whose distribution is controlled by a hierarchy of prior parameters in what is called a *Hierarchical Bayesian model.*

2.2.4 Bayesian multivariate regression

Let us extend Bayesian linear regression to a vector of correlated random variables. In this case the target variable will be the vector $\bar{\mathbf{y}} \in \mathbb{R}^N$. Each observation[4] over the target variable will be denoted as the row vector $\bar{\mathbf{y}}_i$. We assume each observation follows a linear model with basis function vector $\boldsymbol{\phi}_i \in \mathbb{R}^{1\times K}$ and parameter matrix $\mathbf{\Theta} \in \mathbb{R}^{N\times K}$. We now can define the linear relation between $\bar{\mathbf{y}}_i$ and $\boldsymbol{\phi}_i$ as a regression problem with the assumption of normally distributed i.i.d. residuals given by the vector $\mathbf{r_i}$

$$\bar{\mathbf{y}}_i = \boldsymbol{\phi}_i \mathbf{\Theta}^T + \mathbf{r_i}.$$

The residuals $\mathbf{r_i}$ form a vector that follows a multivariate normal distribution such that

$$r_i \sim \mathcal{N}(\mathbf{0}, \Sigma_\varepsilon).$$

As for the single variable case, we stack a set of M observations over the target variable into a single mathematical object. We define matrix $\mathbf{Y} \in \mathbb{R}^{M\times N}$, with each row being an observation $\bar{\mathbf{y}}_i$. Then matrix $\mathbf{\Phi} \in \mathbb{R}^{M\times k}$ is a matrix of basis functions, in which each row represents the set of basis functions for an observation. We similarly stack the $\mathbf{r_i}$ vectors into the matrix $\mathbf{R}$ columns and obtain the matrix form of this multivariate regression problem:

$$\mathbf{Y} = \mathbf{\Phi}\mathbf{\Theta}^T + \mathbf{R}. \tag{2.14}$$

where $\mathbf{R}$ is a matrix of normally distributed i.i.d residuals with variance Σ_ε. The likelihood function over $\bar{\mathbf{y}}_i$ follows

$$p(\bar{\mathbf{y}}_i|\boldsymbol{\phi}_i\mathbf{\Theta}, \Sigma_\varepsilon) \;=\; \mathcal{N}(\bar{\mathbf{y}}_i - \boldsymbol{\phi}_i\mathbf{\Theta}, \Sigma_\varepsilon).$$

We make use of the definition of *trace* of a matrix to express the likelihood as a function of the matrix Eq.(2.14) as

$$\begin{aligned} p(\mathbf{Y}|\mathbf{\Phi}, \mathbf{\Theta}, \Sigma_\varepsilon) &= \prod_{i=1}^{M} p(\bar{\mathbf{y}}_i|\boldsymbol{\phi}_i\mathbf{\Theta}, \Sigma_\varepsilon) \\ &\propto \Sigma_\varepsilon^{-n} \exp\left(-\frac{1}{2}\text{trace}\left(\mathbf{R}^T\Sigma_\varepsilon^{-1}\mathbf{R}\right)\right) \end{aligned}$$

[4]Now each observation is a vector of real numbers.

with

$$\mathbf{R} = \mathbf{Y} - \mathbf{\Phi}\mathbf{\Theta}^T.$$

We look at the trace term and computing the product we obtain

$$\text{trace}\left(\mathbf{R}^T\Sigma_\varepsilon^{-1}\mathbf{R}\right) = \text{trace}\left(\left(\mathbf{Y}^T\mathbf{Y} - 2\mathbf{\Theta}\mathbf{\Phi}^T\mathbf{Y} + \mathbf{\Theta}\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{\Theta}^T\right)\Sigma_\varepsilon^{-1}\right).$$

By completing the square and factorizing we can condition the likelihood over $\mathbf{\Theta}^T$ by considering any term that does not depend on it as a constant. Then we have

$$\text{trace}\left(\mathbf{R}^{\mathrm{T}}\Sigma_\varepsilon^{-1}\mathbf{R}\right) = \text{const} + \text{trace}\left(\left(\mathbf{\Theta}^T - \mathbf{\Theta}_{ML}^T\right)^T\left(\mathbf{\Theta}^T - \mathbf{\Theta}_{ML}^T\right)^T\Sigma_\varepsilon^{-1}\right) \quad (2.15)$$

with

$$\mathbf{\Theta}_{ML}^T = \left(\mathbf{\Phi}^T\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^T\mathbf{Y}$$

being the Maximum Likelihood estimate of $\mathbf{\Theta}^T$.

We now introduce a couple of mathematical tools that will help us reexpress the previous equations. First we start by having matrices $\mathbf{A}$ of size $m \times n$ and $\mathbf{B}$ of size $p \times q$, the *Kronecker product* is

$$\mathbf{C} = \mathbf{A} \otimes \mathbf{B}$$

with $\mathbf{C}$ being the $mp \times nq$ size matrix with elements $c_{\alpha\beta}$ such that

$$\begin{aligned} c_{\alpha\beta} &= a_{ij}b_{kl} \\ \alpha &= p(i-1)+k \\ \beta &= q(j-1)+l. \end{aligned}$$

Another tool we will employ is the *vectorization operator,* for a matrix $\mathbf{A}$ of size $m \times n$. The vec operator denoted as $vec()$ is defined as

$$\begin{aligned} \bar{\mathbf{A}} &= vec(\mathbf{A}) \\ &= \left[\begin{array}{cccccccccc} a_{11} & \dots & a_{1m} & a_{21} & \dots & a_{2m} & \dots & a_{n1} & \dots & a_{nm} \end{array}\right]. \end{aligned}$$

We will employ the properties

$$\begin{aligned} \text{trace}\left(\mathbf{A}^T\mathbf{B}\right) &= \text{vec}\left(\mathbf{A}\right)^T\text{vec}\left(\mathbf{B}\right) \\ \text{vec}\left(\mathbf{ABC}\right) &= \left(\mathbf{C}^T \otimes \mathbf{A}\right)\text{vec}\left(\mathbf{B}\right) \end{aligned}$$

to transform the trace in (2.15) into

$$\text{trace}\left(\mathbf{R}^T\Sigma_\varepsilon^{-1}\mathbf{R}\right) = const + \text{vec}\left(\mathbf{\Theta}^T - \mathbf{\Theta}_{ML}^T\right)^T\left(\Sigma_\varepsilon^{-1}\otimes\mathbf{\Phi}^T\mathbf{\Phi}\right)\text{vec}\left(\mathbf{\Theta}^T - \mathbf{\Theta}_{ML}^T\right). \tag{2.16}$$

We now define the vectors

$$\begin{aligned}\bar{\mathbf{\Theta}} &= \text{vec}\left(\mathbf{\Theta}^T\right)\\ \bar{\mathbf{\Theta}}_{ML} &= \text{vec}\left(\mathbf{\Theta}_{ML}^T\right)\end{aligned}$$

and use the property

$$\text{vec}\left(\mathbf{A}+\mathbf{B}\right) = \text{vec}\left(\mathbf{A}\right) + \text{vec}\left(\mathbf{B}\right)$$

along with the expression presented in (2.16) to derive the following equation

$$\text{trace}\left(\mathbf{R}^T\Sigma_\varepsilon^{-1}\mathbf{R}\right) = const + \left(\bar{\mathbf{\Theta}} - \bar{\mathbf{\Theta}}_{ML}\right)^T\left(\Sigma_\varepsilon^{-1}\otimes\mathbf{\Phi}^T\mathbf{\Phi}\right)\left(\bar{\mathbf{\Theta}} - \bar{\mathbf{\Theta}}_{ML}\right)^T$$

We discard the constant term and obtain the likelihood expressed over $\bar{\mathbf{\Theta}}^T$ given some observations $\mathbf{Y}$

$$p\left(\mathbf{Y}|\mathbf{\Phi},\bar{\mathbf{\Theta}},\Sigma_\varepsilon\right) = \propto \Sigma_\varepsilon^{-K}\exp\left(-\frac{1}{2}\left(\bar{\mathbf{\Theta}} - \bar{\mathbf{\Theta}}_{ML}\right)^T\left(\Sigma_\varepsilon^{-1}\otimes\mathbf{\Phi}^T\mathbf{\Phi}\right)\left(\bar{\mathbf{\Theta}} - \bar{\mathbf{\Theta}}_{ML}\right)^T\right)$$

This representation will be useful in further chapters because now we can model the parameter matrix $\mathbf{\Theta}$ as a single random vector and sample from this vector using a multivariate normal distribution. This will offer us the possibility of setting constrains over all $\mathbf{\Theta}$ coefficients in the same regression problem. As we wish to define a multivariate normal prior of the form

$$p\left(\bar{\mathbf{\Theta}}\right) = \mathcal{N}\left(0,\Sigma_0\right)$$

the posterior will be distributed according to

$$p\left(\bar{\mathbf{\Theta}}|\mathbf{\Phi},\mathbf{Y},\Sigma_\varepsilon\right) \sim \mathcal{N}\left(\bar{\mathbf{\Theta}}_{ML},\left(\Sigma_0^{-1} + \Sigma_\varepsilon^{-1}\otimes\mathbf{\Phi}^T\mathbf{\Phi}\right)^{-1}\right).$$

2.3 Graphical representation of probabilistic models

In probabilistic models, we usually deal with a set of random variables interacting with each other. The distribution of these variables may or may not be dependent on some other random variables of the model. Having a graphical representation

of these interactions is a good way to elucidate and simplify the computation of a distribution of interest.

Among the graphical representation of probabilistic models, Bayesian networks are commonly used in machine learning. In these, all the random variables of the model will be represented by circles. The directed edges represent the probabilistic relationship between these variables through conditional distributions resulting from applying the product rule (Bishop, 2001). We can thus decompose a joint distribution bu expressing it in terms of the conditional distribution of its components

$$p(v_1, v_2 \dots v_i) = p(v_i | v_1, v_2 \dots v_{i-1}) .$$

In a Bayesian network model, each variable v_i is placed as a node in a graph. Then according to our model, for each variable $v_{j \neq i}$ we draw an edge towards v_i if the pdf of $p(v_i | v_1, v_2 \dots v_{i-1})$ contains v_j . We call *parents* of v_i to the set of variables v_j with an edge towards v_i . The resulting graph represents the joint distribution over the variables and it is acyclic (no loops).

The joint distribution for a set of variables $\mathbf{V} = \{v_1, v_2 \dots v_V\}$ of size V, where each variable is denoted as v_i, will be given in terms of the product of the distribution of each variable conditioned on all its parents, denoted as pa_i. The joint distribution over variables is given by

$$p(\mathbf{V}) = \prod_{k=0}^{V} p(v_k | pa_k) .$$

As an example we will represent graphically the linear regression model from Eq.(2.11), Fig.2.1 shows the random variables y_i conditioned on the set of parameters θ. The basis functions ϕ are not random variables, the *residuals* variance σ^2 is assumed to have a fixed known value, and as such, these variables are not enclosed by a circle. A more compact way of representing the joint distribution of variables y_i is through the plate shown in Fig.2.2, which represent a set of N random variables. Finally, Fig.2.3 shows the Bayesian linear regression with a prior given by (2.12). Here we are also considering parameters σ^2 and $\mathbf{\Sigma}_0$ as random variables.

2.3.1 Conditional independence and Markov blanket

From Fig.2.3 with more detail, we see that the prior θ is determined by parameters $\mathbf{\Sigma}_0$; these prior-controlling parameters are usually called *hyper-parameters.* Addi-

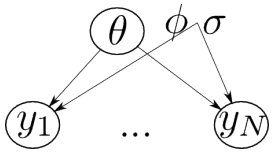


Figure 2.1: Graphical representation for the linear regression model.

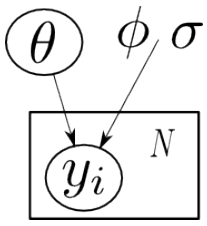


Figure 2.2: Graphical representation of the regression model using plate notation.

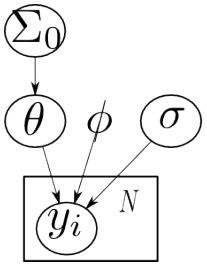


Figure 2.3: Graphical representation of the Bayesian linear regression model.

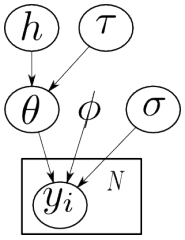


Figure 2.4: Hierarchical Bayesian model for linear regression.

tionally we are assuming $\mathbf{\Sigma}_0$ unknown and random. This kind of construction is called *hierarchical model,* and the prior-controlling distributions are called *hyper-priors*. We can extend the model at many levels by defining new hyper-priors for the hyper-priors.

In Fig.2.4 we extend the linear regression model by assigning an hyper-prior over θ controlled by hyper-parameters τ and h, which at the same time, have prior distributions $p(\tau)$ and $p(h)$. In this model, the random variables τ, h and θ form a hierarchy.

We will use the former hierarchical Bayesian model to explain conditional independence and d-separation. First let us examine the conditional distribution over the observations. Looking to the graphical representation on Fig.2.4 and applying the definition of conditional probability we get the joint distribution over y_i conditioned on θ which is

$$p(\mathbf{y}|\theta) = \prod_{i=1}^{N} p(y_i|\theta)$$

and each variable y_i is said to be *conditional independent* from the variables $y_{j\neq i}$.

Now suppose we are interested in the joint distribution of the hyper-parameters τ and h given an observation over θ. By following the graph, we see that the

distribution is given by

$$\begin{aligned} p(\tau, h|\theta) &= \frac{p(\tau, h, \theta)}{p(\theta)} \\ &= \frac{p(\tau)\, p(h)\, p(\theta|\tau, h)}{p(\theta)} \end{aligned}$$

which does not factorize as $p(\tau|\theta)\, p(h|\theta)$. In this case τ and h are not conditional independent given θ. D-separation is a method to determine which variables are conditionally independent of another set of variables in a directed acyclic graphical model. The method is reviewed in any machine learning textbooks, such as (Bishop, 2001). With this method it is possible to reduce the complexity of the model by deriving a factorized representation of the joint distribution.

An important concept related to conditional independence is the *Markov blanket*. We will explain the latter using the same hierarchical Bayesian model as before. Suppose we are interested in the conditional distribution of θ given all the other variables. In order to derive a general representation for the Markov blanket we will encompass all the variables in the set $\mathbf{V} = \{v_1, v_2, v_3, v_4, v_5\}$, from which $v_1 = \theta$, $v_2 = \tau$, $v_3 = h$, $v_4 = \sigma$ and $v_5 = \mathbf{y}$. The conditional distribution of v_1 given all the other variables, which are contained in the set $\mathbf{V}_{\setminus 1} = \{v_2, v_3, v_4, v_5\}$, is

$$\begin{aligned} p\left(v_1|\mathbf{V}_{\setminus 1}\right) &= \frac{p(\mathbf{V})}{\int p(\mathbf{V})\, dv_1} \\ &= \frac{\prod_{k=1}^{V} p(v_k|pa_k)}{\int \prod_{k=1}^{V} p(v_k|pa_k)\, dv_1} \end{aligned}$$

all the elements of $p(v_k|pa_k)$ that do not depend on v_1 can be taken out of the integral in the denominator and canceled out with the equivalent terms in the numerator. The remaining set will consist of the parent variables of v_1, the children of v_1 and the co-parents of v_1 (variables which share children with v_1). This set is known as the *Markov blanket* for v_1. In general, the Markov Blanket for any v_i in a graphical model is given by its *parents, children* and *co-parents*.

In our example, the Markov blanket for θ includes its parents τ and h, its child node $\mathbf{y}$ and its co-parent σ. Meanwhile the Markov blanket for h will only be contain its child node θ and co-parent τ. The Markov blanket is essential for the inference scheme based on Gibbs sampling that will be presented in the next section.

2.4 Approximate inference of model parameters through sampling

The basic idea of sampling is to substitute an integration over a probability distribution $p(\mathbf{V})$ by averaging a set of samples drawn independently from said distribution. As such, if we wish to estimate the expected value of a function

$$\mathbb{E}[f] = \int f(\mathbf{V}) p(\mathbf{V}) \, d\mathbf{V}$$

we draw N samples $\mathbf{V}^{(n)}$ from $p(\mathbf{V})$ and compute the average by evaluating f for each sample,

$$\hat{f} = \frac{1}{N} \sum_{n=1}^{N} f\left(\mathbf{V}^{(n)}\right)$$

here $\hat{f}$ is an unbiased estimator of $\mathbb{E}[f]$ and its variance decreases with the square root of the number of samples.

In the case of parameter inference in graphical models, we are interested in sampling from the joint distribution $p(\mathbf{V})$ given a set of observations over one or more variables. We will call the subset of observed variables $\mathbf{V}_I$. The *joint posterior* distribution is given by

$$p\left(\mathbf{V}_{\setminus I} | \mathbf{V}_I\right) = \prod_{k=1}^{V} p(v_k | pa_k, \mathbf{V}_I)$$

which is usually intractable. Methods such as ancestral sampling and importance sampling can be employed to sample from said distributions, but are generally not suited for multidimensional problems (Barber, 2012; Bishop, 2001).

A family of methods called Markov Chain Monte Carlo methods (MCMC) are a powerful way of drawing samples of multidimensional and complicated distributions. These samples are drawn conditionally from previous samples given a *transition probability*. If these transitions comply with certain characteristics, they will converge to $p\left(\mathbf{V}_{\setminus I} | \mathbf{V}_I\right)$.

2.4.1 Markov chain Monte Carlo methods

A *Markov chain* is a sequence of random variables $\mathbf{V}^{(1)}$,$\mathbf{V}^{(2)}$,...,$\mathbf{V}^{(n)}$, in which states $\mathbf{V}^{(n-1)}$and $\mathbf{V}^{(n+1)}$ are independent given $\mathbf{V}^{(n)}$ (present state). Thus the distribution over any $\mathbf{V}^{(n)}$ follows

$$p\left(\mathbf{V}^{(n+1)} | \mathbf{V}^{(1)}, \mathbf{V}^{(2)}, \ldots, \mathbf{V}^{(n)}\right) = p\left(\mathbf{V}^{(n+1)} | \mathbf{V}^{(n)}\right)$$

where $p\left(\mathbf{V}^{(n+1)}|\mathbf{V}^{(n)}\right)$ is called the *transition probability.*

Having the transition probability $q\left(\mathbf{V}|\mathbf{V}^{'}\right)$, in which $\mathbf{V}^{'}$ represents any previous member of the chain, a *stationary distribution* with respect to the Markov Chain obeys

$$p\left(\mathbf{V}\right) = \sum_{\mathbf{V}^{'}} q\left(\mathbf{V}|\mathbf{V}^{'}\right) p\left(\mathbf{V}^{'}\right).$$

In MCMC methods we aim at selecting a conditional distribution $q\left(\mathbf{V}|\mathbf{V}^{'}\right)$ such that the desired multivariate joint distribution $p\left(\mathbf{V}_{\setminus I}|\mathbf{V}_I\right)$ is its stationary distribution for arbitrary initial state $\mathbf{V}^{(1)}$. This additional requirement is called *ergodicty,* and q_e is said to be the *equilibrium distribution.* We then can proceed to sample from $q_e\left(\mathbf{V}\right)$ which is equivalent to drawing *dependent* samples from $p\left(\mathbf{V}\right)$. If the transition probabilities are the same for all states, the Markov chain is *homogeneous* and the stationary distribution is given by

$$q_e\left(\mathbf{V}\right) = \int q\left(\mathbf{V}|\mathbf{V}^{'}\right) q_e\left(\mathbf{V}^{'}\right) d\mathbf{V}^{'}$$

A Markov chain obeys the *detailed balance property(Bishop, 2001)* if

$$p\left(\mathbf{V}\right) q\left(\mathbf{V}|\mathbf{V}^{'}\right) = p\left(\mathbf{V}^{'}\right) q\left(\mathbf{V}^{'}|\mathbf{V}\right)$$

This condition ensures that we can transition back and forth to any given state from any other state in the chain. As such it is called *reversible Markov chain.* Being reversible is a sufficient (but not necessary) condition for being a stationary distribution. This is easily verified by

$$\sum_{\mathbf{V}^{'}} p\left(\mathbf{V}^{'}\right) q\left(\mathbf{V}^{'}|\mathbf{V}\right) = \sum_{\mathbf{V}^{'}} p\left(\mathbf{V}\right) q\left(\mathbf{V}^{'}|\mathbf{V}\right) = p\left(\mathbf{V}\right) \sum_{\mathbf{V}^{'}} q\left(\mathbf{V}^{'}|\mathbf{V}\right) = p\left(\mathbf{V}\right)$$

2.4.2 Metropolis-Hastings algorithm

The Metropolis-Hastings algorithm is an MCMC method. Suppose we want to draw samples from the distribution

$$p\left(\mathbf{V}\right) = \frac{\tilde{p}\left(\mathbf{V}\right)}{Z_p} \tag{2.17}$$

where the normalizing constant Z_p is intractable. Metropolis-Hastings allows us to sample from 2.17 without computing Z_p.

We first define a proposal distribution $q\left(\mathbf{V}|\mathbf{V}'\right)$ from which we can draw samples. Having a the current state (last accepted sample) $\mathbf{V}^{(n)}$, we draw a

sample of $\mathbf{V}^{(n+1)}$ from $q\left(\mathbf{V}^{(n+1)}|\mathbf{V}^{(n)}\right)$. We accept $\mathbf{V}^{(n+1)}$ with probability $A\left(\mathbf{V}^{(n+1)},\mathbf{V}^{(n)}\right)$ given by

$$A\left(\mathbf{V}^{(n+1)},\mathbf{V}^{(n)}\right)=\min\left(1,\frac{\tilde{p}\left(\mathbf{V}^{(n+1)}\right)q\left(\mathbf{V}^{(n)}|\mathbf{V}^{(n+1)}\right)}{\tilde{p}\left(\mathbf{V}^{(n)}\right)q\left(\mathbf{V}^{(n+1)}|\mathbf{V}^{(n)}\right)}\right). \tag{2.18}$$

Then the equilibrium distribution $q_e(\mathbf{V})$ will be the desired $p(\mathbf{V})$ for the Markov Chain with transition probability $q(\mathbf{V}|\mathbf{V}')\,A\left(\mathbf{V},\mathbf{V}^{'}\right)$ (Bishop, 2001). We show that the detailed balance condition is obeyed

$$\begin{aligned} p(\mathbf{V})\,q\left(\mathbf{V}'|\mathbf{V}\right)A\left(\mathbf{V}^{'},\mathbf{V}\right) &= \min\left(p(\mathbf{V})\,q\left(\mathbf{V}'|\mathbf{V}\right),p\left(\mathbf{V}^{'}\right)q\left(\mathbf{V}|\mathbf{V}^{'}\right)\right)\\ &= \min\left(p\left(\mathbf{V}^{'}\right)q\left(\mathbf{V}|\mathbf{V}^{'}\right),p(\mathbf{V})\,q\left(\mathbf{V}'|\mathbf{V}\right)\right)\\ &= p\left(\mathbf{V}^{'}\right)q\left(\mathbf{V}|\mathbf{V}^{'}\right)A\left(\mathbf{V},\mathbf{V}^{'}\right) \end{aligned}$$

The main drawback of this algorithm is finding a proposal distribution $q\left(\mathbf{V}|\mathbf{V}^{'}\right)$ that would allow us to efficiently sample from $p(\mathbf{V})$. If we choose a proposal with high acceptance rate, we may get a set of highly correlated samples Bishop (2001). This implies that we are exploring the space of the target distribution in small steps, thus making it necessary to have more samples to properly estimate $p(\mathbf{V})$ from $q_e(\mathbf{V})$. On the other hand, if the rejection rate is too high, It may take longer to get a set of samples from $q_e(\mathbf{V})$ thus again, requiring more time to get the desired samples.

2.4.3 Gibbs sampling

Gibbs sampling is one of the most popular MCMC methods. At each step $n+1$, one draws a sample from the conditional distribution $p\left(v_i^{(n+1)}|\mathbf{V}_{\setminus i}^{(n)}\right)$, where $\mathbf{V}_{\setminus i}^{(n)}$ represents a sample of all elements of $\mathbf{V}$ except v_i. As explained in Section 2.3.1 this probability distribution only depends on the Markov blanket of node v_i in the graphical model.

To see how the stationary distribution of a Gibbs sampler converges to the desired distribution $p(\mathbf{V})$, we will present its derivation according to (Barber, 2012). Let's define the Markov transition probabilities

$$\begin{aligned} q\left(\mathbf{V}^{(n+1)}|\mathbf{V}^{(n)}\right) &= \sum_{i=1}^{N} q\left(\mathbf{V}^{(n+1)}|\mathbf{V}^{(n)},i\right)q(i)\\ q\left(\mathbf{V}^{(n+1)}|\mathbf{V}^{(n)},i\right) &= p\left(v_i^{(n+1)}|\mathbf{V}_{\setminus i}^{(n)}\right)\prod_{j\neq i}\delta\left(v_i^{(n+1)},v_j^{(n)}\right) \end{aligned}$$

where $q(i) > 0$, and $\sum_i q(i) = 1$. The variable v_i is selected and then the conditional distribution is evaluated using the state of the other variables in a previous sample; then the stationary distribution is

$$\begin{aligned}
\int q\left(\mathbf{V}^{'}|\mathbf{V}\right) p(\mathbf{V})\, d\mathbf{V} &= \sum_{i=1}^{N} q(i) \int q\left(\mathbf{V}^{'}|\mathbf{V}, i\right) p(\mathbf{V})\, d\mathbf{V} \\
&= \sum_{i=1}^{N} q(i) \int \prod_{j \neq i} \delta\left(v_i^{'}, v_j\right) p\left(v_i^{'}|\mathbf{V}_{\setminus i}\right) p\left(v_i, \mathbf{V}_{\setminus i}\right) d\mathbf{V} \\
&= \sum_{i=1}^{N} q(i) \int p\left(v_i^{'}|\mathbf{V}_{\setminus i}^{'}\right) p\left(v_i, \mathbf{V}_{\setminus i}^{'}\right) dv_i \\
&= \sum_{i=1}^{N} q(i)\, p\left(v_i^{'}|\mathbf{V}_{\setminus i}^{'}\right) p\left(\mathbf{V}_{\setminus i}^{'}\right) \\
&= \sum_{i=1}^{N} q(i)\, p\left(\mathbf{V}^{'}\right) \\
&= p\left(\mathbf{V}^{'}\right).
\end{aligned}$$

One of the main advantages of Gibbs sampling is that conjugacy in the conditional distributions can be exploited in order to sample exactly from exponential family distributions.

Gibbs sampling is equivalent to perform Metropolis-Hastings with the transition probability $q_i\left(\mathbf{V}^{(n+1)}|\mathbf{V}^{(n)}\right) = p\left(v_i^{(n+1)}|\mathbf{V}_{\setminus i}^{(n)}\right)$. At each sampling step i, we have

$$\mathbf{V}_{\setminus i}^{(n+1)} = \mathbf{V}_{\setminus i}^{(n)}$$

and

$$p\left(\mathbf{V}^{(n)}\right) = p\left(v_i^{(n)}|\mathbf{V}_{\setminus i}^{(n)}\right) p\left(\mathbf{V}_{\setminus i}^{(n)}\right)$$

then the acceptance probability is

$$\begin{aligned}
A\left(\mathbf{V}^{(n+1)}, \mathbf{V}^{(n)}\right) &= \min\left(1, \frac{p\left(\mathbf{V}^{(n+1)}\right) q_i\left(\mathbf{V}^{(n)}|\mathbf{V}^{(n+1)}\right)}{p\left(\mathbf{V}^{(n)}\right) q_i\left(\mathbf{V}^{(n+1)}|\mathbf{V}^{(n)}\right)}\right) \\
&= \min\left(1, \frac{p\left(v_i^{(n+1)}|\mathbf{V}_{\setminus i}^{(n+1)}\right) p\left(\mathbf{V}_{\setminus i}^{(n+1)}\right) p\left(v_i^{(n)}|\mathbf{V}_{\setminus i}^{(n+1)}\right)}{p\left(v_i^{(n)}|\mathbf{V}_{\setminus i}^{(n)}\right) p\left(\mathbf{V}_{\setminus i}^{(n)}\right) p\left(v_i^{(n+1)}|\mathbf{V}_{\setminus i}^{(n)}\right)}\right) \\
&= 1.
\end{aligned}$$

Thus all proposed samples are accepted. As mentioned in Section 2.4.2, a high acceptance rate implies that we may be drawing highly correlated samples from the desired distribution. A drawback of Gibbs sampling and MCMC in

general is that the samples are not independent, and as such, a greater number of samples may be necessary to correctly approximate the target distribution[5]. Also, if the initial state has low probability, the first samples drawn may not be representative of the whole distribution, so usually effective samples are taken after a *burn-in* period. Related to this, it is generally unknown how many samples are required to converge to the equilibrium distribution, so convergence tests may be necessary (Cowles and Carlin, 1996). One of such tests, proposed by (Geweke, 1992), compares the means of two parts of the chain and computes a score based on the *z-test.*

In Fig.2.5 we illustrate Metropolis-Hastings and Gibbs sampling. Suppose we want to sample from $p(v_1, v_2)$. On the left $p(v_1, v_2)$ is given by a multivariate normal distribution (red) and the proposed distribution is another multivariate normal with fixed width (variance). Depending on the variance of the proposal, the "faster" it will cover the desired distribution, but the rejection rate will be higher. If the width is too small the rejection rate will be less, but it would require more "moves". In the center we see Gibbs sampling applied to the same target distribution, the algorithm will move in steps given by the variance of $p(v_1|v_2)$ and $p(v_2|v_1)$. The time to converge to the equilibrium distribution will again depend on these two distributions. Finally to the right we have a case for which Gibbs sampling fails. Here we have two perfectly correlated elements. In this case the algorithm will get "stuck" in a particular mode and will not visit the whole state space. In general Gibbs sampling will fail if the target distribution has two or more modes with no paths connecting them.

There are some techniques to improve or modify the functioning of Gibbs sampling Bishop (2001), among the most relevant are:

- *Collapsed Gibbs sampling (Liu, 1994),* in this method one or more variables are integrated out of the conditional steps. For example, assume we wish to sample from $p(v_1, v_2, v_3)$ using a Gibbs sampling scheme, but we wish to marginalize out v_2[6]. We then derive the collapsed Gibbs sampling steps using these conditional distributions as transition probabilities

$$q_1\left(\mathbf{V}^{(n+1)}|\mathbf{V}^{(n)}\right) = \int p(v_1|v_2, v_3)\, dv_2$$

[5]Usually, every n-th sample is kept and the rest discarded, in order to get independent samples.

[6]We may choose to integrate v_2 out of the sampling scheme if there is a known analytical solution

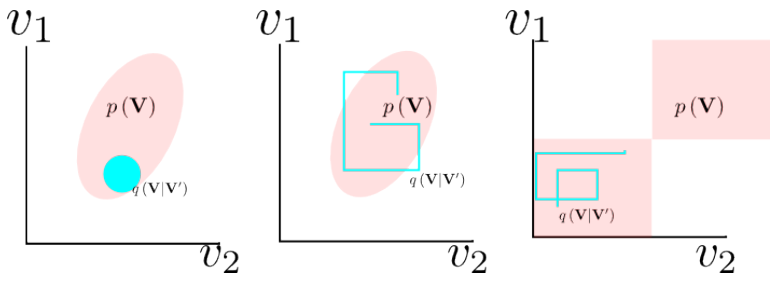


Figure 2.5: On left we have an illustration of Metropolis-Hastings Sampling. We wish to draw samples from $p(\mathbf{V})$(red) , at each step we draw a sample from the proposal distribution (light blue). The width of the proposal determines the acceptance rate. In the middle we have a Gibbs sampling scheme in which the length of each step is determined by a conditional distribution of one variable given the other. In the right we have a bi-modal distribution for two perfectly correlated elements in a vector. Gibbs sampling gets "stuck" in one mode without being able to move to the other.

$$q_2\left(\mathbf{V}^{(n+1)}|\mathbf{V}^{(n)}\right) = \int p(v_3|v_2,v_1)\,dv_2.$$

- *Blocked Gibbs sampling,* here the joint distribution of a group of variables conditioned on the rest is derived and samples are drawn from this joint distribution. As we will be sampling from multivariate distributions in the following chapter, the application of this technique is straightforward. For example, assume we wish to sample from the joint distribution $p(v_1,v_2,v_3,v_4)$, and assume that $p(v_1v_2|v_3,v_4)$ is easy to sample from, then the blocked Gibbs samples uses the transition probabilities

$$\begin{aligned} q_1\left(\mathbf{V}^{(n+1)}|\mathbf{V}^{(n)}\right) &= p(v_1v_2|v_3,v_4) \\ q_2\left(\mathbf{V}^{(n+1)}|\mathbf{V}^{(n)}\right) &= p(v_3|v_2,v_3,v_4) \\ q_3\left(\mathbf{V}^{(n+1)}|\mathbf{V}^{(n)}\right) &= p(v_4|v_1,v_2,v_3)\,. \end{aligned}$$

2.5 Discussion

In this chapter we presented linear ODE-based models for representing gene expression; we argued that they are a reasonable way to approximate a systems dynamics under regular oscillations. We then derived the corresponding representation in the frequency domain using the DFT thereby formulating the problem of parameter estimation as a matrix equation.

Instead of relying on frequentist approaches such as regularized regression we opted for a Bayesian approach. Regression specifically offers some disadvantages in our scenario. Least squares regression is an unbiased estimator of the coefficients but solutions are not sparse. Regularized regression is sparse but does not allow a seamless integration of additional data sources. Even if the new problem is tractable the solutions may not offer useful information about statistical properties of the solutions. These properties are fundamental for our experimental design approach as will be presented in Chapter 5.

We reviewed the selected method for solving this problem, Bayesian inference. We presented the Bayesian multivariate linear regression framework. This will allow us to work with the inherent uncertainties of experimental measurements and errors in approximation by assuming these errors are normally distributed. Being a Bayesian model, it leverages our previous knowledge about the system and its characteristics through the use of prior distributions. The use of prior distributions will also allow us to integrate additional sources of information seamlessly by using a hierarchy of priors. Complex distributions and structural constrains will be encoded and added in a principled way.

We also introduced Bayesian hierarchical models and their representation through graphical models. Finally explored the MCMC family of methods for estimating parameters over statistical models. These methods are of widespread use and provide a robust methodology for parameter inference.

Now that all the tools necessary for inference over a Hierarchical Bayesian model are presented,we aim at implementing a special kind of model for sparse regression in the next chapter . This will allow us to integrate both structure and dynamics of the network in a single model.

Chapter 3

Structure learning for Oscillatory genetic regulatory networks.

3.1 Introduction

As explained in Chapter 1, Genetic regulatory networks are at the core of many biological oscillators. These networks can sustain oscillatory behavior in protein levels through specific architectures involving multiple feedback loops of transcriptional regulation. For example, a transcriptional oscillator is thought to drive the *Arabidopsis thaliana* circadian clock through mutual repression of three transcriptional regulators (Pokhilko et al., 2012; McClung, 2011). The cell cycle is another oscillatory process, which controls cell division and duplication. In the case of *Saccharomyces cerevisiae*, experiments and dynamical models suggest that the cell cycle is the result of a transition between two self maintaining steady states, driven by two antagonistic classes of proteins (Chen et al., 2004). Evidence suggests that a transcriptional network is an important part of this mechanism (Spellman et al., 1998; Li et al., 2004; Orlando et al., 2008).

These oscillators have been the subject of study for many years, but uncovering the exact mechanism is a challenge that involve many complex chemical, genetic and physiological components. It is therefore important to devise computational statistical methods which may guide experimental analyses by inferring potential regulatory interactions directly from time series gene expression data, which is usually easier to obtain.

Network inference is a well established and rich domain of research in systems biology. State of the art methods for regulatory network inference include a

wide variety of techniques from statistics and machine learning. For example, mutual information between gene expression levels under different experimental conditions is used by ARACNE (Margolin et al., 2006) and CLR (Faith et al., 2007), two of the most widely used methods for network reconstruction. GENIE3 (Huynh-Thu et al., 2010), another method which was a top performer at the DREAM network inference challenges, and the more recent extension Jump3 (Huynh-Thu and Sanguinetti, 2015) use random forests to produce a weighted ranking over the network edges. Other methods recently used include regularized regression (Haury et al., 2012), ANOVA (Kuffner et al., 2012) and Hierarchical Gaussian models (Li et al., 2006) Most of these methods focus on steady state data, which is by definition not available for oscillatory networks.

Regularisation-based and Bayesian methods can also be adapted to time series data. Dynamic Bayesian Networks have long been a popular choice in network inference (Dondelinger et al., 2013b; Oates and Mukherjee, 2012, e.g.). Such methods present considerable advantages in being able to quantify uncertainty and to incorporate prior knowledge, but are often severely limited by computational constraints. Optimisation-based methods based on regularised regression (Bonneau et al., 2006, e.g.) present often a scalable alternative at the cost however of some modeling flexibility.

In this chapter we use a first order model of the system dynamics to constrain the network inference; we explicitly take advantage of the oscillatory behavior of the system by employing the frequency-domain representation presented in Eq.(2.7). We build a hierarchical Bayesian model over the network dynamics which can set and infer structural constraints and account for the inevitable uncertainty that experimental settings convey. Furthermore, the method can easily integrate non-trivial side information, for example in the form of sequence similarity between promoter sequence of genes. Experimental results on real and simulated data highlight that the method offers an effective and flexible platform for statistical inference in oscillatory systems, and can uncover non-trivial biological information.

the next section describes the methodology we use, reviewing the linear time-invariant approximation we use as well as introducing the Bayesian hierarchical framework for network inference. We then present an experimental evaluation on three data sets: a synthetic data set from the DREAM network inference challenge, a simulated data set obtained from a state of the art model of the A.

thaliana circadian clock (Pokhilko et al., 2010), and a real data set from the yeast *S. cerevisiae* cell cycle (Orlando et al., 2008). We then conclude the chapter by discussing our method in the light of these experimental results and the existing literature on network inference.

3.2 Modelling and inference in Frequency domain

In Section 2.1.3 Eq.(2.7) we aproximated oscillatory dynamics in the DFT domain by an LTI formulation. Here the DFT of the gene expression levels of N network components over M time samples is computed. These DFT coefficients are stored in a dimension M real vector by stacking the real and imaginary part of the first $M/2$ coefficients. These column vectors are gathered in the spectra matrix $\mathbf{X}^q$, where the superindex q indicates the number of the experiment. The matrix $\mathbf{U}^q$ represents the input signal of the network (light input for example). The matrix $\dot{\mathbf{X}}$ represents the time derivative of the gene expression levels, computed in the frequency domain by multiplying matrix $\mathbf{X}$ by the derivative factor matrix $\mathbf{D}$ and coefficient matrices $\mathbf{A}$ and $\mathbf{C}$ represent the linerised system dynamics encoding interaction parameters and decay rates.

To account for any discrepancies between the linearized model and the true system dynamics, we assume normally distributed error with variance σ_D^2. Thus Eq.(2.7) plus error is

$$\dot{\mathbf{X}}^q = \mathbf{X}^q\mathbf{A}^T + \mathbf{U}^q\mathbf{C}^T + \mathbf{R}^q \tag{3.1}$$

where $\mathbf{R}^q \in \mathbb{R}^{M\times N}$, such that their elements r_{ij} follow the distribution

$$r_{ij} \sim \mathcal{N}\left(0, \sigma_D^2\right).$$

Then the joint distribution for the residuals are given by

$$\mathrm{p}\left(\mathbf{R}^q|\dot{\mathbf{X}}^q, \mathbf{X}^q, \mathbf{A}, \mathbf{U}, \mathbf{C}, \sigma_D\right) \;\propto\; \sigma_D^{-\mathrm{M}} \exp\left(-\frac{1}{2\sigma_D^2}\mathrm{trace}\left(\mathbf{R}^q\right)^T\mathbf{R}^q\right) \tag{3.2}$$

where the residuals $\mathbf{R}^q$ (mis match between the linearized model and the true system dynamics) come from eq 2.7 factorized as

$$\mathbf{R}^q = \dot{\mathbf{X}}^{\mathrm{q}} - \begin{bmatrix} \mathbf{X}^q & \mathbf{U}^q \end{bmatrix} \begin{bmatrix} \mathbf{A^T} \\ \mathbf{C}^T \end{bmatrix}. \tag{3.3}$$

In general, multiple time series, each one being an experiment or a replicate, may be available. Denoting with Q the number of time series, the overall joint probability, under an assumption of normal i.i.d error between series, can be generalized as:

$$\mathrm{P}\left(\{\mathbf{R}^q\}\,|\,\left\{\dot{\mathbf{X}}^q\right\}\{\mathbf{X}^q\}\,\mathbf{A},\mathbf{U},\mathbf{C},\sigma_D\right)=\prod_{q=1}^{\mathrm{Q}}\mathrm{P}\left(\mathbf{R}^q|\dot{\mathbf{X}}^q,\mathbf{X}^q,\mathbf{A},\mathbf{U},\mathbf{C},\sigma_D\right)$$

and explicitly is

$$\begin{aligned}\mathrm{P}\left(\{\mathbf{R}^q\}\,|\mathbf{.}\right) &\propto \sigma_D^{-\mathrm{M}Q}\exp\left(-\frac{1}{2\sigma_D^2}\sum_{q=1}^{Q}\mathrm{trace}\left(\left(\mathbf{R}^q\right)^T\mathbf{R}^q\right)\right)\\ &\propto \sigma_D^{-\mathrm{M}Q}\exp\left(-\frac{1}{2\sigma_D^2}\mathrm{trace}\left(\sum_{q=1}^{Q}\left(\mathbf{R}^q\right)^T\mathbf{R}^q\right)\right)\end{aligned}\tag{3.4}$$

which is a product of Gaussian densities.

Notice that the form of Eq.(3.4) is identical to a regression problem as presented in Section 2.2.4 [1] with likelihood given by Eq.(3.2). The inference problem of estimating the interaction and input response matrices $\left[\begin{array}{cc}\mathbf{A^T} & \mathbf{C^T}\end{array}\right]^{\mathrm{T}}$ in Eq.(4.11) can therefore be attacked using the vast repertoire of regression methods. Regularized regression methods have been tested in a network inference context, see Charbonnier et al. (2010); Bergersen et al. (2011); Bonneau et al. (2006); Haury et al. (2012). Here, we opt for a hierarchical Bayesian approach, that will allow us to leverage prior knowledge and integrate other sources of information.

3.2.1 Hierarchical Bayesian modeling

To interpret dynamical systems in a network perspective, we assume that the interaction matrix in our LTI representation 3.1 has a sparse structure representing discrete interactions between regulators and target genes. We introduce the *structural adjacency matrix* $\mathbf{H}\in\mathbb{R}^{\mathrm{N}\times\mathrm{N}}$, which sits at the top of the hierarchy. This matrix contains elements $h_{ij}=1$ if gene j regulates gene i for $i\neq j$. In this Bayesian approach, a sparsity inducing prior over elements of $\mathbf{H}$ is necessary to aid identifiability and interpretability. The prior form chosen for elements h_{ij} is

[1] With the derivative spectra taking the place of the target variables $\mathbf{Y}$ and the observed spectra being the basis functions $\mathbf{\Phi}$

a Bernoulli distribution, with parameter w which has a Beta distribution prior due to conjugacy.

We chose a spike and slab prior to relate the connection matrix $\mathbf{H}$ and interaction matrix $\mathbf{A}$. This distribution consists of a mixture of a degenerate distribution and a long tailed distribution. The form chosen is derived from the one presented in Ishwaran and Rao (2005), where the a_{ij} elements are drawn from a scale-mixture model where a zero-mean normal distribution has variance governed by hyper-parameter τ_{ij}. In this form, the hyper-variance $\mathrm{h}_{ij}\tau_{ij}^2$ has a continuous bimodal distribution. With this prior, the posterior distribution of the less relevant parameters is shrunk towards zero and the non-zero elements are selected by the distributions tail. The advantage of the continuous distribution implied by the scale-mixture model of Ishwaran and Rao (2005) lies primarily in the fact that we avoid the need to parametrize these bimodal distributions manually.

Thus, the hierarchical model is defined by equations:

$$\begin{array}{rcl} \mathrm{P}\left(\{\mathbf{R}^q\}\,|\,\left\{\dot{\mathbf{X}}^q\right\},\{\mathbf{X}^q\}\,\mathbf{A},\mathbf{U},\mathbf{C},\sigma_D\right) & = & \prod_{q=1}^{\mathrm{Q}}\mathrm{P}\left(\mathbf{R}^q|\dot{\mathbf{X}}^q,\mathbf{X}^q,\mathbf{A},\mathbf{U},\mathbf{C},\sigma_D\right) \\ \mathrm{P}\left(a_{ij}|h_{ij},\tau_{ij}\right) & \sim & \mathcal{N}\left(0,h_{ij}\tau_{ij}^2\right) \\ \mathrm{P}\left(h_{ij}|w\right) & \sim & (1-w)\,\delta_{v0}+w\delta_1 \\ \pi\left(w\right) & \sim & \mathrm{Beta}\left(\mathrm{a}_1,\mathrm{a}_2\right) \\ \pi\left(\tau^{-2}\right) & \sim & Gamma\left(b_1,\mathrm{b}_2\right) \\ \pi\left(\sigma_D^{-2}\right) & \sim & Gamma\left(\mathrm{c}_1,\mathrm{c}_2\right). \end{array} \tag{3.5}$$

The parameter σ_D accounts for uncertainty related to noise and model mismatch, for example arising from the linear approximation to the system dynamics. The parameter $v0$ is introduced for numerical stability and is fixed to the value of 0.005. The hyperparameters $a_{1,2}$, $b_{1,2}$ and $c_{1,2}$ can be fixed to reflect prior beliefs, or set to vague values to reflect prior ignorance; in the rest of the paper they are set to the default values of (1, 1), (5 , 50) and (0.001,0.001) respectively.

3.2.2 Sequence information integration

A major advantage of hierarchical modeling is the possibility of integrating different data sources. By branching from the top of the hierarchy, we can define models for different network related characteristics and keep all the information coupled by the top of the hierarchy. For example, protein interaction and binding

data from ChIP-chip or ChIP-seq experiments can be used in a straightforward manner to modulate the prior probabilities over matrix $\mathbf{H}$, for example by adjusting the parameter w for individual edges.

Hierarchical models also allow us to exploit more subtle sources of structural information derived from an analysis of sequence information. Transcription factors bind to the promoter region of their targets by recognizing specific motifs, short DNA words; thus co-regulated genes (genes that are regulated by a common transcription factor) should share common motifs in their promoted regions. We use this information to draw the basic model for our sequence integration approach. As the transcription binding sites share a common motif, we assume that the similarity between two promoter regions varies proportionally to the number of shared regulators. In this way, an observed pairwise similarity matrix $\mathbf{S} = [s_{ij}]$ between gene promoters, derived from a multiple alignment method like (Sievers et al., 2011) or an alignment-free method (Sims et al., 2009), can be related to the structural adjacency matrix at the top of the hierarchical model. Assuming for simplicity a Gaussian observation model, we can then incorporate sequence similarity by positing the following relationship between promoter similarity scores and the structural adjacency matrix

$$\mathrm{p}\left(s_{ij}|\mathbf{H},\boldsymbol{\beta},\sigma_{seq}\right) \propto \sigma_{seq}^{-1/2} \exp\left(-\frac{1}{2\sigma_{seq}^{2}}\left(s_{ij} - \sum_{l=1}^{\mathrm{N}} h_{il}h_{jl}\beta_{l}\right)^{2}\right) \tag{3.6}$$

Here the parameter $\{\beta_l\}$ $1 \leq l \leq N$ is the similarity "induced" by the $l-th$ transcription factor (a proportionality constant), and the product $h_{il}h_{jl}$ equals 1 if and only if genes i and j are both regulated by l. In Fig.3.1 we illustrate the basic idea behind this model, with three promoter sequences, sequence 2 and 3 are bound by three factors in common. Even though the binding sites are not identical, the similarity between their sequences should help to differentiate them from a promoter region with only one of these factors binding to it.

This model is a form of additive clustering (Mirkin, 1987). By conditioning on $\mathbf{H}$, we can derive the distribution $p(\beta_l|...)$, which is a Gaussian with non-negativity constraints, (see appendix Eq.(4)). This distribution can be used for sampling posterior values of β; in our applications, however, we preferred to fix the value of β to its non-negative maximum likelihood solution, effectively approximating this conditional posterior with a δ function. The similarity score variance σ_{seq} is given a weakly informative inverse Gamma prior. By completing the square we

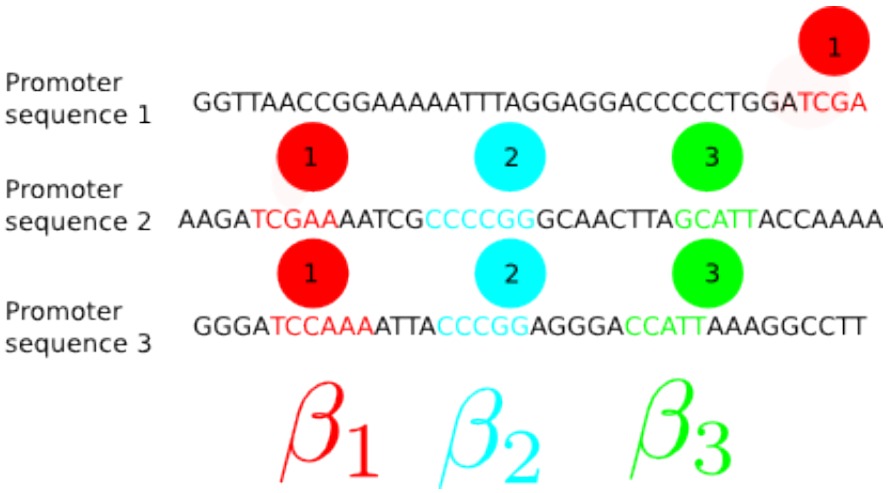


Figure 3.1: Illustration of the sequence similarity model. Promoter sequences 2 and 3 are regulated by the same 3 transcription factors. The corresponding binding sites induce a greater sequence similarity between these promoters regions. These induced similarities are represented by proportionality constants β.

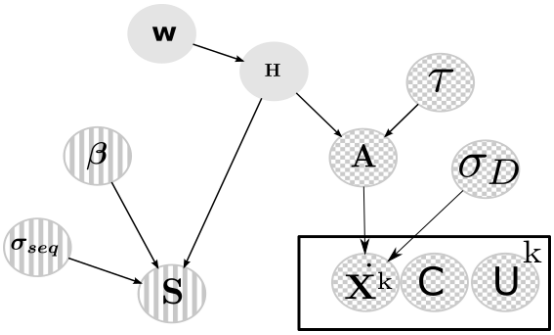


Figure 3.2: Hierarchical Bayesian model, on top of the hierarchy (green) lies the adjacency matrix $\mathbf{H}$ and sparsity parameter w. In chequered circles the frequency-domain gene expression model and its parameters. In yellow the stripes sequence similarity and its parameters.

can derive a Gaussian distribution for the β_l parameters, for its derivation and estimation see Appendix Section 1. The overall structure of the model is depicted graphically in Fig.3.2.

3.2.3 Inference

Inference of parameters $\{\mathbf{A}, \mathbf{C}, \mathbf{H}, \sigma_D, w, \boldsymbol{\tau}\}$ is done through a simple Gibbs sampling scheme. Given conjugacy among distributions, sampling of these parameters is straightforward for all distributions except $\mathrm{p}(\beta_l)$. This distribution is not con-

jugate, so a Metropolis within Gibbs would be necessary for exact inference. In order to improve performance and given the fact that retrieving the distribution over β_l is not an objective, we use the non-negative least square estimate for the vector $\boldsymbol{\beta}$. Convergence was tested by applying Geweke diagnostic (Geweke, 1992) over the last 1000 samples of matrix $\mathbf{H}$. Mathematical derivations of the required conditional posteriors and the general sampling algorithm are presented in the following section.

3.3 Gibbs sampler for the Hierarchical Bayesian model.

We start from the likelihood function of the observed spectra given our model

$$\begin{aligned}
\mathrm{p}\left(\mathbf{R}^q|\sigma\right) &\propto \sigma_D^{-\frac{\mathrm{M}}{2}} \exp\left(-\frac{1}{2\sigma^{-2}}\mathrm{tr}\left(\left(\mathbf{R}^q\right)^T\mathbf{R}^q\right)\right) \\
\prod_k \mathrm{p}\left(\mathbf{Q}_k|\sigma\right) &\propto \sigma_D^{-\frac{\mathrm{M+K}}{2}} \exp\left(-\frac{1}{2\sigma^{-2}}\sum_k \mathrm{tr}\left(\left(\mathbf{R}^q\right)^T\mathbf{R}^q\right)\right) \\
\mathbf{R}^q &= \left(\dot{\mathbf{X}}^q - \left[\begin{array}{cc}\mathbf{X}^q & \mathbf{U}^q\end{array}\right]\left[\begin{array}{c}\mathbf{A}^{\mathrm{T}} \\ \mathbf{C}^{\mathrm{T}}\end{array}\right]\right).
\end{aligned}$$

We want to derive the conditional distribution of the parameters $\left[\begin{array}{c}\mathbf{A}^{\mathrm{T}} \\ \mathbf{C}^{\mathrm{T}}\end{array}\right]$, we define the matrix $\boldsymbol{\Theta} = \left[\begin{array}{c}\mathbf{A}^{\mathrm{T}} \\ \mathbf{C}^{\mathrm{T}}\end{array}\right]$ and the design matrix.Then we have the likelihood function over all experiments is

$$\mathrm{p}\left(\{\mathbf{R}^q\}|\left\{\dot{\mathbf{X}}^q\right\},\{\boldsymbol{\Phi}^q\},\boldsymbol{\Theta},\sigma_D\right) \propto \prod_q \mathrm{p}\left(\mathbf{R}^q|\dot{\mathbf{X}}^q,\boldsymbol{\Phi}^q,\boldsymbol{\Theta},\sigma_D\right)\mathrm{p}\left(\boldsymbol{\Theta}|\mathbf{H},\boldsymbol{\tau}\right).$$

The expression can be factored in the following manner

$$\begin{aligned}
\mathrm{p}\left(\{\mathbf{R}^q\}|\boldsymbol{.}\right) &\propto \prod_{\mathrm{q}} \exp\left(-\frac{1}{2\sigma_D^2}\mathrm{Tr}\left(-2\left(\left(\boldsymbol{\Phi}^q\right)^T\dot{\mathbf{X}}^q\right)^{\mathrm{T}}\boldsymbol{\Theta} + \boldsymbol{\Theta}^{\mathrm{T}}\left(\left(\boldsymbol{\Phi}^q\right)^T\boldsymbol{\Phi}^q\right)\boldsymbol{\Theta}\right)\right) \\
&\propto \exp\left(-\frac{1}{2\sigma_D^2}\mathrm{Tr}\left(-2\boldsymbol{\eta}^{\mathrm{T}}\boldsymbol{\Theta} + \boldsymbol{\Theta}^{\mathrm{T}}\boldsymbol{\Psi}\boldsymbol{\Theta}\right)\right) \\
\boldsymbol{\eta}^{\mathrm{T}} &= \left(\sum_{\mathrm{q}}\left(\boldsymbol{\Phi}^q\right)^T\dot{\mathbf{X}}_k\right) \\
\boldsymbol{\Psi} &= \left(\sum_q\left(\boldsymbol{\Phi}^q\right)^T\boldsymbol{\Phi}^q\right).
\end{aligned}$$

Now by using the vectorization transformation for an arbitrary matrix $\mathbf{M}$, such that $\bar{\mathbf{M}} = \mathrm{vec}(\mathbf{M})$ and properties

$$\begin{aligned} \mathrm{Tr}\left(\mathbf{M}^{\mathrm{T}}\mathbf{N}\right) &= \bar{\mathbf{M}}^{\mathrm{T}}\bar{\mathbf{N}} \\ \mathrm{vec}(\mathbf{MN}) &= (\mathbf{I}\otimes\mathbf{M})\,\bar{\mathbf{N}} \end{aligned}$$

we will complete the square in order to derive the conditional distribution of the spectra and its derivative given the parameters

$$\begin{aligned} \mathrm{p}\left(\left\{\dot{\mathbf{X}}^q\right\},\{\boldsymbol{\Phi}^q\}|\boldsymbol{\Theta},\sigma_D\right) &\propto \exp\left(-\frac{1}{2\sigma_D^2}\left(-2\mathrm{Tr}\left(\boldsymbol{\eta}^{\mathrm{T}}\boldsymbol{\Theta}\right)+\mathrm{Tr}\left(\boldsymbol{\Theta}^{\mathrm{T}}\boldsymbol{\Psi}\boldsymbol{\Theta}\right)\right)\right) \\ &\propto \exp\left(-\frac{1}{2\sigma_D^2}\left(-2\bar{\boldsymbol{\eta}}^{\mathrm{T}}\bar{\boldsymbol{\Theta}}+\bar{\boldsymbol{\Theta}}^{\mathrm{T}}\mathrm{vec}\left(\boldsymbol{\Psi}\boldsymbol{\Theta}\right)\right)\right) \\ &\propto \exp\left(-\frac{1}{2\sigma_D^2}\left(-2\bar{\boldsymbol{\eta}}^{\mathrm{T}}\bar{\boldsymbol{\Theta}}+\bar{\boldsymbol{\Theta}}^{\mathrm{T}}\left(\mathbf{I}\otimes\boldsymbol{\Psi}\right)\bar{\boldsymbol{\Theta}}\right)\right). \end{aligned} \tag{3.7}$$

The spike and slab prior over the parameters, that is, the LTI coefficients are normally distributed with variance $h_{ij}\tau_{ij}^2$, the term h_{ij}can take a value of 1 or close to zero. If the value of h_{ij} is close to zero then the values drawn from $\mathrm{p}(\boldsymbol{\Theta}|\mathbf{H},\boldsymbol{\tau})$ follow a very narrow distribution with mean zero. If the value of h_{ij} equals 1, then the slab term τ_{ij}^2 dominates allowing draws from a "wider" distribution,

$$\mathrm{p}\left(\theta_{ij}|h_{ij},\tau\right) \sim \mathcal{N}\left(0,h_{ij}\tau_{ij}^2\right). \tag{3.8}$$

Then by using vectorization we write the prior in canonical form as a diagonal matrix whose entries are given by the spike and slab prior coefficients $h_{ij}\tau_{ij}^2$, thus each entry in this matrix has a bimodal distribution

$$\begin{aligned} \mathrm{p}\left(\boldsymbol{\Theta}|\mathbf{H},\boldsymbol{\tau}\right) &\propto \exp\left(-\frac{1}{2}\bar{\boldsymbol{\Theta}}^{\mathrm{T}}\boldsymbol{\Gamma}\bar{\boldsymbol{\Theta}}\right) \\ \boldsymbol{\Gamma} &= \mathrm{diag}\left(h_{ij}\tau_{ij}^2\right). \end{aligned}$$

Finally multiplying eq.(3.7) by the Gaussian with hypervariance given by the spike and slab prior of eq.(3.8) and completing the square we get the analytical expression for the posterior distribution over the parameters $\boldsymbol{\Theta}$

$$\mathrm{p}\left(\left\{\dot{\mathbf{X}}^q\right\},\{\boldsymbol{\Phi}^q\}|\boldsymbol{\Theta},\sigma_D\right)\mathrm{p}\left(\boldsymbol{\Theta}|\mathbf{H},\boldsymbol{\tau}\right) \propto$$

$$\exp\left(-\frac{1}{2\sigma^2}\left(-2\bar{\boldsymbol{\eta}}^{\mathrm{T}}\bar{\boldsymbol{\Theta}}+\bar{\boldsymbol{\Theta}}^{\mathrm{T}}\left(\mathbf{I}\otimes\boldsymbol{\Psi}\right)\bar{\boldsymbol{\Theta}}+\bar{\boldsymbol{\Theta}}^{\mathrm{T}}\sigma_D^2\boldsymbol{\Gamma}\bar{\boldsymbol{\Theta}}\right)\right)$$

thus

$$\begin{aligned}\mathrm{p}\left(\boldsymbol{\Theta}|\left\{\dot{\mathbf{X}}^q\right\},\{\boldsymbol{\Phi}^q\},\mathbf{H},\boldsymbol{\tau},\sigma\right) &\sim \mathcal{N}\left(\bar{\boldsymbol{\mu}},\sigma^2\boldsymbol{\Sigma}\right) &(3.9)\\ \bar{\boldsymbol{\mu}} &= \boldsymbol{\Sigma}^{-1}\bar{\boldsymbol{\eta}}^{\mathrm{T}} &(3.10)\\ \boldsymbol{\Sigma}^{-1} &= \mathbf{I}\otimes\boldsymbol{\Psi}+\sigma_D^2\boldsymbol{\Gamma}. &(3.11)\end{aligned}$$

The h_{ij} coefficients are then Bernoulli distributed with parameter w, this parameter can be set individually for each h_{ij}.We chose to set a unique parameter for the whole network in order to reduce complexity and improve convergence time. This can also be justified if we think of it as a parameter controlling the overall complexity of the network, as it encodes the number of connections allowed

$$\mathrm{p}\left(h_{ij}|w\right)\sim\left(1-w\right)\delta_{v0}+w\delta_1$$

The conditional distribution of the complexity parameter w given the elements of $\mathbf{H}$ is Beta distributed due to conjugacy and it is a function of the number of one and zeros in $\mathbf{H}$

$$\mathrm{p}\left(w|\mathbf{H}\right)\sim\mathrm{Beta}\left(c_1+\#\{h_{ij}=1\},c_2+\#\{h_{ij}=v_0\}\right). \tag{3.12}$$

The conditional distribution of τ^{-2} given the θ and h parameters is Gamma distributed due to conjugacy

$$\mathrm{p}\left(\tau_{ij}^{-2}|a,\theta_{ij},h_{ij}\right) \sim \mathrm{Gamma}\left(a_1+\frac{1}{2},a_2+\frac{\theta_{ij}^2}{2h_{ij}}\right). \tag{3.13}$$

The conditional distribution of σ_D^{-2} given the residuals is Gamma distributed due to conjugacy and is a function of the number of coefficients, the number of experiments and residues

$$\mathrm{p}\left(\sigma_D^{-2}|\left\{\mathbf{R}^q\right\},b\right)\sim\mathrm{Gamma}\left(b_1+\frac{\mathrm{MQ}}{2},b_2+\frac{\sum_q\mathrm{tr}\left(\left(\mathbf{R}^q\right)^T\mathbf{R}^q\right)}{2}\right). \tag{3.14}$$

The likelihood of the similarity scores is a normal distribution around the additive clustering formulation given the adjacency matrix $\mathbf{H}$ and the induced similarity coefficients β

$$\mathrm{p}\left(s_{\mathrm{ij}}|\mathbf{H},\boldsymbol{\beta},\sigma_{seq}\right) \propto \sigma_{seq}^{-1/2}\exp\left(-\frac{1}{2\sigma_{seq}^{2}}\left(s_{ij}-\sum_{l}^{\mathrm{N}}h_{il}h_{jl}\beta_{l}\right)^{2}\right). \quad (3.15)$$

having vector$\bar{\mathbf{S}}=[s_{ij}]_{i<j}$ representing the off-diagonal elements of the upper triangular matrix of $\mathbf{S}$, vector$\bar{\mathbf{h}}_i$ representing the $i-th$ row vector of $\mathbf{H}$ and $\circ$ representing the element wise product (Hadamard product). Then the distribution of the upper diagonal elements is:

$$\mathrm{p}\left(\bar{\mathbf{S}}|\mathbf{H},\boldsymbol{\beta},\sigma_{seq}\right) \propto$$

$$\sigma_{seq}^{-\mathrm{D}/2}\exp\left(-\frac{1}{2\sigma_{seq}^{2}}\left(\bar{\mathbf{S}}-\left[\bar{\mathbf{h}}_i\circ\bar{\mathbf{h}}_j\right]_{i<j}\boldsymbol{\beta}\right)^{\mathrm{T}}\left(\bar{\mathbf{S}}-\left[\bar{\mathbf{h}}_i\circ\bar{\mathbf{h}}_j\right]_{i<j}\boldsymbol{\beta}\right)\right).$$

We use the latter notation to express the conditional distribution for σ_{seq}^{2} that is Gamma distributed due to conjugacy

$$\mathrm{p}\left(\sigma_{seq}^{-2}|\bar{\mathbf{S}},\boldsymbol{\beta}\right) \sim \mathrm{p}\left(\bar{\mathbf{S}}|\mathbf{H},\boldsymbol{\beta},\sigma_{seq}^{-2}\right)\mathrm{p}\left(\sigma_{seq}^{-2}\right) \quad (3.16)$$

$$\sim \mathrm{Gamma}\left(a_1+\frac{\mathrm{D}^2}{2},a2+\frac{\left(\bar{\mathbf{S}}-\left[\bar{\mathbf{h}}_i\circ\bar{\mathbf{h}}_j\right]_{i<j}\boldsymbol{\beta}\right)^{\mathrm{T}}\left(\bar{\mathbf{S}}-\left[\bar{\mathbf{h}}_i\circ\bar{\mathbf{h}}_j\right]_{i<j}\boldsymbol{\beta}\right)}{2}\right).$$

Instead of sampling from $\mathrm{p}\left(\beta_l|\boldsymbol{\cdot}\right)$ we use the non-negative least square estimate of $\boldsymbol{\beta}$, by solving the quadratic form

$$\min_{\boldsymbol{\beta}}\left(\bar{\mathbf{S}}-\left[\bar{\mathbf{h}}_i\circ\bar{\mathbf{h}}_j\right]_{i<j}\boldsymbol{\beta}\right)^{\mathrm{T}}\left(\bar{\mathbf{S}}-\left[\bar{\mathbf{h}}_i\circ\bar{\mathbf{h}}_j\right]_{i<j}\boldsymbol{\beta}\right);\ s.t.\beta_l\geq 0 \quad (3.17)$$

We define matrices

$$\mathbf{H}_{ij}^{v0}=\left[\bar{\mathbf{h}}_i\circ\bar{\mathbf{h}}_j\right]_{i<j}\text{ such that }h_{ij}=v0,$$

and

$$\mathbf{H}_{ij}^{1} = \left[\bar{\mathbf{h}}_i \circ \bar{\mathbf{h}}_j\right]_{i<j} \text{ such that } h_{ij} = 1.$$

The conditional over the h_{ij} parameters is straightforward to derive by applying the product rule; care has to be taken though, if we include the sequence part of h_{ij} stop being conditional independent, so the Gibbs sampler now has to update the h_{ij} one by one conditioned on all the previous values for $\mathbf{H}$. This increases the computation time.

$$\mathrm{p}\left(h_{ij}|h_{/ij},\boldsymbol{\beta},\sigma_{seq}^{-2},\mathbf{S},\boldsymbol{\Theta},w,\tau,v_0\right) = P\left(\mathrm{h}_{ij} \mid w\right) P\left(\theta_{ij} \mid h_{ij},\tau_{ij}^{2}\right) \mathrm{p}\left(\bar{\mathbf{S}}|h_{/ij},h_{ij},\boldsymbol{\beta},\sigma_{seq}\right)$$

$$\begin{aligned}
\mathrm{p}\left(h_{ij}|\boldsymbol{.}\right) &\sim \frac{m_{v0}}{m_{v0}+m_1}\left(1-w\right)\delta_{v0} + \frac{m_1}{m_{v0}+m_1} w\delta_1 \qquad (3.18)\\
m_{v0} &= \sigma_{seq}^{-\mathrm{D}/2} v_0^{-1/2}\\
&\times \exp\left(-\frac{1}{2}\left(\frac{\theta_{ij}^2}{v_0\tau_{ij}^2} + \frac{1}{\sigma_{seq}^2}\left(\bar{\mathbf{S}}-\mathbf{H}_{ij}^{v0}\boldsymbol{\beta}\right)^{\mathrm{T}}\left(\bar{\mathbf{S}}-\mathbf{H}_{ij}^{v0}\boldsymbol{\beta}\right)\right)\right)\\
m_1 &= \sigma_{seq}^{-\mathrm{D}/2}\\
&\times \exp\left(-\frac{1}{2}\left(\frac{\theta_{ij}^2}{\tau_{ij}^2} + \frac{1}{\sigma_{seq}^2}\left(\bar{\mathbf{S}}-\mathbf{H}_{ij}^{1}\boldsymbol{\beta}\right)^{\mathrm{T}}\left(\bar{\mathbf{S}}-\mathbf{H}_{ij}^{1}\boldsymbol{\beta}\right)\right)\right).
\end{aligned}$$

With these expressions we present the algorithm for sampling from the posterior distribution $p\left(w,\mathbf{H},\boldsymbol{\beta},\sigma_{seq}^{-2},\mathbf{S},\boldsymbol{\Theta},w,\tau,\sigma_D^{-2}|\mathbf{X}\right)$ will be given by the algorithm 3.1.

3.3.0.1 Program Inputs

Due to the fact that that the DFT computed by the FFT algorithm yields complex numbers, a real representation of the coefficients is needed. We work with the so-called Real Discrete Fourier Transform. It consists of stacking the real over the imaginary part of the first $(M-1)/2$ FFT coefficients. We denote this $M \times N$ matrix $\mathbf{X}$. Figures 3.3 and 3.4 show plots for the DFT real and imaginary parts of the *A. thaliana* data-set, with a photoperiod of 12 hours. The magnitude and phase spectra is also shown, finally the RDFT is presented at the bottom of the picture. In figure 3.5 we appreciate the light input signal at different frequencies modeled as an ON/OFF square signal. Hyperparameters a_1 and a_2 are set to 1 so w is uniformly distributed in the interval $[0,1]$. Parameters b_1 and b_2 are set so the hypervariance $h_{ij}\tau^2$ has a continuous bi-modal distribution, according to

Algorithm 3.1 Gibbs sampler for the DSS model. We condition over each parameter individually in order to draw samples from the joint posterior $p\left(w, \mathbf{H}, \boldsymbol{\beta}, \sigma_{seq}^{-2}, \mathbf{S}, \Theta, w, \tau, \sigma_D^{-2} | \mathbf{X}\right)$.

Inputs: K time series of M time-points for N gene expression levels, encoded in matrices $\{x_k\}$. Prior hyper-parameters $a_1, a_2, b_1, b_2, c_1, c_2$. Optional similarity matrix $\mathbf{S}$. Outputs: Joint conditional posterior distribution $p(\mathbf{H}, \mathbf{A}, \mathbf{C}, \beta, w, \tau, \sigma_D, \sigma_s | \{\mathbf{X_k}\})$

1. Obtain the DFT of $\{x_k\}$ and the corresponding RDFT coefficient matrices $\{\mathbf{X_k}\}$
2. Compute the derivatives $\left\{\dot{\mathbf{X}}_\mathbf{k}\right\}$
3. Sample from the conditional distribution over the LTI coefficients, given in eq. (3.9)
4. Sample from the conditional distribution overτ^{-2} given by eq. (3.13)
5. Sample $\mathbf{H}$ from eq. (3.18), to account for the decay rates we set diagonal elements h_{ii} to 1, and set the diagonal elements of matrix $\mathbf{A}$ to negative.
6. Sample w from eq. (3.12)
7. Sample σ_D from eq. (3.14)
8. OPTIONAL sample σ_{seq} from eq. (3.16) and β from the nonnegative least squares solution to equation (3.17).
9. Return to step 3

Note: A burn in period of 4000 samples is considered in the general purpose implementation of the model.

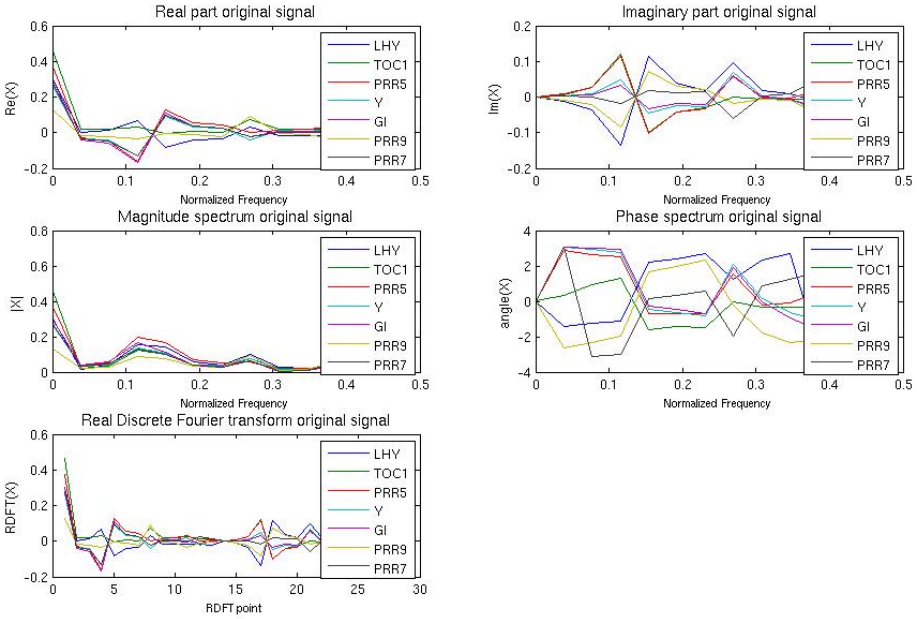


Figure 3.3: Spectra for the *A. thaliana* circadian clock simulation with a 12 hour photo-period

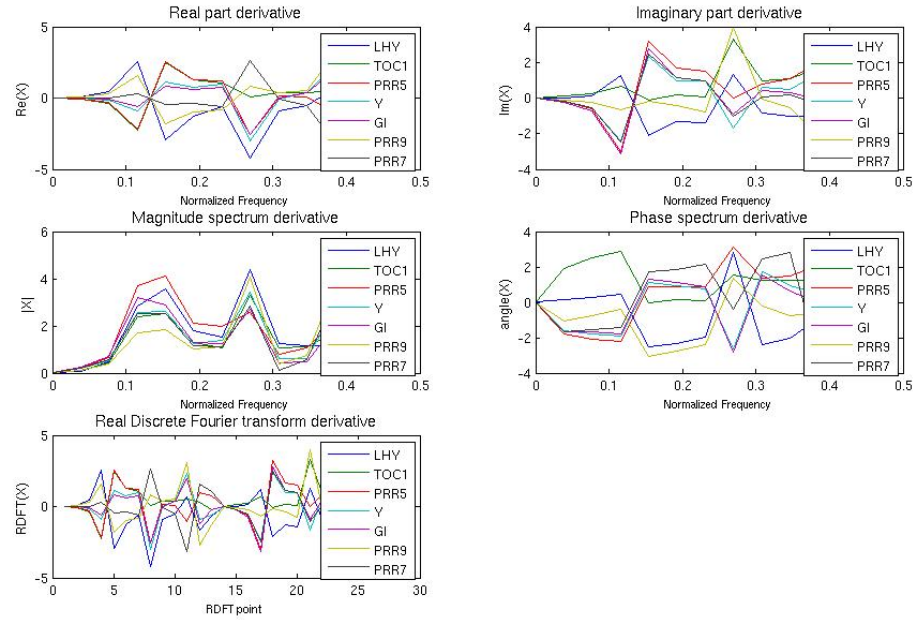


Figure 3.4: Derivative Spectra for the *A. thaliana* circadian clock simulation with a 12 hour photo-period

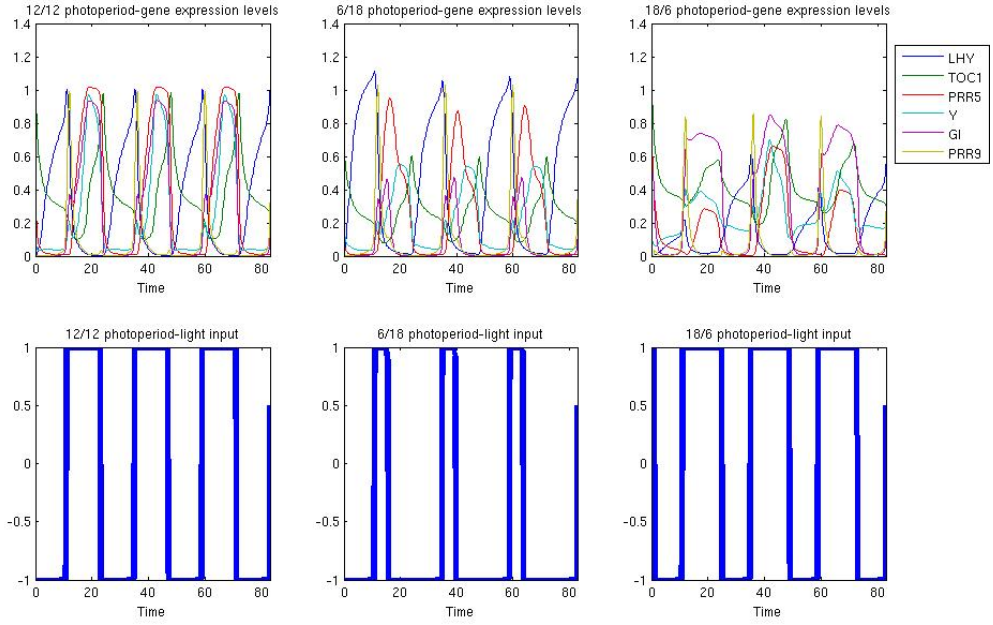


Figure 3.5: Three examples of light input for the *A. thaliana* circadian clock simulation.

the recommendations of Ishwaran and Rao (2005), in which they set them to 5 and 50 respectively. Alternative parametrization of 50 and 500 was also tested yielding better results in some cases. The hyperparameters c_1 and c_2 for σ_D^{-2} are set to 0.001 and 0.001, this is a weak prior reflecting uncertainty about the linearity of the system. Hyperparamters d_1 and d_2 are set to 10 and 0.001, this parametrization required a manual tunning, as the scale parameter σ_{seq}^{-2} having a weak prior resulted in the effects of the sequence similarity model to vanish. By modifying this prior we can give more "weight" to the sequence similarity clustering.

3.3.0.2 Output

The Gibbs sampler presented in the previous section allows us to draw samples from the joint conditional distribution $p(\mathbf{H}, \mathbf{A}, \mathbf{C}, \beta, w, \tau, \sigma_D, \sigma_s | \{\mathbf{X_k}\})$. The marginal distribution for each of the models parameters can be drawn from this joint distribution, and the expected value for each parameter equals to the average of the samples. For example, figure 3.6 illustrates the expected value for matrix $\mathbf{H}$ obtained from averaging over 1000 samples drawn from the marginal distribution $p(\mathbf{H} | \{\mathbf{X_k}\})$. This figure shows in dark red those elements with higher

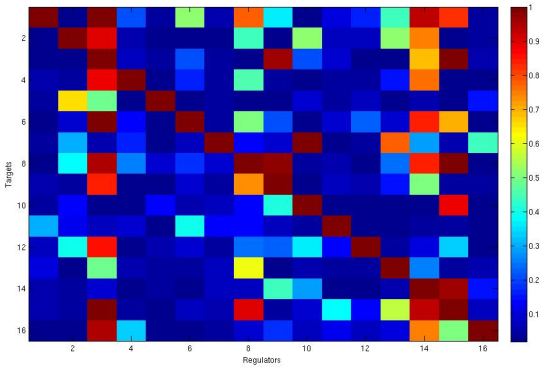


Figure 3.6: Heat-map representing the expected value for $p(\mathbf{H}|\boldsymbol{\cdot})$ obtained by averaging the last 1000 samples. Rows represent targets and columns regulators. The diagonal indicates the decay parameters λ.

probability of a regulatory interaction under the model assumptions, except the diagonal elements, which represent the decay rates of the equation model. The AUPR were computed by thresholding the off-diagonal elements of this matrix for each data-set.

3.3.1 Incorporating protein complexes into the model

The formation of protein complexes is an important post-transcriptional process that affects the behavior of the genetic regulatory network. In order incorporate these complexes in our inference scheme some requisites are needed

- We know, or at least have a candidate list of protein complexes and their components.
- We possess the gene expression levels of the components of the complex for all time series.

Under these assumptions, lets suppose we have a set of proteins $\{z_1 \dots z_m\}$ forming a protein complex z_c such that its production rate is a function of z_c concentrations, that is

$$\frac{dz_c}{dt} = f\left(\prod_{i=1}^{m} z_i\right) \tag{3.19}$$

Usually protein level measurements are unobserved, thus we are going to approximate 3.19 through the observed gene expression levels of $\{z_1 \dots z_m\}$, which we will denote as z_c^e. Then we apply our linear model to express z_c as

$$\frac{dz_c}{dt} = \alpha \prod_{i=1}^{m} z_i^e - \lambda z_c \tag{3.20}$$

Using the DFT on both sides of 3.20 we get

$$\mathbf{D}Z_c = \alpha DFT\left(\prod_{i=1}^{m} z_i^e\right) - \lambda Z_c \tag{3.21}$$

where $\mathbf{D}$ is the time domain derivative operator, thus we can approximate the DFT spectrum of z_c by solving for Z_c in Eq.(3.21) which yields

$$Z_c = \alpha \left(\mathbf{D} + \lambda \mathbf{I}\right)^{-1} DFT\left(\prod_{i=1}^{m} z_i^e\right) \tag{3.22}$$

we set parameters $\lambda = \alpha = 1$ and we proceed to plug the estimated spectrum as part of matrix $\mathbf{U}$. This in order to account for the post transcriptional nature of this component and the fact that we used very strong assumptions in order to derive it (we fixed the elements that form part of the complex) so its pointless to try to infer any regulators over it. On the other hand, this approximation may be useful for inferring which elements are being regulated by z_c.

To illustrate the results of this approximation we simulated the Circadian clock model presented in (Pokhilko et al., 2012), this clock includes a three protein complex called EC. We simulated the ODE model and down sampled it to obtain 24 samples per day for three days. We computed the RDFT spectrum of the ODE-simulated EC. Then we proceeded to simulate EC through the approximation given in Eq.(3.22). The results are presented in Fig.3.7. Here we can see that two of the biggest components are being approximated by the LTI approximation depending on the value of parameter α. The main contention is that these two components being approximated are enough for our regression scheme to distinguish the effects of EC over other members of the network.

3.4 Results

In this section we assess the performance of our method on two realistic simulated data sets and a real data set, comparing its performance to two other state of the art methods. We call our method DSS, for DFT-based Spike and Slab model. The first simulated data set was generated from a well known model for the *A. thaliana* circadian clock network (Pokhilko et al., 2010). This model is a non-linear ODE-based model which exhibits regular oscillations (for suitable parametrisations),

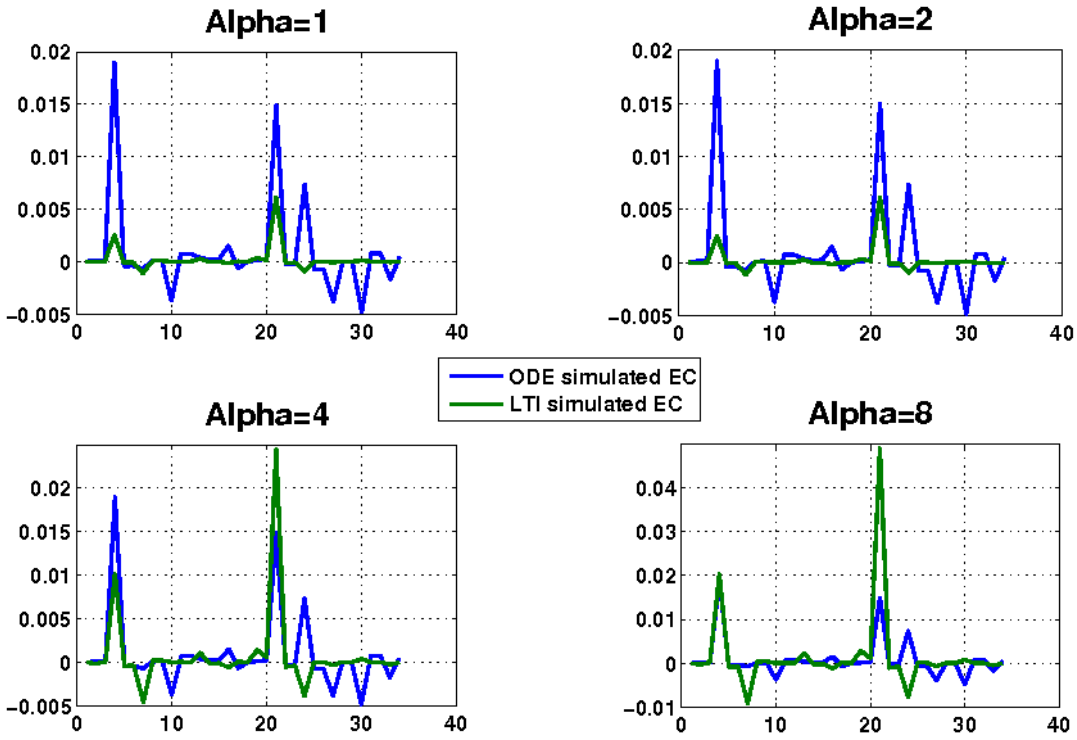


Figure 3.7: Approximation to the DFT spectra of EC. The DFT coefficients are represented in RDFT form, first half correspond to the real part and second half to the imaginary part. The LTI approximation to the simulated ODE spectra is shown in green for different values of alpha. Depending on parameter alpha the two biggest components of the spectra are reasonably approximated, as can be seen for the case of $\alpha = 4$.

thus matching one of our main modeling assumptions. However, it is a non-linear model, hence introducing an element of model mismatch. As a second synthetic benchmark data set we used one of the data sets provided by the DREAM 4 challenge (Marbach et al., 2010). This is again a non-linear model, which exhibits damped oscillatory dynamics in some of the nodes; thus, this data set presents considerably more elements of model mismatch. The last experiment tested the method on a real data set of gene expression levels obtained in a micro-array experiment for the *S. cerevisae* cell cycle transcriptional network (Orlando et al., 2008).

Results were assessed in terms of area under the Precision-Recall (AUPR) curve; PR curves plot the fraction of correctly called instances versus the ratio of true positives over true positives plus false negatives. An ideal classifier would give a AUPR of 1, while a random baseline would return the ratio of positives negatives. Inference of the models parameters was conducted by Gibbs Sampling from the model presented in (Fig.3.2) . In total, 5000 samples were obtained. The last 1000 samples were selected and averaged to compute the conditional probability of a link $\mathrm{p}(h_{ij} = 1|\cdot)$ given the model and the expression data, see Appendix Section 1.1.1 and Section 1.1.2 for details into the inputs and outputs of the program.

3.4.1 Competing methods

As a first comparison, in order to establish the validity of our claim that frequency domain analysis is beneficial for oscillatory networks, we sought to compare our results with a complete analogue in time domain. To do this, we implemented a spline-based alternative to the DFT, using cubic splines interpolation as means of computing the time domain derivative, while the rest of the hierarchical model was left unchanged. As competing methods to assess the performance of DSS we selected GENIE3 (Huynh-Thu et al., 2010), which is based on random forests, and the ODE-regression based Inferelator (Bonneau et al., 2006; Greenfield et al., 2013).

In a network of N genes, GENIE3 solves N regression problems by predicting, using random forests, the expression level of each gene as a function of the other N-1 genes (putative regulators). Then the relative importance of each gene expression is evaluated and the putative gene interactions are ranked. GENIE3

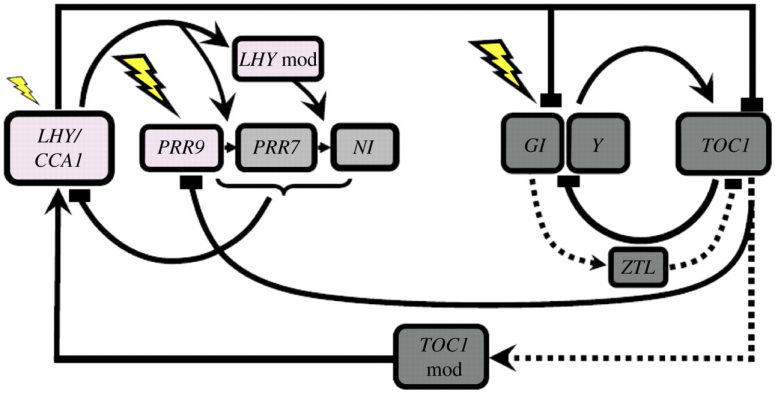


Figure 3.8: *A. thaliana* circadian clock model, transcriptional elements LHY, PRR9, PRR7, NI, Y, and 'TOC1. Post-transcriptional elements ZTL, 'TOC1mod and LHY-mod. Light input is represented by a lighting symbol. Activating interactions are represented by solid line with arrows, repression by solid line with rectangles at the end, post transcriptional interactions are represented by dashed lines. .

was designed for steady state data, but time-series adaptation can be readily derived and was provided to us by one of the authors. The Inferelator estimates the parameters of an ODE system using regression with L1-regularization over a finite element approximation of the derivative. The method has been extended (Greenfield et al., 2013), with new functionalities to incorporate prior information over the network links, and to use alternative optimisation methods for model selection, including the elastic-net (regularization over L1 and L2 norms) and Bayesian regression with best subset selection.

Finally, as a simple baselines, we implemented a L1 regularised version of the regression problem in Eq.(3.3), using the LASSO implementation (Tibshirani, 1994).

3.4.2 *A. thaliana* circadian clock

As a first example we used data generated from a well known oscillatory network model, the *A. thaliana* circadian clock. The data consists of simulated mRNA measurements from the model found in Pokhilko et al. (2010). This non-linear model has 7 transcription factors and 2 post transcriptional elements ZTL and LHYmod. In order to replicate experimental conditions, we assume that only mRNA data is available, so protein concentrations for the transcriptional and post-transcriptional elements are assumed unobserved. The transcription factors

used for network inference are 'LHY ', 'TOC1 ', 'PRR5 ', hypothetical gene 'Y ', 'GI ', 'PRR9' and 'PRR7', the post-transcriptional elements are not considered. A graphical representation of the model can be observed in Fig.3.8. This model was simulated for 3 cycles obtaining 28 samples. The procedure was performed with a light/dark photo period of 12/12, 6/18, 8/16, 18/6 and 20/4 hours which are represented in our model by binary input signals $\mathbf{U}$. This design of our study is created to mimic a realistic experimental setting as in Edwards et al. (2010); the biological rationale for such design is that stimulating the system with these different inputs may tease out the contribution of the main drivers of the clock at different times of day. We also simulated knock-out mutants ΔTOC1, ΔPRR7PRR9, ΔLHY and ΔGI by the same procedure as presented in Pokhilko et al. (2010) with photo periods of 12/12 hours. These experiments amount to 14 time series; as these data are directly the outputs of an ODE model (without any additional noise) we define this idealised data set as the *noiseless* data set. To assess statistically the performance and robustness of our method, we generated additional noisy datasets by adding Gaussian white noise with a Signal-to-noise (SNR) of 50 (low noise regime, as could be found in e.g. luciferase reporter time series) and 10 (high noise regime, similar to a noisy microarray time series). For each noise level, we generated 100 independent data sets. An example of the simulated expression levels is plotted in the upper left panel of Fig 3.11. Using the model specification as ground truth, we proceeded to draw the PR-curves for the different methods and computed the area under the PR-curve for all the resulting networks. In Fig.**??** we appreciate an example of the sampler output as a heatmap of the posterior probability over matrix $\mathbf{H}$ and in Fig.**??** an example of the PR curves from which the area under the curve is computed. These areas are plotted for the noiseless (simulated data without added noise) and noisy data in the upper right panel of Fig.3.11. To test the effect of including side information, we simulated a between-gene similarity matrix by drawing β_l from a uniform distribution $\mathrm{U}(0.1, 0.6)$ and using Eq.(3.6). In this case we notice an important improvement by observing an increment in the AUPR to 0.68 in the noiseless case, 0.63 ± 0.07 at 50SNR and 0.59 ± 0.12 at 10SNR (both statistically significant at $p < 1e-4$ when compared to results without side information). The difference between the spline solution with side information and the DSS solution with side information was not statistically significant in our experiments at different noise levels. The principal objective of using this simple

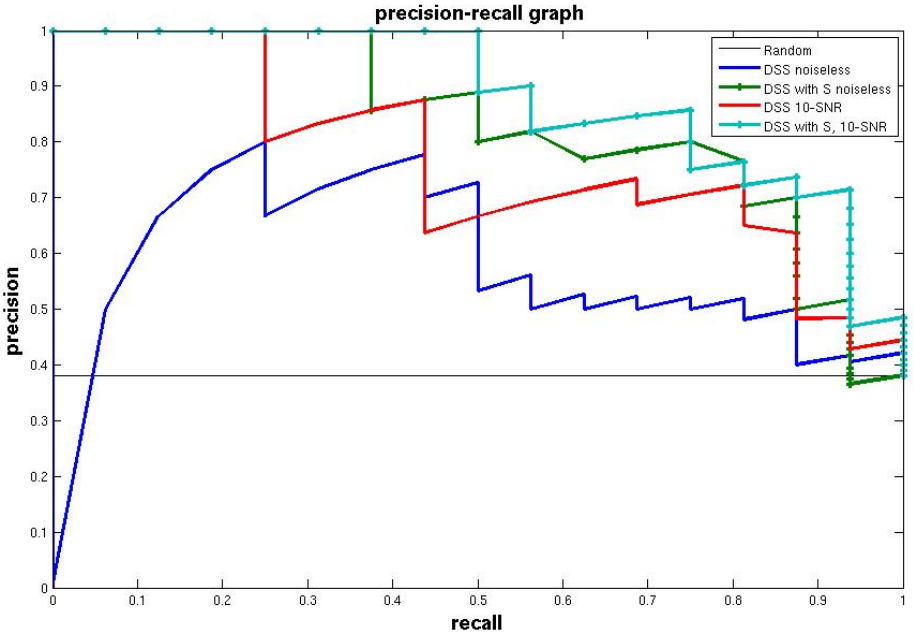


Figure 3.9: PR curves for the *A. thaliana* circadian clock network

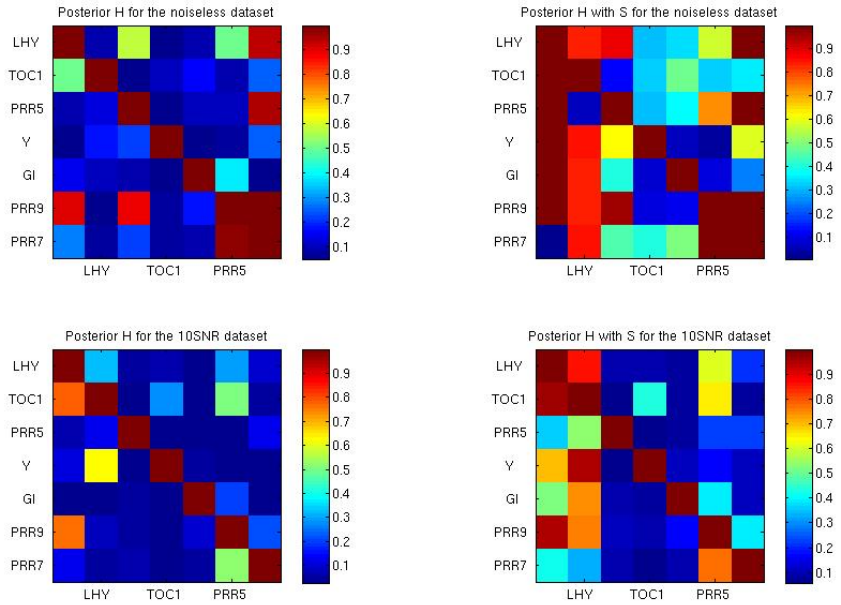


Figure 3.10: Heatmaps representing the posterior probability for the *A. thaliana* circadian clock network

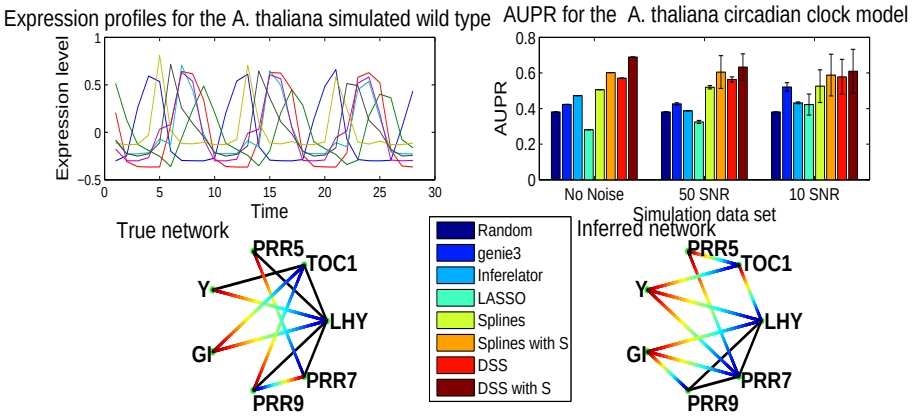


Figure 3.11: Top left are the simulated gene expression profiles for the wild type data set with SNR 100. Top right are the AUPR values for the 2 different noise levels. Bottom left is the true network topology, going from blue (regulators) to red (targets). Bottom right is the inferred network topology obtained by setting a threshold of 0.5 over the inferred matrix $\mathbf{H}$ (average over the 100 repetitions at 10SNR)

simulated similarity matrix was to confirm that structural information can be retrieved and used as aid for inference. By clustering the co-regulated elements we added additional structural constraints into the inference scheme. Finally we included a graphical representation of the true network (Fig.3.11 bottom left) and a network resulting from averaging over all inferred networks at 10SNR and setting a threshold of 0.5 over the inferred matrix $\mathbf{H}$ (Fig.3.11 bottom right). We notice that the 0.5 threshold, while reasonable, is still arbitrary and is used here only for the purposes of graphical visualisation. The full output from the method is a probability over the existence of edges, and can be better visualised as a heatmap, see Appendix Section 1.2 and Section 2. Directed edges go from blue (regulators) to red (targets), black edges mean bidirectional regulation. As can be appreciated important features such as the bidirectional regulation between 'LHY'-'PRR7' and 'LHY'-'PRR9' are recovered. Errors are related to the roles of 'PRR7' and 'PRR9' regulating 'GI' instead of 'TOC1'. This may be due to the method confounding the effects of 'TOC1' over these former elements as being closer to the expression patterns of GI. This difficulty discriminating between the roles of the 'PRR' genes is also expressed by inferring the spurious bidirectional edge between 'PRR7'-'PRR9'.

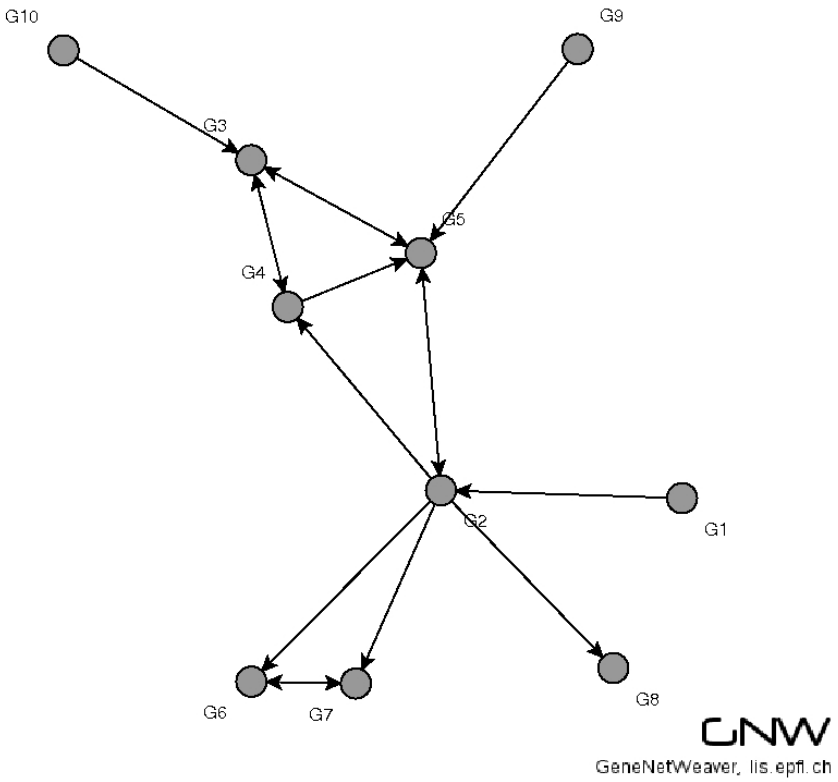


Figure 3.12: DREAM4 challenge network with 10 nodes, of those 3 are inputs, node G9 was subjected to a perturbation for half the time points.

3.4.3 DREAM Challenge

As a second example, we considered a data set from the fourth edition of the DREAM competition (Marbach et al., 2010).This data set is obtained from simulating a 10-node network, of which three nodes are input nodes; 15 regulatory links are present. Three simulations were present, one with an ODE-based system, another one with a Stochastic Differential Equation (SDE) system and a third one with SDE-based system and added experimental noise. Five time series are provided for each system, a time series contains 21 samples. The network is subjected to a single node perturbation, which mathematically corresponds to a change in the basal expression parameter, so the mean expression level of the node changes for half of the time points. The expression profiles for the set of ten genes in one time series is presented in (Fig.3.15 top left). This data set does not comply with the main assumption of the model (it shows irregular damped oscillations); we therefore expect performance not to be optimal, but it is still useful to evaluate comparatively the model under such a model mismatch sce-

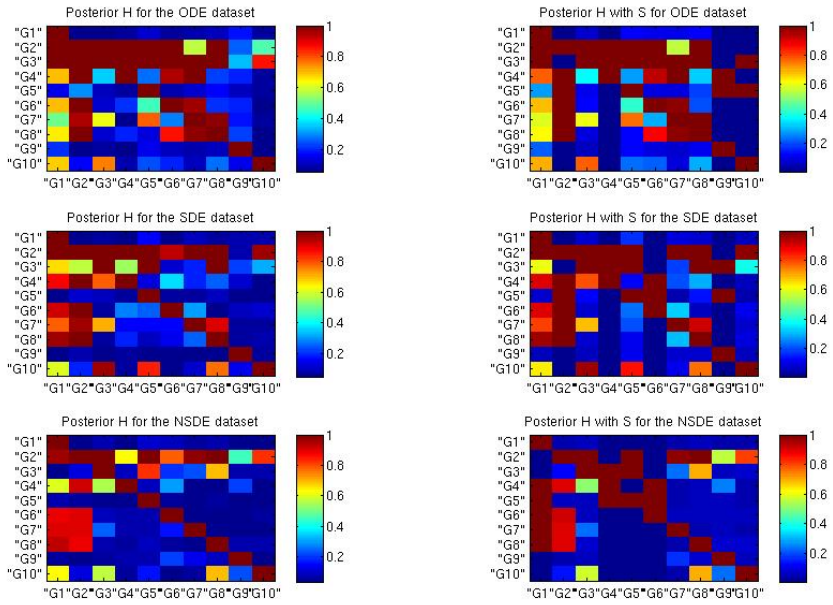


Figure 3.13: Heatmaps representing the posterior probability for the Dream4 Challenge network

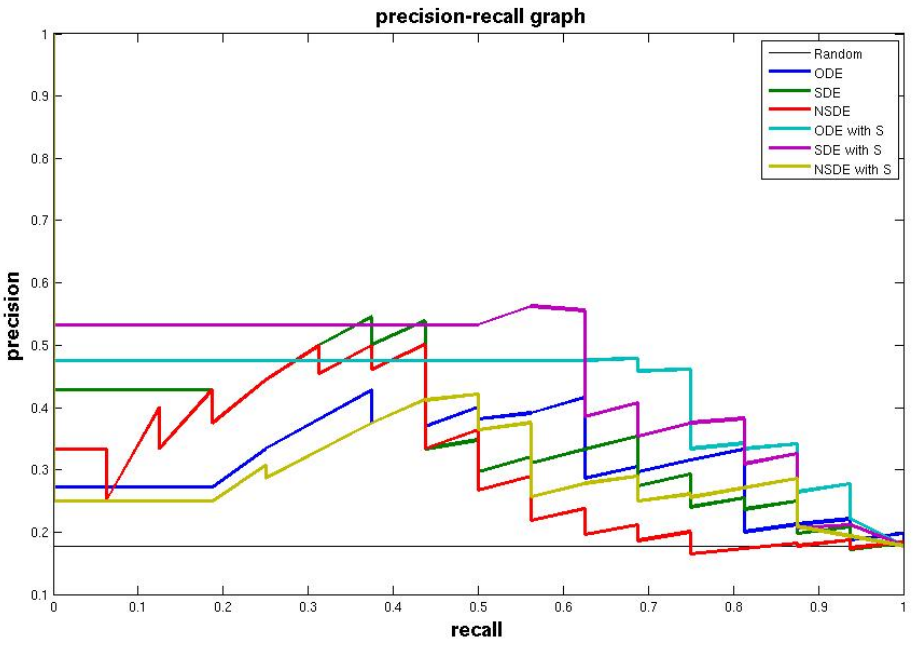


Figure 3.14: PR curves for the DREAM4 challenge network

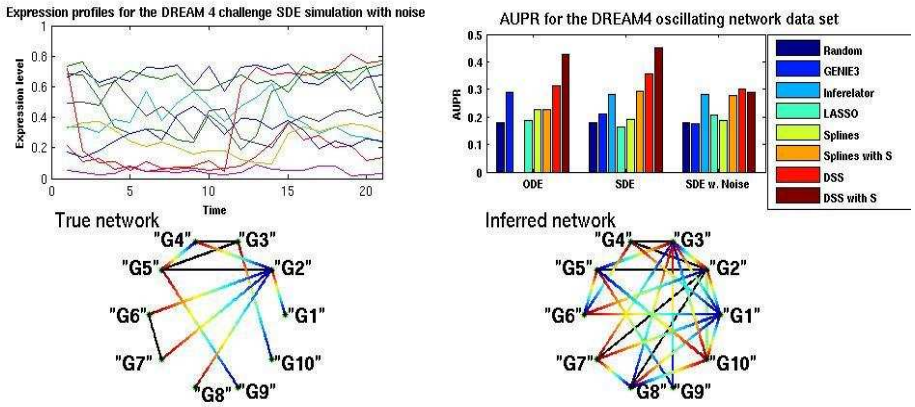


Figure 3.15: Top left is the expression profiles for the SDE model with experimental noise, node "G9" in red presents a perturbation over half the time points. Top right are the AUPR values for the three simulation models. Bottom left is the true network topology, from blue (regulators) to red (targets). Bottom right is the inferred network obtained by setting a threshold of 0.5 over the inferred matrix $\mathbf{H}$

nario.The 10-node oscillatory network that was part of the DREAM4 challenge supplementary information data set is presented in Fig.3.12 along with examples of their respective posterior distributions over matrix $\mathbf{H}$ in Fig.3.13 and pr curves in Fig.3.14. Fig.(Fig.3.15) shows a comparison of the area under the P-curve for the three simulated systems. Of these, DSS achieves better performance in the ODE-based simulation, by having an AUPR of 0.31, higher than the nearest best method (GENIE3). Inferelator could not be executed on this data set due to numerical issues (some expression levels are exactly zero in this example). The performance improved for the SDE based simulation, by achieving an AUPR of 0.35, above inferelator's 0.27. Slightly worse results were achieved for the SDE model with experimental noise, achieving an AUPR of 0.3. By simulating a sequence similarity matrix performance was improved for both ODE and SDE solutions. In the case of SDE the solution improved dramatically to 0.42.

As in the previous experiment, the network and its inferred counterpart are presented in Fig 3.15 bottom left and bottom right respectively. The inferred network is obtained by setting the threshold to 0.5 over the inferred adjacency matrix for the SDE data with added similarity matrix. As can be observed in the true network, nodes "G1" and "G10" are constant inputs. Node "G9" is subjected to perturbation for half the time points, thus its effect is propagated through the network by node "G5".

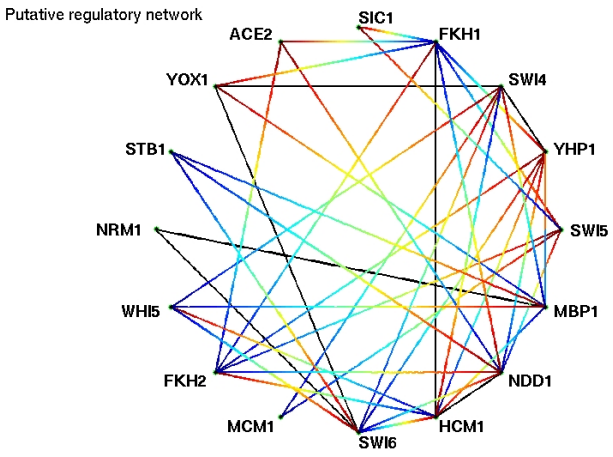


Figure 3.16: Putative yeast regulatory network edges go from blue (regulators) to red (targets)..

In the inferred network we can observe some interesting characteristics. First, nodes "G1" and "G9" are identified as input nodes, node "G10" is incorrectly identified as an output only node. Node "G2" maintains its out-degree of 4 even though it's regulators are not correctly identified. Nodes "G9" and "G5" are shown with increased in and out-degree, this may also be due to the confounding effects of the their "parent-son" relationship, specially considering that the perturbed "G9" node has the biggest amplitude of the gene expression profiles, as appreciated by the red curve in the top left plot in Fig.3.15.

3.4.4 *S. cerevisae* cell cycle

For the last experiment we used a real time series data set collected during the *S. cerevisiae* cell cycle. Our evaluation is based on the genes identified by (Haase and Wittenberg, 2014; Orlando et al., 2008) and some of their interactions on the dynamical model found in Chen et al. (2004). The main transcriptional elements selected were 'SWI5', 'YHP1', 'SWI4', 'FKH1', 'SIC1', 'ACE2', 'YOX1', 'STB1', 'NRM1', 'WHI5', 'FKH2', 'MCM1', 'SWI6', 'HCM1', 'NDD1' and 'MBP1'. The putative regulatory network used as ground truth is shown in Fig.3.16 and was built based on these references.

- Regulation of SIC1 by SWI5 as in Weiss (2012).
- Regulation of SWI4 by YHP1 as in Bähler (2005).

- Regulation of YHP1, SWI4, YOX1 and HCM1 by SWI4 as in MacIsaac et al. (2006).
- Regulation of SWI5 and ACE2 (Haase and Wittenberg, 2014); YHP1 (MacIsaac et al., 2006); SIC1, YOX1 and HCM1 (Venters et al., 2011); NDD1 (Ostrow et al., 2014); by FKH1.
- Regulation of SWI6 and MBP1 by NRM1 as in MacIsaac et al. (2006).
- Regulation of SWI6, SWI4 and MBP1 by WHI5 as in Haase and Wittenberg (2014).
- Regulation of SWI5, YHP1 and FKH1 (MacIsaac et al., 2006); ACE2 and NDD1 (Haase and Wittenberg, 2014) by FKH2.
- Regulation of SWI4 and SWI5 by MCM1 as in MacIsaac et al. (2006).
- Regulation of SWI4, FKH1, YOX1, NRM1, HCM1 and NDD1 as in MacIsaac et al. (2006).
- Regulation of YHP1, FKH1, FKH2, WHI5 and NDD1 by HCM1 as in Pramila (2006).
- Regulation of SWI5 and ACE2 (Haase and Wittenberg, 2014); YHP1 and HCM1 as in MacIsaac et al. (2006).
- Regulation of YHP1, FKH1, YOX1, NRM1 and HCM1 by MBP1 as in MacIsaac et al. (2006).
- Regulation of NDD1 (MacIsaac et al., 2006); SWI6 and MBP1 by the interaction of transcription factor MBF with STB1 (Stillman, 2013).
- Regulation of SWI4 and SWI6 by its cobinding with MCM1p as in Haase and Wittenberg (2014).

The source for the gene expression data is (Orlando et al., 2008), it contains 2 wild type replicates and two mutant replicates ($\Delta clb1,2,3,4,5,6$) each one containing 14 samples for each gene during approximately 2 cell cycles. Additionally, we downloaded promoter sequence information from (Zhu and Zhang, 1999) for all the network elements. We then proceeded to use the multiple alignment software Clustal Omega (Sievers et al., 2011) to obtain an alignment-based similarity

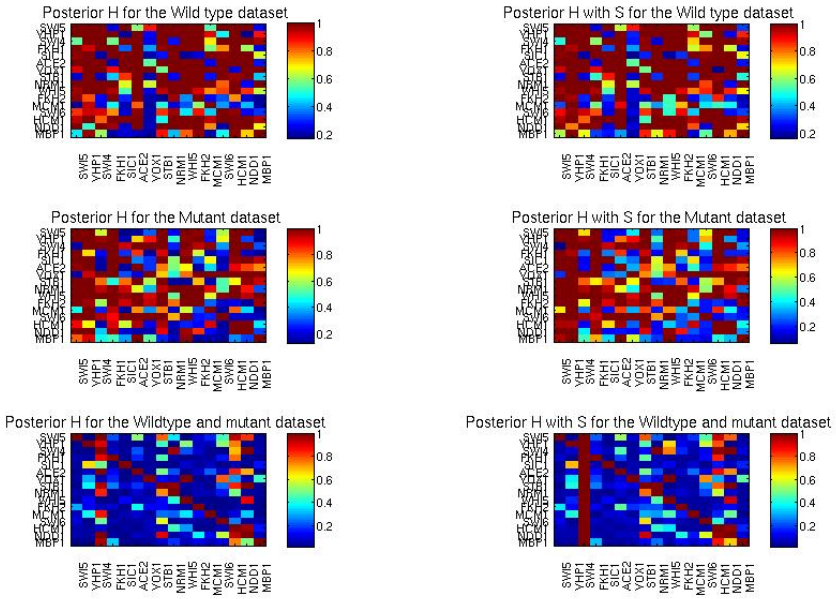


Figure 3.17: Heatmaps representing the posterior probability for the *s. cerevisiae* network

matrix **S** between sequences. As an alternative way of encoding sequence information, an alignment-free similarity matrix **S2** was built using the method described in Sims et al. (2009).

We tested three subsets of data, one containing only the wild type expression profiles, other containing only the mutants expression levels, the last data set was the normalized concatenation of both. As an example of the observed gene expression levels, Fig.3.20 top panel shows the gene expression levels for the wild type conditions.The posterior distribution over matrix **H** is presented in heatmap form in Fig.3.17. The AUPR from applying the various methods to this data are presented in Fig.3.20 bottom left panel. In this case DSS identifies the putative network well above the random baseline of 2.1 and above the competing methods. In the case of wildtypes the AUPR of DSS was of 0.24. In the mutant data sets the performance of DSS improves by including sequence similarity achieving an AUPR of 0.2607 and 0.2608 for S and S2 respectively. The best overall performance was achieved by using the combined data set with sequence similarity matrix S2, resulting in an AUPR of 0.267. The PR curves are shown in Fig.3.18.

The network in (Fig.3.20) bottom right is obtained by setting the threshold of 0.9 to the inferred network from the combined wild type and mutant dataset

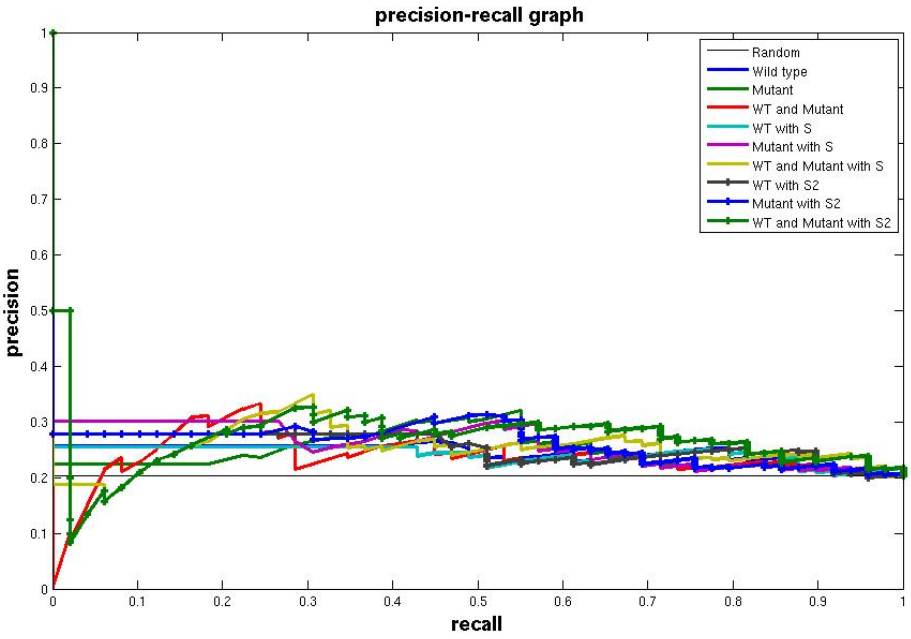


Figure 3.18: PR curves for the *s. cerevisiae* network

with added similarity matrix. In this case FKH1 has a central role in the inferred network, being fully connected to the other elements. Even though this fully connected position is biologically implausible, it does reinforce the important role of FKH1 in the cell cycle, e.g. its role in regulating the M-phase response (Kumar et al., 2000). Another noticeable inferred link concerns the post transcriptional regulation of SWI6 by WHI5p (Turner et al., 2012); this regulation was also considered as part of the ground truth network, as in the case of the yeast cell cycle transcriptional and post transcriptional regulations are intertwined (Haase and Wittenberg, 2014). Also worth noticing the regulation of SWI6 by YOX1 (member of the SBF complex) even though evidence suggests causality may be in the opposite direction (Venters et al., 2011). SWI4 and SWI6 form part of transcription factor complexes SBF and MBF, as such, their regulations may be confounded. This can be appreciated in the regulation of NRM1 by SWI4 in the inferred network, when in fact NRM1 appears to be regulated by SWI6 (DeJesus and Ioerger, 2013). The transcriptional activator NDD1 is essential during the S-phase (Loy et al., 1999), NDD1p along MCM1p bind to FKH2p (Haase and Wittenberg, 2014), this effect may be observable in the inferred network by directed edges from NDD1 to YOX1 and from YOX1 to FKH2.

By observing the AUPR plot we see that mutant data appears to be more

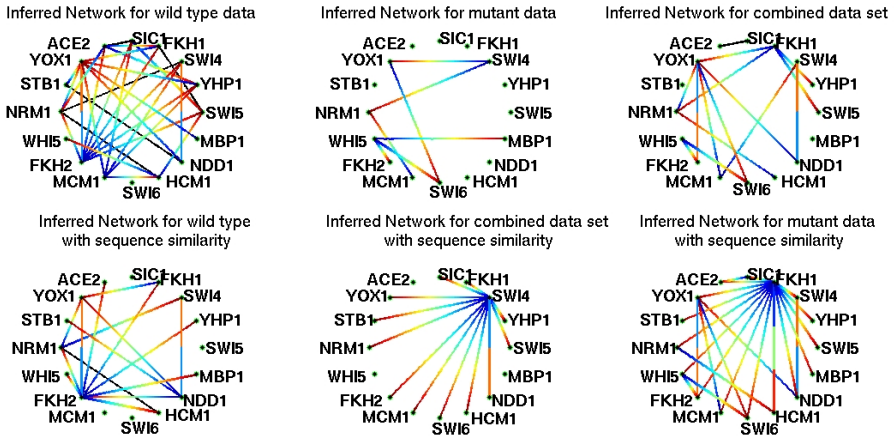


Figure 3.19: Inferred yeast networks for different data subsets with and without sequence information, edges go from blue (regulators) to red (targets)..

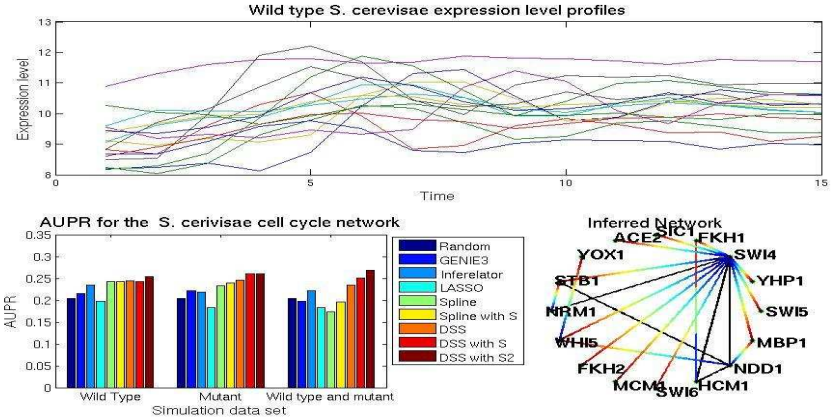


Figure 3.20: Top wild type yeast expression profiles for the selected genes, bottom left AUPR for the three different data combinations, wild type, mutants, and both. Bottom right network obtained by setting a threshold of 0.9 over matrix $\mathbf{H}$

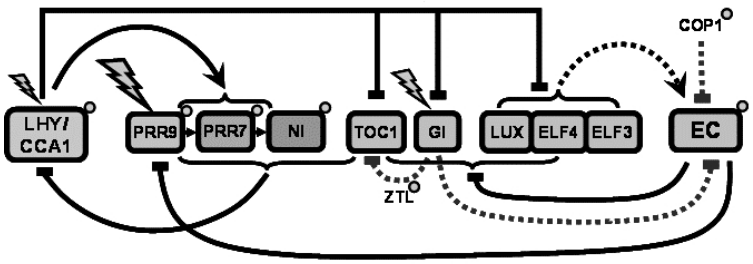


Figure 3.21: Arabidopsis thaliana circadian clock model as shown in Pokhilko et al. (2012). Here the evening complex EC regulates the expression of the PRR genes and other evening loop elements such as 'TOC1 and GI. At the same time it is post transcriptionally regulated by GI and COP1.

informative in this case than wild type, being only marginally inferior to the combined data set with similarity matrix. Part of the experimental design in selecting mutations in Orlando et al. (2008) was aiming at attenuating the effects of the post-transcriptional elements of the cell cycle; the stronger performance of our method on the mutant data sets may be explained by this experimental design. Generally, the DSS solution will find the most relevant edges in the network to explain the observed dynamics, while the DSS with similarity method will find the most relevant solution that includes a grouping of the proposed edges according to the similarity matrix. So both results can be analysed separately and may offer additional insight over the whole network behavior. With this purpose the six inferred networks and the putative ground truth are included in In Fig.3.19. The inferred networks after thresholding the value of $p(h_{ij} = 1)$ are presented, the putative ground truth matrix is presented on Fig.3.16

3.4.5 Circadian clock model with EC complex

As a final test we wished to execute DSS for a model with a protein complex. For this, we chose the one presented in Pokhilko et al. (2012) (shown in appendix Fig.). This model for the *A. thaliana* circadian clock involves the EC protein complex.

According to the model, the protein complex EC is a repressor formed by ELF3, ELF4 and LUX that is in charge of repressing the expression of the morning loop component PRR9. EC then is an important component of the the evening loop, taking the place of hypothetical component Y in the previous 2010 model

used in subsection .

Before applying DSS for structure learning on this model, we need to account for the effects of the EC complex under the model assumptions. We are going to explicitly employ the prior information we possess over the formation of this complex. In the model, these equations explain the formation of EC:

$$\begin{aligned} \frac{d}{dt}c_{\mathrm{EC}} &= p_{26}c_{LUX}c_{E34} - f\left(L, G_n, c_{COP1d}, c_{COP1n}\right)c_{EC} \\ c_{E34} &= p_{25}c_{E4}c_{E3}/g\left(c_{Lux}, c_{COP1d}, c_{COP1n}\right) \end{aligned} \tag{3.23}$$

Protein concentrations are denoted as c_i, mRNA concentrations c_i^m, index i correspond to the labels for ELF3 (E3), ELF4 (E4) and LUX (LUX). Additional components include nuclear proteins COP1d, COP1n, G_n and the light input L. The functions f and g are used for notation brevity, to imply that the remaining terms are a function of said components.

Protein concentrations are unobserved, as such we approximate Eq.(3.23) using the product of mRNA levels $c_{LUX}^m c_{E3}^m c_{E4}^m$:

$$\frac{d}{dt}c_{\mathrm{EC}} = \alpha c_{LUX}^m c_{E3}^m c_{E4}^m - \lambda c_{EC}$$

as previously explained in Subsection 3.3.1, we approximate the spectrum of EC by

$$\mathbf{C}_{EC} = \alpha\left(\mathbf{D} + \lambda\mathbf{I}\right)^{-1} DFT\left(c_{LUX}^m c_{E3}^m c_{E4}^m\right)$$

where $\mathbf{C}$ is the discrete frequency spectrum and $\mathbf{D}$ is the time derivative as in Section 2.1.3.

Then the spectrum $\mathbf{C}_{EC}$ was incorporated into matrix $\mathbf{U}$ along with the light input. Additionally we fixed parameter $w = 0$ for all h_{ij} in the columns representing the regulation of any component of the network by ELF3, ELF4 and LUX. This was done to avoid the confounding of the effects of these three components with the effect of EC, and given that our prior information is that these three components only interact with the network through EC. These three elements can still be regulated by any of the other elements, including EC (in this way we can include a kind of auto-regulation).Then the inference task is inferring the regulating interactions of components LHY, 'TOC1, GI, PRR5, PRR7, PRR9, ELF3, ELF4 and LUX.

We simulated the ODE system of (Pokhilko et al., 2012) and down sampled it in order to obtain 8 samples per day during 3 days. We simulated photo periods 12/12, 6/18, 8/16, 18/6 and 20/4. We also simulated knock-out mutants ΔLHY, ΔLHY-TOC1, ΔLHY-GI and ΔPRR7-PRR9. We then proceeded to execute DSS for this data-set using the noiseless samples and with 10-SNR white Gaussian noise added. For the noisy case we generated 20 data-sets. DSS was set to the standard parameters as in Section 3.4.2.

Results are plotted in Fig.3.22. On the top left we appreciate the gene expression levels for the wild type simulations along with the light input (red). On the top right we appreciate the AUPR for DSS being the highest at 0.346 for expression only and 0.76 with similarity information incorporated. The spline implementation yielded comparable results at 0.3 for expression and 0.68 with similarity. We also appreciate the method performance for the noisy time series yields a mean AUPR of 0.32 with standard deviation of 0.06 in the expression only model. With added similarity we observe a slight improvement in the mean AUPR, to 0.3479, but with a higher standard deviation of 0.1180.

In bottom left we have the true network with edges going from blue to red, black being bidirectional regulation. We appreciate on bottom right the network resulting from thresholding the joint posterior over $\mathbf{H}$ at 0.6. In these two networks we appreciate the dominance of LHY, DSS can recover the bidirectional regulation of the PRR5 and PRR9. Even though the regulation of 'TOC1 by LHY is rescued, the regulation of LHY by 'TOC1 wasn not observed. Interestingly, the method can infer the photo-regulation of LHY and GI, as the regulation of EC over GI. On the other hand the regulation of EC over 'TOC1, an important component of the model, couldn not be retrieved. Finally it is worth observing that the method infers a regulation of GI over element ELF4 and LUX, in the model post transcriptional regulation of GI over EC is present.

3.5 Discussion

Inference of gene regulatory networks from expression data is one of the best studied problems in systems biology. Despite this considerable collective effort, the general problem remains ill-posed and, in the absence of extensive data sets and strong domain expertise, a solution to this problem remains elusive. In this light, it is of interest to consider more delimited problems which may be amenable

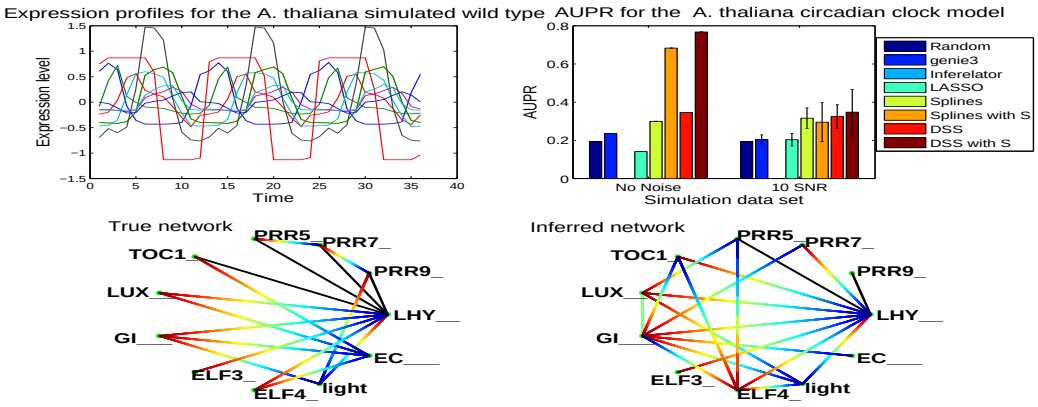


Figure 3.22: Top left are the simulated gene expression using the model of (Pokhilko et al., 2012), profiles for the wild type data set with SNR 100. Top right are the AUPR values for the 2 different noise levels. Bottom left is the true network topology, going from blue (regulators) to red (targets). Bottom right is the inferred network topology obtained by setting a threshold of 0.6 over the inferred matrix $\mathbf{H}$

to specialised but more effective solutions. Oscillatory systems present a prime example of such a problem: while they obviously constitute a specialised subset of regulatory networks, in our opinion they are sufficiently widespread to warrant tailor-made solutions. DSS couples a simplified mechanistic approach (LTI) with frequency domain information to provide such a method. LTI methods in the time domain for *A. thaliana* with experimental data have been studied in Dalchau (2012). Our results on the circadian clock simulation suggest that this frequency domain approach can indeed be fruitful when the model assumptions are reasonably met. As Results over the DREAM and *S. cerevisae* data sets suggest that the method can perform competitively with state of the art methods also when the model assumptions are not precisely met (damped oscillatory behavior); however, in these cases the methods competitive advantage is smaller or inexistent.

The use of derivative and ODE information in a network inference framework has some precedents. A method that is in spirit similar to our approach is Inferelator (Bonneau et al., 2006). It casts the network inference problem as a a parameter inference problem over a first order differential equation system, then estimates the system parameters via regularized regression over a finite differences solution to the system. Recently Bayesian approaches that make use of the derivative information have also been proposed. in Oates and Mukher-

jee (2012) a probabilistic model for integrating a linearized version of network dynamics in a regression framework is presented. (Dondelinger et al., 2013a) attacked the problem of parameter inference of an ODE system jointly with a Bayesian regression over the gene expression levels. The basis of this model is a Gaussian process with product of experts likelihood, not dissimilar from our model in Eq.(3.3). However, the authors in Dondelinger et al. (2013a) did not attempt a joint parameter estimation and variable selection problem, stopping short of formulating the problem in terms of network inference. Basis functions in time domain (splines) have already been applied to network inference problems in systems biology, primarily to model unknown non-linear transition functions (Morrissey et al., 2011). The distinctive part of our work is the proposal of a frequency domain approach for oscillatory systems, and in particular the embedding of our method within a hierarchical framework where integration of additional information is natural. We expect that non-linearities encoded as basis functions as in Morrissey et al. (2011) would be a valuable extension of our work and likely result in an improvement in performance.

While we believe that the DSS method provides promising results, there are several inherent limitations in our approach. Importantly, the LTI approximation implies that self regulation is confounded with decay, so such types of interactions cannot be identified. Empirical results also seem to suggest that post transcriptional interactions may be confounded with transcriptional interactions; this is to be expected, as post-transcriptional interactions are not modelled in our framework. For such reasons, direct application to models that include complex post-transcriptional interactions, such as (Pokhilko et al., 2012), requires stronger supposition about the postranscriptional interactions in order to couple the formation of protein complexes. Furthermore, as all Bayesian network inference methods, DSS also suffers from multi-modal posterior distributions. The use of auxiliary information, such as sequence similarity, can be beneficial to ameliorate this problem. Many different types of auxiliary information can be considered, and indeed alternative models for incorporating sequence similarity could also be used. A major strength of a Bayesian hierarchical model is that different models for auxiliary information could be easily incorporated within the DSS framework.

Chapter 4

Experimental design for inference over the *A. thaliana circadian clock network.*

4.1 Introduction

As discussed in previous chapters, modern biochemical experiments for measuring gene expression can be very complex and require sophisticated techniques for measuring and controlling the variables involved. These experiments are often costly in both researcher time and other resources. For this reason, it is important to minimize the number of experiments while maximizing their information content.

Experimental design is the branch of statistics and operations research which is concerned with maximizing the information content of novel experiments. From a statistical point of view, the utility criterion for evaluating an experiment is a function of the probabilistic model chosen to represent the data-generating process. Depending on the objective of the experiment, the selection criterion can be

- Maximize the information content of an experiment in order to estimate a set of parameters (*estimation criterion*)
- Improve the prediction qualities of a fitted model (*prediction criterion*) .

In the following chapter we use a Bayesian approach to experimental design for dynamical models of biological systems. We restrict our attention to gene reg-

ulatory network (GRN) models, where the systems dynamics are generated by mutual interactions between genes which can modulate each others rate of expression as exposed in Chapters 1 and 2.

Dynamical systems such as ordinary differential equations (ODE) are widespread techniques for modeling GRNs. Previous work has considered experimental design and model selection techniques for non-linear ODE-based models of biological processes. Liepe et al. (2013) employ an approach based on mutual information which could be evaluated using Monte-Carlo simulations. This method is computationally intensive and crucially requires prior knowledge over the model components and their interactions: the structure and functional form of the equations defining the models is assumed known, and all the uncertainty is in the parametrisation. In reality, most models in systems biology are subject to considerable structural uncertainty, and clarifying the structure of interactions is the primary goal of systems biology experiments.

In this work we extend the Bayesian experimental design approach to models with structural uncertainty, formalized as hierarchical Bayesian models. We derive a *Bayesian experimental design score* for quantifying the information gain offered by different experiments. The abstract view of the method is shown on Fig.4.1. We start by using some preliminary data (in the form of observed oscillatory expression levels) to learn a (posterior) probability distribution over a linear approximation of the system as in 3.2. Then we proceed to simulate an experimental intervention by constraining some components of the model to fixed values (the specific details of how we model interventions are given in section 4.4.2). We use the probability distribution over our linear system (represented by the blurred arrows in the learnt system of the figure) to perform a probabilistic simulation the gene expression levels of all the other components given the experimental intervention (in the figure, the blurred lines represent uncertainty over the experimental outcomes). We will then proceed to choose the experimental intervention that would potentially reduce maximally the uncertainty over the learnt system.

This chapter will introduce basic concepts of information theory an optimal design for linear models. Then we develop these concepts for the DSS framework. We then finally illustrate our approach on a benchmark systems biology problem, the circadian clock of the *Arabidopsis thaliana* model plant (Pokhilko et al., 2012). We consider three classes of possible experiments: alterations to the light-dark

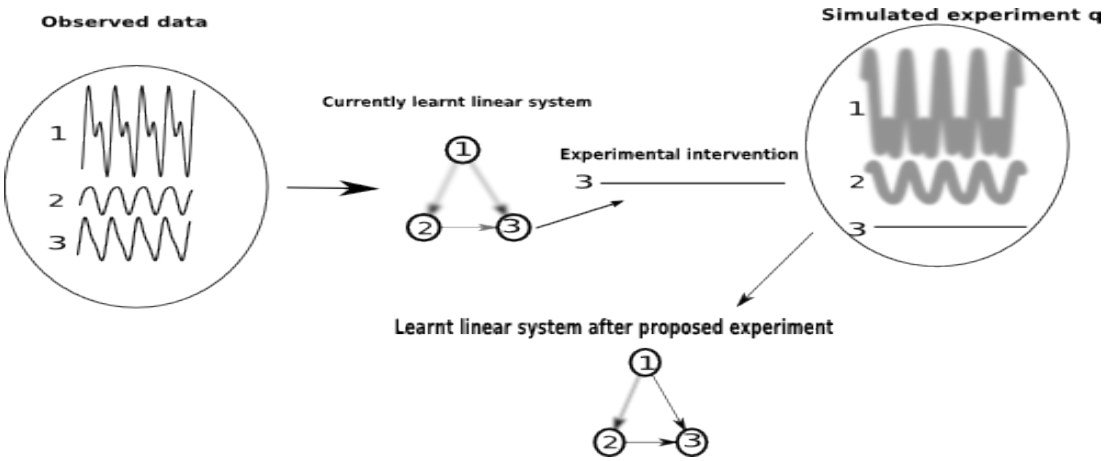


Figure 4.1: Basic illustration of our experimental design approach. After a set of observations the distribution over the learnt system (blurred arrows) is used to draw samples of the experimental outcomes given an intervention (uncertainty over the outcomes is also represented by blurred functions). The aim is to choose the experiment that reduces the uncertainty over the learnt system (represented by the system with well defined arrows in the figure).

input provided to the plant, direct measurements of regulatory links via chromatin immuno-precipitation (ChIP), and gene knock-outs. These commonly performed experiments are very different in terms of costs, and our preliminary results on their relative informativeness could be useful for practitioners.

4.2 Basic concepts of Information theory.

4.2.1 Entropy and mutual information

The first concept we introduce , *Shannon information content* defined over over a *discrete* random variable y with probability$P(y)$. It measures the 'degree of surprise on learning the value of y' (Bishop, 2001), and is given by

$$h(y) = -\log P(y)$$

as the logarithm of 0 is not defined, we define $H(y) = 0$ if $P(y_i) = 0$.

The average Shannon information content of y is called *entropy* or *marginal entropy,* we will denote it by $H(p_y)$. It is obtained by averaging the Shannon information over all the possible values of y. By having y_i represent one of the

possible values for the discrete variable y, the marginal entropy is

$$H(p_y) \equiv -\sum_i P(y_i) \log P(y_i)$$

This concept of entropy can be extended for a continuous distribution $p(y)$ as the expected value of the Shannon information. This is known as *differential entropy* and denoted by $H(p_y) \equiv \mathbb{E}_y[-\log p(y)]$. Explicitly the entropy is given by the integral

$$H(p_y) = -\int p(y) \log p(y)\, dy \tag{4.1}$$

which is a measure of the uncertainty in a distribution (Barber, 2012).

We can define different kinds of entropy between distributions. We start with the concept of *joint entropy* of random variable y and a random variable z with marginal distributions $p(y)$ and $p(z)$ with joint distribution $p(y,z)$ is

$$H(p_y, p_z) = -\int\int p(y,z) \log p(y,z)\, dydz. \tag{4.2}$$

If the variables y and z are independent, then $H(p_y, p_z) = H(p_y) + H(p_z)$.

The *conditional entropy* of y given another random variable z is equal to the expected value of the entropy of $p(y|z)$, that is

$$\begin{aligned} H\left(p_{y|z}\right) &= -\int p(z) \int p(y|z) \log p(y|z)\, dzdy \\ &= -\int p(y,z) \log p(y|z). \end{aligned} \tag{4.3}$$

The conditional entropy measures the uncertainty that remains about y when z is known(MacKay, 2003).

The joint entropy is a function of the conditional entropy and marginal entropy, by the equation

$$H(p_y, p_z) = H(p_y) + H\left(p_{z|y}\right) = H(p_z) + H\left(p_{y|z}\right).$$

The *mutual information* measures the reduction in uncertainty about y that results from observing z (MacKay, 2003) and is defined as

$$I(y;z) \equiv \int\int p(y,z) \log \frac{p(y,z)}{p(y)\, p(z)}. \tag{4.4}$$

The mutual information is related to the joint entropy, marginal entropy and conditional entropy, we will first derive one of such identities

$$I(y;z) = \int\int p(y,z) \log \frac{p(y,z)}{p(y)\, p(z)} dydz$$

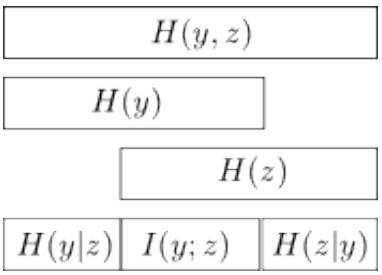


Figure 4.2: Relationship between joint entropy, marginal entropy, conditional entropy and mutual informationBarber (2012).

$$
\begin{aligned}
&= \int\int p(y,z)\left[\log\frac{p(y,z)}{p(z)} - \log p(y)\right] dydz \\
&= \int\int p(y,z)\left[\log p(y|z) - \log p(y)\right] dydz \\
&= \int\int p(y,z)\log p(y|z)\, dydz - \int \log p(y) \int p(y,z)\, dydz \\
&= \int\int p(y,z)\log p(y|z)\, dydz - \int p(y)\log p(y) \\
&= -H\left(p_{y|z}\right) + H\left(p_y\right).
\end{aligned}
$$

Proceeding in a similar way, we can derive the following identities

$$
\begin{aligned}
I(y;z) &= I(y;z) \\
&= H\left(p_y\right) - H\left(p_{y|z}\right) \\
&= H\left(p_z\right) - H\left(p_{z|y}\right) \\
&= H\left(p_y\right) + H\left(p_z\right) - H\left(p_y, p_z\right) \\
&= H\left(p_{y,z}\right) - H\left(p_{y|z}\right) - H\left(p_{z|y}\right)
\end{aligned}
$$

these identities are illustrated in the diagram of Fig.4.2.

The last concept to review is the *conditional mutual information* between variables y and z given the random variable w; this is

$$I(y;z|w) = H\left(p_{y|w}\right) - H\left(p_{y|z,w}\right)$$

which measures how much information about y is conveyed by z assuming w is known.

4.2.1.1 Relative entropy

The *relative entropy* or *Kullback-Leibler divergence* (KL-divergence) is a measure of the difference between two *probability distributions*. Having $p(y)$ and $q(y)$

being two probability distributions over random variable y, the KL-divergence is

$$KL(p\|q) = \int p(y) \log \frac{p(y)}{q(y)} dy$$

this measure is *unbounded*, *positive* and not *symmetric*, that is

$$\begin{aligned} KL(p\|q) &\geq 0 \\ KL(p\|q) &\neq KL(q\|p). \end{aligned}$$

The entropy of a distribution $p(y)$ can be expressed as the KL-divergence between $p(y)$ and a *uniform distribution* $u(y)$. That is

$$\begin{aligned} KL(p\|u) &= \int p(y) \log \frac{p(y)}{u(y)} dy \\ KL(p\|u) &= \int p(y) \log p(y)\, dy - \int u(y) \log u(y)\, dy \\ KL(p\|u) &= -H(y) + const \end{aligned}$$

this says that the distribution p will be more informative about y the more different it is from the uniform distribution.

Similarly, by Eq.(4.4), the mutual information between two random variables y with distribution $p(y)$ and z with distribution $p(z)$ is given by the KL-divergence of the joint distribution $p(y,z)$ and the distribution given by the product of marginals $p(y)p(z)$

$$\begin{aligned} I(y;z) &= \int\int p(y,z) \log \frac{p(y,z)}{p(y)p(z)} dydz \\ &= KL(p(y,z)\|p(y)p(z)) \end{aligned}$$

More insight can be gained by viewing $I(y;z)$ as the expected value of the KL divergence between a conditional and a marginal probability distribution

$$\begin{aligned} I(y;z) &= \int\int p(y,z) \log \frac{p(y,z)}{p(y)p(z)} dydz \\ &= \int\int p(y,z) \left[\log \frac{p(y,z)}{p(z)} - \log p(y) \right] dydz \\ &= \int\int p(y,z) \left[\log p(y|z) - \log p(y) \right] dydz \\ &= \int\int p(y,z) \log p(y|z)\, dydz - \int\int p(y,z) \log p(y)\, dydz \\ &= \int p(z) \int p(y|z) \log p(y|z)\, dzdy - \int p(z) \int p(y|z) \log p(y)\, dydz \\ &= \int p(z) \int p(y|z) \log \frac{p(y|z)}{p(y)} dzdy \\ &= \mathbb{E}_z \left[KL(p(y|z)\|p(y)) \right]. \end{aligned} \tag{4.5}$$

4.2.2 Fisher information matrix

The KL-divergence is useful in determining how two distributions are similar. By having a set of observations over experimental outcomes given by vector $\mathbf{y}^q$, a set of basis functions Φ^q and a parameter vector θ, the likelihood function is given by the distribution $p(\mathbf{y}^q|\Phi^q,\theta)$, see Section 2.2.

We wish derive the information that y conveys about θ having a likelihood function $p(\mathbf{y}^q|\Phi^q,\theta)$. Now suppose we want to determine the effect of a small change of the parameter values has over the distribution $p(\mathbf{y}^q|\Phi^q,\theta)$. This small change will be given by the variables δ_{θ_i} such that $|\delta_{\theta_i}| \ll 1$ for all i components of θ. Lets call Δ_θ the vector with elements $\theta_i + \delta_i$. Then, the difference between the original distribution $p(\mathbf{y}^q|\Phi^q,\theta)$ and the distribution $p(\mathbf{y}^q|\Phi^q,\Delta_\theta)$, can be computed by their KL-divergence:

$$KL\left(p(\mathbf{y}^q|\Phi^q,\theta)\,\|p(\mathbf{y}^q|\Phi^q,\theta+\Delta_\theta)\right) = -\int p(\mathbf{y}^q|\Phi^q,\theta)\,\Delta\log p(\mathbf{y}^q|\Phi^q,\theta)\,d\mathbf{y}^q. \tag{4.6}$$

where $\Delta\log p(\mathbf{y}^q|\Phi^q,\theta) = \log p(\mathbf{y}^q|\Phi^q,\theta+\Delta_\theta) - \log p(\mathbf{y}^q|\Phi^q,\theta)$

This KL divergence of Eq.(4.6) is a function that has a global optimum when both distributions are the same. We are interested on this point, so we use Taylor expansion up to the second term around θ. Let's denote KL_{Δ_θ} the KL-divergence presented in Eq.(4.6), and for notation brevity we will prescind of explicitly denoting $p(\mathbf{y}^q|\Phi^q,\theta)$ and write everything in function of p. Thus the Taylor expansion for the term $\Delta\log p$ around θ is

$$\Delta\log p = \Delta_\theta^T\left(\nabla_\theta\log p\right) + \frac{1}{2}\Delta_\theta^T\left(\mathbf{H}\left(\log p\right)\right)\Delta_\theta + \ldots \tag{4.7}$$

as the function has a global minima in θ then $\nabla_\theta\log p = 0$; thus we can express $\Delta\log p$ in terms of the Hessian matrix. We substitute Eq.(4.7) into Eq.(4.6) and yields

$$KL_{\Delta_\theta} \approx -\frac{1}{2}\int p\Delta_\theta^T\left(\mathbf{H}\left(\log p\right)\right)\Delta_\theta dy.$$

where $\mathbf{H}\log p$ is the *Hessian matrix*[1] of $\log p$.

[1]For a function $f(\mathbf{x})$, for $\mathbf{x}\in\mathbb{R}^N$ and $f(\mathbf{x})\in\mathbb{R}$ the Hessian is the matrix $\mathbf{H}(f)\in\mathbb{R}^{N\times N}$ such that its entries $\mathbf{H}(f)_{ij}$ are

$$\mathbf{H}(f)_{ij} = \frac{\partial^2}{\partial x_i\partial x_j}f.$$

We rearrange the terms and derive the representation

$$KL_{\Delta_\theta} \approx -\frac{1}{2}\sum_i \sum_j \delta_i \delta_j \mathcal{I}(\theta)_{i,j}$$

where

$$\mathcal{I}(\theta)_{i,j} = -\int p(\mathbf{y}^q|\Phi^q,\theta)\,\frac{\partial^2}{\partial\theta_i \partial\theta_j}\log p(\mathbf{y}^q|\Phi^q,\theta)\,dy \tag{4.8}$$

is the *Fisher information matrix*, it is a metric for the amount of information that the variable y carries over the parameter vector θ.

Applying the Faà di Bruno's formula to the second partial derivative of $\log p$ we have

$$\frac{\partial^2}{\partial\theta_i\partial\theta_j}\log p = \frac{1}{p}\frac{\partial^2}{\partial\theta_i\partial\theta_j}p - \frac{1}{p^2}\frac{\partial p}{\partial\theta_i}\frac{\partial p}{\partial\theta j}$$

by the chain rule $\frac{\partial}{\partial\theta_i}\log p = \frac{1}{p}\frac{\partial p}{\partial\theta_i}$, then, the second partial derivative in terms of $\log p$ is

$$\frac{\partial^2}{\partial\theta_i\partial\theta_j}\log p = \frac{1}{p}\frac{\partial^2}{\partial\theta_i\partial\theta_j}p - \frac{\partial}{\partial\theta_i}\log p\frac{\partial}{\partial\theta j}\log p \tag{4.9}$$

finally we substitute Eq.(4.9) into Eq.(4.8), the integral term can be re expressed as

$$-\int p\frac{\partial^2}{\partial\theta_i\partial\theta_j}\log p dy = \int\left(\frac{\partial^2}{\partial\theta_i\partial\theta_j}p - p\frac{\partial}{\partial\theta_i}\log p\frac{\partial}{\partial\theta j}\log p\right)dy$$

if $p(\mathbf{y}^q|\Phi^q,\theta)$ obeys the regularity condition (see (Kullback, 1968)):

$$\int \frac{\partial^2}{\partial\theta_i\partial\theta_j}p(\mathbf{y}^q|\Phi^q,\theta)\,dy = 0$$

we can re express Eq.(4.8) as

$$\mathcal{I}(\theta)_{i,j} = \mathbb{E}_{p(y|\theta)}\left[\frac{\partial}{\partial\theta_i}\log p(y|\theta)\,\frac{\partial}{\partial\theta_j}\log p(y|\theta)\right]$$

thus in these cases the Fisher information matrix is positive semidefinite.

4.3 Optimal design

Classical approaches to statistical experimental design have been developed primarily for linear regression models. In this class of models the data is assumed to be linearly dependent to a set of experimental covariates. As seen in Section 2.2.3 Eq.(2.11), the mathematical setting is the following: let an experiment q be given

an experimental design Φ^q and parameters θ, and denote the experimental observations for experiment q as $\mathbf{y}^q$. In a linear regression model, the experimental outputs are assumed to be a linear combination of the covariates such that

$$\mathbf{y}^q = \Phi^q\theta + \epsilon \tag{4.10}$$

where ϵ is zero-mean Gaussian noise with variance σ^2. The likelihood function will be

$$p\left(\mathbf{y}^q|\Phi^q,\theta\right) = \prod_{i=1}^{M}\mathcal{N}\left(y_i^q - \boldsymbol{\phi}_i^q\theta,\sigma^2\right) \tag{4.11}$$

For the linear regression model the Fisher information matrix will be given by the *precision matrix*, that is

$$\mathcal{I}\left(\theta\right) = \frac{1}{2\sigma^2}\Phi^T\Phi$$

which is equal to the the inverse of the *covariance matrix* for this model.

An experimental design for the linear model with a sample size n, assumes that the experimenter can choose the basis functions ϕ_i is contained in the design matrix Φ^q. The objective,see Pukelsheim (2006), will be to attain the smallest covariance matrix (or biggest Fisher information matrix, according to a *Loewner ordering*[2] or a *monotonic matrix function*[3].

In this framework an experimental design "specifies $l < n$ distinct basis function vectors ϕ_i and assigns to them frequencies n_i that sum to n", thus defining a measure over the design space (Chaloner and Verdinelli, 1995). Its objective is to tell the experimenter to perform n_i observations under the experimental conditions, these experimental conditions determine the basis functions ϕ_i. For

[2]For Hermitian matrices $\mathbf{A}$ and $\mathbf{B}$ of rank k, the Loewner ordering is defined by:

- $\mathbf{A} \geq \mathbf{B} \Longleftrightarrow \mathbf{A} - \mathbf{B} \geq 0 \Longleftrightarrow \mathbf{A} - \mathbf{B} \in PSD\left(k\right)$; where $PSD\left(k\right)$ is the set of positive semi-definite matrices of rank k.
- $\mathbf{A} > \mathbf{B} \Longleftrightarrow \mathbf{A} - \mathbf{B} > 0 \Longleftrightarrow \mathbf{A} - \mathbf{B} \in PD\left(k\right)$; where $PD\left(k\right)$ is the set of positive definite matrices of rank k.

This ordering has these properties for all Hermitian matrices $\mathbf{A}$, $\mathbf{B}$ and$\mathbf{C}$:

- *anti symmetry,* $\mathbf{A} \geq \mathbf{B}$ and $\mathbf{B} \geq \mathbf{A} \Longrightarrow \mathbf{A} = \mathbf{B}$
- *reflexivity,* $\mathbf{A} \geq \mathbf{A}$
- *transitivity,* $\mathbf{A} \geq \mathbf{B}$ and $\mathbf{B} \geq \mathbf{C} \Longrightarrow \mathbf{A} \geq \mathbf{C}$

Thus, the Loewner ordering is a partial ordering.

[3]A matrix function is defined over the domain of Hermitian matrices. This function is *isotonic* if it preserves a Loewner ordering and *antitonic* if It is order reversing. In either case, it is called *monotonic*.

an experiment with *design* ξ^q , its *support* is given by the set of l basis functions, denoted by $supp(\xi^q) = \Phi^q = \{\phi_1, \phi_2 ... \phi_l\}$.

For estimation purposes, the optimality criteria depends on the choice of matrix function from which to evaluate the information matrix. The most popular is the *D-optimal* criterion or maximize $\det(\mathcal{I}(\theta)/n)$. This criterion maximizes the differential Shannon information of the parameters and minimizes the variance of the estimates (Kreutz and Timmer, 2009).

In our setting, the support of the design ξ^q is dependent of the spectra of the gene expression levels, as such it is not known a priori. Because of this, it is not possible to apply classical optimal design criteria. Instead we employ its Bayesian counterpart, *Bayesian experimental design*. In this framework we are going to exploit the mutual information between the prior and posterior distributions over the model parameters after performing an experiment. This has the added benefit of quantitatively assessing the information content of an experiment with respect to parameter estimation objectives.

4.4 Bayesian experimental design.

4.4.1 Information content of an experiment.

In his seminal work, Lindley (1956) sets experimental design in a decision-theory framework. He states that the previous knowledge over a system is encoded in the prior probability of its model parameters. The knowledge about parameters θ obtained after an experiment, given the observations y^q and experimental conditions ξ^q will be contained in the posterior distribution $p(\theta|y^q, \xi^q)$. Thus the information gained after an experiment can be expressed in terms of the KL-divergence between both distributions

$$I(\theta; \mathbf{y}^q) = \mathbb{E}_{\mathbf{y^q}}[KL(p(\theta|\mathbf{y}^q) \| p(\theta))]$$

Thus the *utility* of an experiment q with conditions ξ^q (which we will denote by $U(\theta; \mathbf{y}^q; \xi^q)$) is obtained by solving

$$U(\theta; \mathbf{y}^q; \xi^q) = \int\int \log \frac{p(\theta|\mathbf{y}^q, \xi^q)}{p(\theta)} p(\theta, \mathbf{y}^q|\xi^q)\, d\theta dy^q. \tag{4.12}$$

This utility function gives rise to what is known as *Bayesian D-optimal design* (Chaloner and Verdinelli, 1995). In order to choose the best experimental design,

the objective is to maximize the value of the utility function $U(\theta, y^q, \xi^q)$ over the set of parameters and (unobserved) responses. Unlike classic optimal design, we aim at leveraging prior information encoded in the prior distribution of the parameters.

Whereas these ideas were introduced in the linear regression case, extending to different scenarios is conceptually trivial; however, the computational simplifications afforded by linear models are then lost, giving rise to an analytically intractable problem. Liepe et al. (2013) employ the same utility criteria over a set of parameters for a nonlinear system of differential equations and then proceed to compute the utility function by Monte Carlo simulation. This requires at each step to simulate the experimental outcomes by solving the system, a procedure which may incur in severe computational overhead depending of the model size and parameters. Furthermore, the model structure is assumed fixed; introducing uncertainty in the model structure would add a further dimension to the already complex computational problem, ruling out all but the simplest problems.

4.4.2 Bayesian experimental design for the Frequency-LTI model

Having specified the DSS family of models in Chapter 3, we now discuss in detail the experimental design techniques for three classes of experiments. The starting point is a prior distribution over LTI coefficients, which in itself could be (and, generally, is) the posterior distribution from some previous experiments. The crucial problems are two, how can an experimental perturbation be encoded mathematically within the model? how can we compute the utility score for a perturbation?

The answer to these questions depends on the specific perturbation considered; here we focus on three commonly employed experiments. The first type are changes in the external input to the system, the U matrix in Eq.(2.7). We denote this class of experiments as *photo-period experiments*, since in the case study of *A. thaliana* the input matrix represents the light inputs to the circadian clock. The second type are mutagenic experiments, where a single gene is removed from the system (*knock-out*). The third type are observation experiments, where presence/ absence of one or more edges is observed directly through experiments such as Chromatin Immunoprecipitation (ChIP) or any affinity-binding detection methods.

Notice that observation experiments are somewhat different from the other types, as they do not constitute a perturbation of the system; for this reason, in the following we describe experimental design methodologies for observation experiments separately.

4.4.2.1 Photo-period experiments and knock-out experiments

In the DSS setting, we frame experimental design for photo-period and knock-out settings as choosing the best experiment q defined as $i$nterventions in matrix $[\mathbf{X^q U}^q]$ that maximizes the *information gain* over the parameters $\mathbf{\Theta} = [\mathbf{A}, \mathbf{C}]$ of the linear dynamical model of Eq.(2.7). An $i$ntervention consists of setting a column of $\mathbf{U}^q$ or $\mathbf{X}^q$ to a known value ξ^q (zero in case of knock-out experiments or the frequency spectrum for a light signal in the case of photo-period experiments). We will denote the intervened element as column(s) $\mathbf{X}_i^q$ and the rest of the columns as $\mathbf{X}_{\setminus i}^q$.

The utility function of Eq.(4.12) can be computed by calculating the KL-divergence between the current distribution of the LTI-coefficients (either prior distribution or posterior distribution of a previous experiment) and the posterior distribution over said parameters after performing the desired experiment. This implies that we have to be able to compute the expected value of the next experiment's observations, in order to compute the mutual information and thus the utility of the next experiment. Explicitly this utility function is

$$U\left(\mathbf{\Theta}; \mathbf{X}^q; \xi^q\right) = \int\int p\left(\mathbf{X}_{\setminus i}^q, \mathbf{\Theta} | \mathbf{X}_i^q = \xi^q\right) \log \frac{p\left(\mathbf{\Theta} | \mathbf{X}_{\setminus i}^q, \mathbf{X}_i^q = \xi^q\right)}{p\left(\mathbf{\Theta}\right)} d\mathbf{X^q} d\mathbf{\Theta}$$

the prior (current knowledge) $p\left(\mathbf{\Theta}\right)$ does not depend on the next, simulated experiment (we simulate using the current knowledge), as such, the selection criteria can be stated in terms of the numerator as the integral

$$\int\int \mathbf{p}\left(\mathbf{X}_{\setminus \mathbf{i}}^{\mathbf{q}}, \mathbf{\Theta} | \mathbf{X}_{\mathbf{i}}^{\mathbf{q}} = \xi^{\mathbf{q}}\right) \log \mathbf{p}\left(\mathbf{\Theta} | \mathbf{X}_{\setminus \mathbf{i}}^{\mathbf{q}}, \mathbf{X}_{\mathbf{i}}^{\mathbf{q}} = \xi^{\mathbf{q}}\right) \mathbf{dX^q d\Theta} \tag{4.13}$$

The conditional distribution $p\left(\mathbf{\Theta} | \mathbf{X}_{\setminus i}^q, \mathbf{X}_i^q = \xi^q\right)$ as derived in Trejo Banos et al. (2015a) is a result of a Linear regression model with Gaussian likelihood. As such the conditional over the coefficients $\mathbf{\Theta}$ can be obtained by factorizing, and is

$$\log p\left(\mathbf{\Theta} | \mathbf{X}^q, \xi^q\right) \quad \propto \quad \log \left[\det\left(\sigma_D^{-2} \mathbf{\Sigma}^{-1}\right)^{-1/2}\right] \tag{4.14}$$

$$-\frac{1}{2\sigma_D^2}\left(-2\bar{\boldsymbol{\eta}}^{\mathrm{T}}\bar{\boldsymbol{\Theta}}+\bar{\boldsymbol{\Theta}}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\bar{\boldsymbol{\Theta}}\right) \tag{4.15}$$

We evaluate Eq.(4.13) through Monte Carlo simulation by drawing a sample from the joint distribution

$$p\left(\mathbf{X}_{\setminus i}^{q},\boldsymbol{\Theta}|\mathbf{X}_{i}^{q}=\xi^{q}\right)=p\left(\mathbf{X}_{\setminus \mathbf{i}}{}^{q}|\boldsymbol{\Theta},\mathbf{X}_{i}^{q}=\xi^{q}\right)p\left(\boldsymbol{\Theta}\right) \tag{4.16}$$

The Monte Carlo algorithm will consist of integrating $U_{DSS}\left(\bar{\boldsymbol{\eta}},\boldsymbol{\Sigma},\boldsymbol{\Theta}\right)_{DSS}$ over both random variables

$$\frac{1}{S_1}\sum_{s_1=1}^{S_1}\left(\frac{1}{S_2}\sum_{s_2=1}^{S_2}\log p\left(\boldsymbol{\Theta}^{(s_1)}|\mathbf{X}_{\setminus i}^{q(s_2)},\mathbf{X}_{i}^{q}=\xi^{q}\right)\right) \tag{4.17}$$

we draw a sample $\boldsymbol{\Theta}^{(s_1)}$ from $p(\boldsymbol{\Theta})$, then we evaluate Eq.(4.14) by drawing samples $\mathbf{X}_{\setminus i}^{q(s_2)}$ from the conditional distribution term of Eq.(4.16).

4.4.2.2 Sampling from the conditional distribution over the spectra

In Section 3.2 we presented the DSS likelihood function $p\left(\mathbf{R}^{q}|\dot{\mathbf{X}}^{q}\mathbf{X}^{q},\boldsymbol{\Theta},\sigma^{2}\right)$, with $\boldsymbol{\Theta}=\left[\begin{array}{cc}\mathbf{A}^{T} & \mathbf{C}^{T}\end{array}\right]$ for the Frequency-LTI model, which is a Gaussian. The likelihood is

$$p\left(\mathbf{R}^{q}|\mathbf{.}\right)=\mathcal{N}\left(\mathbf{R}^{q},\sigma^{2}\right). \tag{4.18}$$

which is a function of the squared residuals. The residual is given by the matrix equation (see Section 2.1.3)

$$\mathbf{R}^{q}=\mathbf{D}\mathbf{X}-\mathbf{X}\mathbf{A}^{\mathbf{T}}-\mathbf{U}\mathbf{C}^{\mathbf{T}}$$

which can be factorized using the *vec* operator

$$vec\left(\mathbf{D}\mathbf{X}-\mathbf{X}\mathbf{A}\right)-vec\left(\mathbf{U}\mathbf{C}^{T}\right)$$

using the Kronecker product, it can be factorized as

$$\left(\mathbf{I}\otimes\mathbf{D}-\mathbf{A}^{T}\otimes\mathbf{I}\right)vec\left(\mathbf{X}\right)-vec\left(\mathbf{U}\mathbf{C}^{T}\right). \tag{4.19}$$

By substituting Eq.(4.19) into 4.18 we get (we prescind of the first term for notation brevity):

$$p\left(\dot{\mathbf{X}}^{q}|\mathbf{.}\right)\propto\exp\left(\frac{1}{\sigma^{2}}\left(\left(\mathbf{I}\otimes\mathbf{D}-\mathbf{A}^{T}\otimes\mathbf{I}\right)\bar{\mathbf{X}}-\mathbf{U}\bar{\mathbf{C}}\right)^{T}\left(\left(\mathbf{I}\otimes\mathbf{D}-\mathbf{A}^{T}\otimes\mathbf{I}\right)\bar{\mathbf{X}}-\mathbf{U}\bar{\mathbf{C}}\right)\right) \tag{4.20}$$

where $\bar{\mathbf{X}} = vec(\mathbf{X})$ and $\mathbf{U}\bar{\mathbf{C}} = vec\left(\mathbf{U}\mathbf{C}^T\right)$. By applying the technique of completing the square (Bishop, 2001), we can get the conditional distribution over the frequency spectra, from which we can draw samples as it is a Gaussian of the form

$$p\left(\mathbf{X}^q|\mathbf{\Theta}, \sigma^{\mathbf{2}}\right) \sim \mathcal{N}\left(\eta, \Lambda^{-1}\right) \tag{4.21}$$

with

$$\Lambda = \frac{1}{\sigma^2}\left(\mathbf{I}\otimes\mathbf{D} - \mathbf{A}^T\otimes\mathbf{I}\right)^T\left(\mathbf{I}\otimes\mathbf{D} - \mathbf{A}^T\otimes\mathbf{I}\right) \tag{4.22}$$

$$\eta = -\Lambda^{-1}\left(\mathbf{I}\otimes\mathbf{D} - \mathbf{A}^T\otimes\mathbf{I}\right)^T\bar{\mathbf{U}\mathbf{C}}. \tag{4.23}$$

4.4.2.3 Conditioning over a subset of spectra

Lets suppose we select column i of $\left[\begin{array}{cc} X^q & \mathbf{U}\end{array}\right]$ to be perturbed, in the DSS setting it amounts to fixing the spectrum of that column to a fixed known value, lets call this value ξ^q. Then we can draw samples from the the rest of the matrix $\left[\begin{array}{cc} X^q & \mathbf{U}\end{array}\right]$ elements, through matrix factorization[4].

Having a experimental perturbation i with fixed spectrum $\mathbf{X}_i^q = \xi^q$, then the spectra of the rest of the components conditioned on parameters $\left\{\xi^q, \mathbf{B}, \sigma^{\mathbf{2}}\right\}$ is a function of η and Λ presented in the previous section Eq.(4.22) and Eq.(4.23).

We split η into η_2 which contains all those elements of η that correspond to the selected i column, η_1 the rest of the elements. Matrix Λ^{-1} is split accordingly, thus the conditional distribution is expressed as

$$p\left(\mathbf{X}^q_{\setminus i}|\mathbf{A}, \mathbf{C}, \mathbf{X}_i^q = \gamma\right) \sim \mathcal{N}\left(\tilde{\mathbf{m}}, \tilde{\Sigma}\right)$$

[4]For a multivariate normal distribution $\mathcal{N}(\mathbf{m}, \Sigma)$, with mean vector $\mathbf{m} \in \mathbb{R}^{p+q}$ and covariance matrix $\Sigma \in \mathbb{R}^{(p+q)\times(p+q)}$, we can condition a subset $\mathbf{m_1} \in \mathbb{R}^p$ of $\mathbf{m}$, given the rest of the elements of $\mathbf{m}$, denoted $\mathbf{m}_2 \in \mathbb{R}^p$, set to a fixed value γ. Thus if we have

$$\mathbf{m} = \left[\begin{array}{c} \mathbf{m_1} \\ \mathbf{m_2}\end{array}\right] \tag{4.24}$$

the covariance matrix can be subdivided in blocks $\mathbf{A} \in \mathbb{R}^{p\times p}$, $\mathbf{B} \in \mathbb{R}^{q\times q}$ and $\mathbf{C} \in \mathbb{R}^{p\times q}$ such that

$$\Sigma = \left[\begin{array}{cc} \mathbf{A} & \mathbf{C} \\ \mathbf{C}^T & \mathbf{B}\end{array}\right]. \tag{4.25}$$

By conditioning we find that the vector $\mathbf{m_1}$ is distributed according to

$$\begin{aligned} p(\mathbf{m_1}|\mathbf{m_2} = \gamma) &\sim \mathcal{N}\left(\tilde{\mathbf{m}}, \tilde{\Sigma}\right) \\ \tilde{\mathbf{m}} &= \mathbf{m_1} + \mathbf{C}\mathbf{B}^{-1}(\gamma - \mathbf{m_2}) \\ \tilde{\Sigma} &= \mathbf{A} + \mathbf{C}\mathbf{B}^{-1}\mathbf{C}^T \end{aligned} \tag{4.26}$$

$$\begin{aligned}\tilde{\mathbf{m}} &= \eta_1 + \Lambda_{\eta_1\eta_2}^{-1}\left(\Lambda_{\eta_2\eta_2}^{-1}\right)^{-1}(\xi^q - \eta_2)\\ \tilde{\Sigma} &= \mathbf{\Sigma_{xx}} + \Lambda_{\eta_1\eta_2}^{-1}\left(\Lambda_{\eta_2\eta_2}^{-1}\right)^{-1}\Lambda_{\eta_2\eta_2}^{-1}. \end{aligned} \quad (4.27)$$

4.4.3 Finding the information content of an edge.

As a complement to the previous scores, we wished to account for an additional source of information, direct observations over DNA-protein interactions. A result of this kind of experiment can be viewed as an observation over element h_{ij} of matrix $\mathbf{H}$

Here the observed gene expression spectra are considered a fixed set $\mathbf{X}^q$. Having these observations, we aim at choosing which link h_{ij} possess the highest mutual information for learning parameters $\mathbf{\Theta}$. This can be represented in terms of the conditional mutual information, which is a function of two conditional entropies such that $I(\mathbf{\Theta}; h_{ij}|\mathbf{X}^q) = H(\mathbf{\Theta}|\mathbf{X}^q) - H(\mathbf{\Theta}|\mathbf{X}^q, h_{ij})$.

The conditional entropy is not a function of the selected link, so its computation is not necessary for discriminating between links. Then we introduce the utility function U_h equal to the negative conditional entropy of variable $\mathbf{\Theta}$ given the gene expressions $\mathbf{X}^q$ and the observed link h_{ij}

$$\begin{aligned} U_h(\mathbf{\Theta}, \mathbf{X}^q, h_{ij}) &= -H(\mathbf{\Theta}|\mathbf{X}^q, h_{ij})\\ &= \sum_{\gamma\in\{0,1\}} p(h_{ij}=\gamma)\times \\ &\quad \int p(\mathbf{\Theta}|\mathbf{X}^q, h_{ij}=\gamma)\log p(\mathbf{\Theta}|\mathbf{X}^q, h_{ij}=\gamma)\, d\mathbf{\Theta} \end{aligned}$$

where $p(\mathbf{\Theta}|\mathbf{X}^q, h_{ij}=\gamma)$ is the posterior distribution over $\mathbf{\Theta}$ given a fixed value for link h_{ij} (either 0 or 1).

We evaluate the integral by drawing samples from the conditional posterior $p(\mathbf{\Theta}|\mathbf{X}^q, h_{ij}=\gamma)$, for $\gamma \in \{0,1\}$, and evaluating $\log p(\mathbf{\Theta}|\mathbf{X}^q, h_{ij}=\gamma)$. We integrate by Monte Carlo method, with samples s_3 and s_4 drawn from the posterior distribution $p(\mathbf{\Theta}|\mathbf{X}^q, h_{ij}=\gamma)$. As such the utility criterion is

$$\begin{aligned} U_h(\mathbf{\Theta}; \mathbf{X}^q; h_{ij}) &= \frac{1}{2S_3}\sum_{s_3=1}^{S_3}\log p\left(\mathbf{\Theta}^{(s_3)}|\mathbf{X}^q, h_{ij}=0\right)\\ &\quad + \frac{1}{2S_4}\sum_{s_3=1}^{S_4}\log p\left(\mathbf{\Theta}^{(s_4)}|\mathbf{X}^q, h_{ij}=1\right) \end{aligned} \quad (4.28)$$

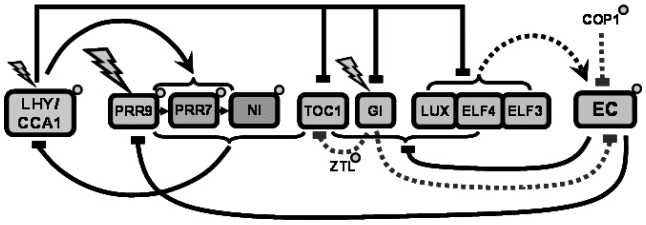


Figure 4.3: Circadian clock model for *A. thaliana,* as shown in Pokhilko et al. (2012). Transcriptional elements LHY, PRR579, GI, 'TOC1, LUX, ELF4 and ELF3 are assumed observed. While the expression levels of the Evening Complex (EC) is unobserved, along with other post-transcriptional interactions involving ZTL and COP1.

4.5 Experimental design for the *A. thaliana* circadian clock model

in Pokhilko et al. (2012) we observe a state of the art model of the *A. thaliana* circadian clock network. It consists of the transcription factors LHY/CCA1 LHY (LATE ELONGATED HYPOCOTYL) and CCA1 (CIRCADIAN CLOCK ASSOCIATED 1), these execute an activating interaction with the transcriptional co-regulators PRR9, PRR7 and PRR5/NI (PSEUDO-RESPONSE Regulators 9, 7, 5/night inhibitor) which at the same time are interlocked in a negative feedback loop with LHY/CCA1. This feedback loop is thought to be the responsible for peak activity of day-time components.

On the other hand we have the evening loop, thought to be driven by EC (Evening complex), composed by the binding of ELF3 (EARLY FLOWERING 3), ELF4 (EARLY FLOWERING 4) and the GARP transcription LUX (LUX ARRHYTHMO) which controls LHY expression by a double negative connection (Pokhilko et al., 2012). A graphical representation of the model is shown in Fig.4.3.

4.6 Results.

We simulate the *A. thaliana* circadian clock model, we selected and sub sampled the simulated data in order to get 12 samples over one light/dark cycle for a Wild Type population. We ran DSS and collected 10000 samples of the joint posterior over the model parameters. We executed DSS using standard parameters as in

3.4 and evaluated the mutual information criterion 4.17, we draw 1000 samples, thus setting parameter $S_1 = 1000$. We draw 100 samples for each gene expression level at each step, thus setting parameter $S_2 = 100$ First, we chose photo-periods of 6/18, 8/16, 18,6 and 20/24, we computed the DFT of a {-1,1} light input (ξ^q) and added it to the spectra matrix. Thus drawing samples from the conditional distribution

$$p\left(\mathbf{X}^q|\mathbf{\Theta}, \sigma^{\mathbf{2}}, \mathbf{U} = \xi^q\right)$$

Then we selected a set of knock out mutants commonly seen in experimental settings. In this way knock-out mutants ΔLHY, ΔLHY-GI, ΔLHY-TOC1 and ΔPRR7-PRR9 were simulated by conditioning the rest of the gene spectra given that the intervened genes have a constant spectrum of zero.

$$p\left(\mathbf{X}^q_{\setminus i}|\mathbf{\Theta}, \sigma^{\mathbf{2}}, \mathbf{X}^q_i = \mathbf{0}\right)$$

In Fig.4.4 we present the results of evaluating Eq.(4.17) for these two set of experiments. The boxes go from the 25th to the 75th percentiles and the red bar indicates the median score. It shows photo-period experiments having a median score between 220 and 225, while the knock-out mutants show less median values ranging from 210 to 217. It is of interest that the lowest information gain looks to be accredited to the ΔLHY-TOC1 double mutant, being these two genes the main drivers of circadian oscillations. This may be due to the nature of the mutual information criterion, as it accounts for the reduction in uncertainty over the estimation of parameters. It seems plausible that the disruption of these two components alters clock behavior enough that parameter inference is less reliable, as the score suggests that the uncertainty over the model behavior grows. This may be in fact another source of information about the importance of these two clock components.

Complementary, we computed the conditional mutual information for Chip experiments according to Eq.(4.28). First we simulated Wild-type gene expression levels for 12 samples over a 24 hour period, using the same procedure as in the previous paragraph. Then, we selected a set of candidate links to observe, these include those known to be part of the true network, and those involving the EC components. Each one of these links was set to their possible values (one and zero), and the posterior distribution calculated for each case, this implies running DSS twice for each studied link with standard parameters as proposed Section 3.4.

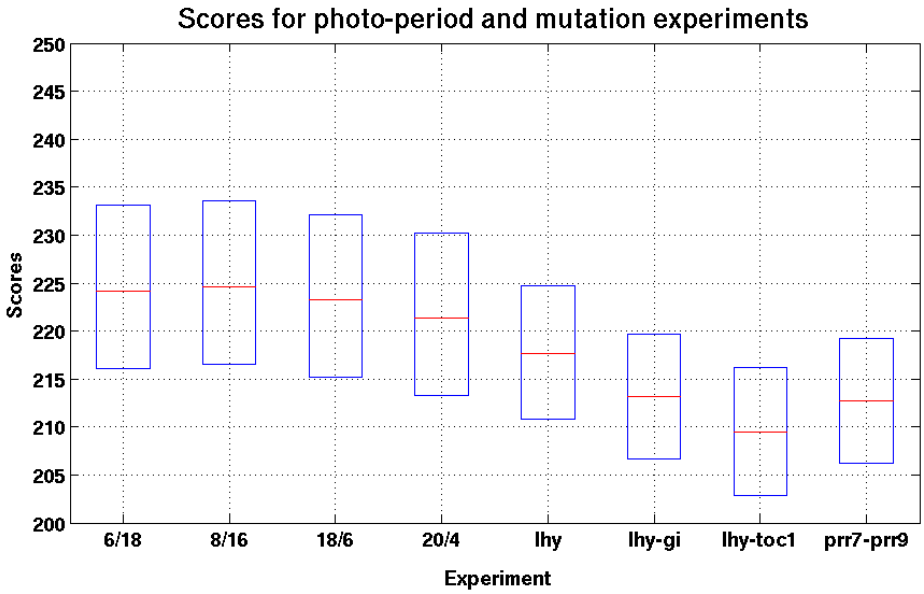


Figure 4.4: Box plot for the evaluation of the DSS criterion, higher score means higher mutual information between experimental design and experimental outcomes. From left to right, photo periods of 6/18, 8/16, 18/6 and 20/4. Then knockout Mutants ΔLHY, ΔLHY-GI, ΔLHY-TOC1 and ΔPRR7-PRR9.

We show the resulting scores in Fig.4.5. In this scatter plot, regulators are shown in the x axis, and the scores are presented through colored dots. Each dot is labeled according to the putative regulation tested (the regulators target is marked by a ->). Here we observe that the regulating interactions involving the elements of the EC complex (LUX, ELF4 and ELF3) as regulators show the lest information. This is not surprising as model assumptions are that the EC complex is the transcription factor involved in the evenning regulation, and its effects even though essential, are not directly observable through its components. On the other hand we find that the most useful information seems to be related to the elucidation of the role of the light input over LHY and specially GI, with the highest score of 437, above of the mean value of 432.7. Another interesting interactions include that for LHY its most useful observation would be its regulation of 'TOC1, correspondingly, LHY would be the most informative interaction to observe for 'TOC1. As stated earlier, the interaction between these two components is the main driver of the morning oscillator.

Taking in account these two complimentary criteria, some decisions about the utility of the experiments can be made. In these case, it seems to points towards

light-related experiments, as the expected mutual information for all the photoperiod experiments seems to be on par. This at the same time could be validated by the fact that light-input nodes of the network seem to be the most informative in first instances.

Finally the LHY-TOC1 double mutant score suggest that the behavior of the system under these circumstances is more uncertain. This seems to be corroborated by biological studies that show severe disruption of the circadian clock in the LHY-CCA-TOC1 triple mutant (Ding et al., 2007). The authors mention these experiments along with the LHY-CCA-GI mutant as key in providing experimental evidence to support a relationship between a core loop and a secondary loop of the clock. With only one data set available we were able to suggest one biologically relevant experiment by interpreting the mutual information curve.

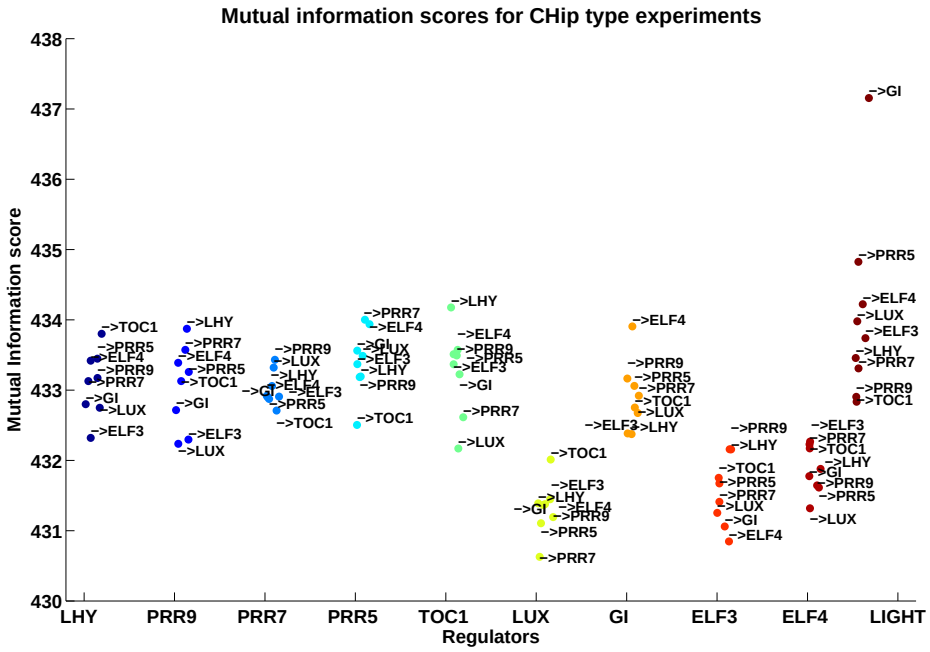


Figure 4.5: Scatter plot of the conditional mutual information scores for observations over some edges. Each score is labeled with the represented interaction. The regulating interactions are symbolized by a "->" as "->targets", with the regulator being the label on the x axis tick. From left to right we have regulators LHY, PRR9, PRR7, PRR5, 'TOC1, LUX, GI, ELF3, ELF4 and photo-regulation in case of light inputs.

As a complement we wished to observe the distribution properties over the spectra $\mathbf{X}^q$ and the solution trajectories to the LTI system. For this, we proceeded to sample a set of spectra given a sample from $\{\mathbf{A}, \mathbf{C}\}$ by Eq(.4.27). Then

we converted these samples to the time domain by the IDFT Eq.(2.5) . We drew 1000 thousand samples from this distribution for each of the previously mentioned experiments, the median values are plotted in Fig.4.6. Here we observe some interesting characteristics. Component LHY appears to follow two different patterns of activity in radically different conditions; it presents a peak activity at the twelfth hour mark for photo periods of 6/18 and 20/4, whereas at 8/16 and 18/6 peak activity occurs at the fourth hour and second hour marks respectively . We hypothesize That this amounts to the model capturing some of the light-input dependency of LHY, but may not process sufficient information to discern phase information (LHY as morning component of the clock has peak activity in the early onset of the photo period).

We wished to see how much uncertainty over the sampled spectra remains after one experiment, for this we plotted the 25th and 75th percentiles of the samples at each time point. The resulting percentile values were almost identical at each time point across genes in all experiments, as such, we proceed to plot only the average of these values and show them as error bars in figure . Here we can appreciate in color the modes of the gene expression level, meanwhile their values oscillate between -0.01 and 0.01, the percentile values go from -0.1 to 0.1. This shows the great amount of uncertainty around the system behavior. Interestingly, the shape and size of the shaded area is very similar along all these experiments, being narrower at 6 and 18 hours and wider at 0, 12 and 22 hours. How much of these uncertainty is because the initial experimental setting and how much is because of the system dynamics is an interesting avenue for further research.

4.7 Conclusions

We have presented a methodology for Bayesian experimental design in biological dynamical systems with structural uncertainty. Experimental design is a branch of classical computational statistics which is gaining increasing attention in systems biology, due to inherent complexity and uncertainty of biological systems. Adapting classical methods to modern systems biology is problematic, as sources of uncertainty are ubiquitous in systems biology data, leading to computationally intractable problems and/ or predictions with large associated uncertainty. In general, handling both parametric and structural uncertainty in nonlinear sys-

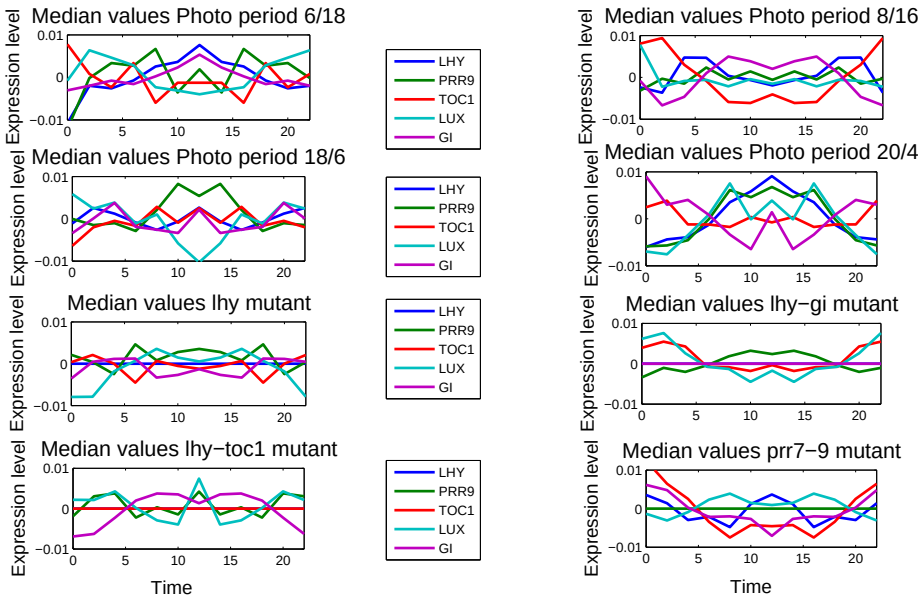


Figure 4.6: Median values for 1000 samples of an experiment resulting expression levels. Time in hours on the x axis. The transcriptional elements of the clock LHY, PRR9, 'TOC1, LUX and GI where chosen.

tems is highly problematic. Earlier work such as (Liepe et al., 2013) chose to focus on non-linear systems without structural uncertainty. However, in many biological systems, such as oscillatory systems, it may be preferable to approximate the system dynamics to gain computational savings which will enable structural uncertainty to be considered in experimental design. Our results on the *A. thaliana* clock model show that this approach can be fruitful, highlighting potentially large differences in information content for different classes of experiments, and for different individual experiments in each class. These results are potentially precious for practitioners, whose prime preoccupation is often the prioritisation of experiments in the face of technical and resource limitations.

There are several directions along which the approach could be further developed. A simple, but potentially useful, extension would be to modify the utility function by explicitly accounting for the different costs of different experiments. It would also be of interest to develop strategies for planning multiple experiments, as the information gain is generally a non-linear function on the space of possible experiments. While the same approach can be easily deployed for small sets of experiments, the general issue of multiple experimental design yields a very challenging discrete optimisation problem. We envisage that ideas from

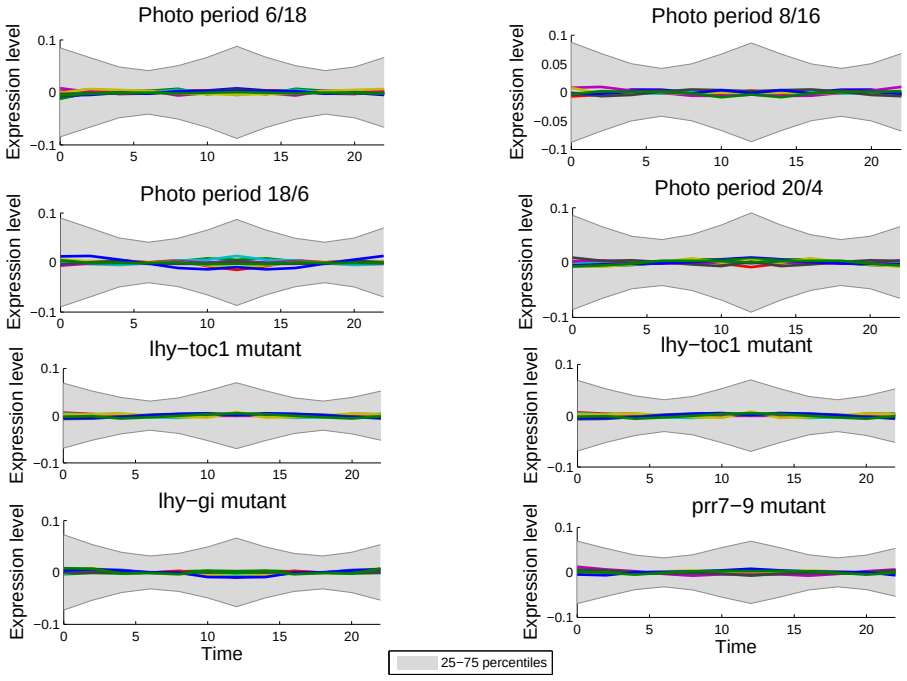


Figure 4.7: Error bars showing the distribution of data between the 25th and 75th percentiles (gray area). Time in hours is shown in the x-axis. The modes of the drawn spectra are plotted in color. Notice how the modes and the error bars appear to be in two different scales. This may reflect the great amount of uncertainty about the system dynamics after just one experiment.

reinforcement learning could be effective in this scenario.

Chapter 5

Conclusions and future work

Oscillatory behavior in biological systems is a difficult phenomenon to analyze. How can the complex nonlinear relationships in a genetic regulatory networks result in regular and robust behaviors?

Given the scope of the problem, approximations are used in order to understand the "blueprints" of this machinery. Here we started from a classic framework for studying biological systems, linear ODE models. We approached this classic modeling paradigm in a novel way for systems biology by using its frequency domain representation. We then embedded this linear model into a Bayesian framework in order to account for both linearization error and uncertainties derived from biological experimentation.

The resulting method DSS showed promising performance when the assumptions of regular oscillatory behavior is fulfilled. We tested our method under non oscillatory conditions and its performance remained comparable to other current methods. Still the model and its experimental design expansion present some limitations and potential for improvement.

5.1 DSS model criticism and extension

The DSS model introduced in Chapter 3 is based on the fundamental assumption that the linearized dynamics of the system are representative of the true dynamics. For this, the system would require to be in an equilibrium point. This assumption obviously isn not met in oscillatory systems. We aimed at alleviating this problem by setting the Bayesian hierarchical model on top of this linearization. As discussed at the end of Chapter 3, modeling the nonlinearities

(as deviations from the equilibrium point) along with the linear dynamics, may prove to be a productive extension to the method.

Another modeling compromise, working with ODE, leaves out intrinsic and extrinsic noise terms. Accounting for these factors should provide a more detailed approach to elucidate networks interactions. For example in (Ocone et al., 2013) a hybrid discrete-continuous stochastic model for parameter inference is used. Exploring a frequency domain extension could provide us with a good angle of attack to properly characterize oscillatory systems.

From the statistical point of view, the form of sparsity inducing prior plays a pivotal role. We chose a spike and slab prior to perform shrinkage and model selection. There are forms of spike and slab prior that use an alternative formulation, such as in (Goodfellow et al., 2012; Lázaro-gredilla and Titsias, 2011). The chosen formulation has the advantage of being easier to parametrize and had been evaluated for model selection purposes, evaluating other alternatives could be productive. Nevertheless, no matter the form of prior we use, Bayesian inference will be a computationally intensive endeavor. As such we will be limited to a relatively small number of dimensions (experiments were conducted up to 20 elements). This problem may be alleviated, for example we used a blocked Gibbs sampling technique. Still, the method's applicability in high dimensional problems seems unlikely (Castillo et al., 2014). This is a disadvantage of Bayesian methods in general. Still the advantages offered in terms of data integration and uncertainty quantification are exploited extensively in our experimental design framework.

We chose Gibbs sampling as it was very amenable to our conjugate model. Still some issues related to convergence are present, we chose to set a fixed number of samples based in empirical results, in order to have a general method that could be run in different situations. Some cases showed a non-convergence for a few elements of matrix $\mathbf{H}$ (when similarity scores are included into the model), methodologies to alleviate this include averaging over many chains. We opted against it for performance, as we preferred a good enough solution for general cases allowing for fine tuning the number of samples for particular cases that may require more exact results, and even then, convergence is not guaranteed given the multi-modal nature of the problem.

Approximate inference techniques, such as variational inference (Bishop, 2001) or more sophisticated approximations may allow us to deal with bigger systems.

Optimising the code could also be possible and provide performance benefits.

Another important consideration is the coarseness of the noise model. The trade offs of adding a more sophisticated noise model instead of the normality assumptions have to be explored. It seems tempting to conclude that a better noise model will imply more accurate results, but this may imply an intractable inference problem. A simpler modification would be to increase the number of noise variance parameters (having a variance parameter for each time series, or even a noise parameter for each gene). In this case the model would remain conjugate, but the parameter space would increase, and thus MCMC convergence may suffer.

The parameters for the additive clustering model were estimated using the nonnegative least square estimate. Even though the use of point estimates for nuisance parameters is a known machine learning technique, it is desirable to explore alternatives, including more sophisticated sampling techniques.

Additionally, the additive clustering model used to represent promoter sequence similarities is overly simplistic. It does not account for any of the biological and stochastic processes involved in transcription factor binding. A probabilistic model that accounts for these factors may improve the clustering of related nodes in the network. Additionally, an approach such as consensus clustering (Monti et al., 2003), may well be an interesting option for integrating clusters derived from different data sources into the DSS model.

5.2 Experimental design considerations

We used DSS to propose an experimental design formulation to assess the utility of some common experimental settings. We are able to exploit the linearity of the DSS model, along with its latent network structure, for deriving an experimental assessment criterion. The proposed scheme is able to obviate the simulation of a system of coupled ODE, by sampling from a posterior probability distribution.

We proposed an experiment-by-experiment assessment. That is, we can only evaluate the utility of the next experiment. A way to assess the utility of a sequence of experiments along with the cost of each experiment, could lead to the development of an effective experimental program.

The selection criteria is based on maximizing the mutual information between the experimental outputs and the parameters to be estimated. Another informa-

tion related criteria may be adapted depending on the experiments' objectives. Diverse Bayesian experimental design criteria, for example the ones presented in (Chaloner and Verdinelli, 1995; Kreutz and Timmer, 2009) exist and could be adapted straightforwardly in the DSS model according to the experimental objectives.

The relation between the usefulness of an experiment and the utility function proves to be more subtle than one could deduce. As the utility function only accounts for the reduction in uncertainty about the linear system, the researcher has to assess what's the potential knowledge to be gained.

Conversely an experimenter can propose a given experiment under our modeling assumptions and measure the potential information gain after the experiment he proposes in order to give more weight to an hypothesis. Still, he has to follow the same consideration, some relevant components of the osccillatory system may be found by executing the experiments that will yield the highest estimated uncertainty.

5.3 Impact and future perspective

The research in this thesis has led to novel developments that promise to lead to better statistical tools for analysing oscillatory biological systems. The devised method offers promising properties as part of a exploratory step, in which we have a set of oscillating components and wish to derive a candidate network from this. Another applicability is to identify potential targets of a biological oscillator straightforwardly.

From the methodological point of view we showed that using all prior information available can improve our learning task, but it will always depend on the nature of our system and measurements made. This uncertainty and error has to be included in any experimental design setting. If the model is flexible and principled enough, like DSS, many useful properties will be exploitable. In our case it allowed us to simulate a dynamical system by drawing from a probability distribution instead of solving a great number of systems of ODE.

Short term plans are devoted to implement jDSS and jJump3 as a service in Biodare. This would allow scientist to execute these sophisticated algorithms online and improve analysis and collaboration. For this task a re-implementation of Jump3 and DSS into Java is essential to improve performance and service,

presumably a high demand.

It is our contention that integrating the frequency domain approach to a HRM-like framework offers an attractive prospect. For this, alternative frequency domain representations of a signal have to be explored, and find one that is amenable. Additionally an efficient inference algorithm has to be derived in order to provide a useful service for researchers.

Finally, as new techniques are being discovered and evolve, the integration of new data sources is essential. How to integrate this data, and which experiments to perform in order to yield the most useful data should be the main objectives for expanding on the presented work.

Appendix A

Inference algorithms for the Biodare repository

Biodare (Zielinski et al., 2014) is an online data repository for biological time series data and its main focus is circadian oscillators. Each data set contains details about the experiment authors and the experimental conditions involved in the data acquisition. It also provides six period estimation algorithms and various tools for data aggregation and transformation (Zielinski et al., 2014).

The experimental data and accompanying meta-data is stored in XML and then transformed into a native data representation using XLST. It is designed for easy data sharing, processing and analysis in circadian research. It is a promising tool to help enhance collaboration between research laboratories. Part of its objectives is to integrate more sophisticated methods for data analysis.

In this chapter we develop a stand-alone application that uses exported Biodare data sets. This data is processed and then is evaluated using any of two possible statistical inference algorithms. These algorithms aim at reconstructing genetic regulatory networks from gene expression data, (Huynh-Thu and Sanguinetti, 2015) and (Trejo Banos et al., 2015a) .

We will first present the HRM algorithm that reconstructs promoter levels and predicts gene expression levels under varying experimental conditions using a discrete-continuous hybrid stochastic model (Ocone et al., 2013). The stochastic modeling framework of HRM is the basis for the non-parametric method for network inference Jump3 (Huynh-Thu and Sanguinetti, 2015). Then the DFT-based spike and slab model for network inference DSS presented in Chapter 3 is briefly overviewed.

The last part of the chapter explains the architectural design for the applications as well as a brief documentation of the programs. This results in the stand alone but Biodare compatible applications jJump3 and jDSS.

A.1 The methods

The following methods were developed using statistical tools for biological systems modeling. HRM and Jump3 are formulated in a Stochastic Differential Equation (SDE) framework. DSS is a deterministic approach to networks dynamic using a system of Ordinary Differential Equations ODE. We will first introduce HRM as the basis for Jump3.

A.1.1 HRM

HRM is a framework for modeling genetic regulatory networks through a hybrid continuous/discrete stochastic process (Ocone et al., 2013). In this model, the promoters (transcription factor bindings) are modeled as a binary variables (occupied or unoccupied).

The states are denoted as μ, the probability of $\mu = 1$ at time t is denoted as $p_1(t)$ and $p_0(t)$ for the complementary case. These marginal probabilities obey the chemical master equation, see (Ocone et al., 2013), as:

$$\begin{aligned} \frac{dp_1(t)}{dt} &= -f_-(t)\,p_1(t) + f_+(t)\,p_0(t) \\ \frac{dp_0(t)}{dt} &= -f_+(t)\,p_0(t) + f_-(t)\,p_1(t) \end{aligned}$$

where the functions f_- and f_+ represent the probability of changing from one state to the other per unit of time. These functions are modeled as a loglinear function of the transcription factor concentrations and a positive constant respectively. The production of protein x_i is then modeled as the stochastic differential equation:

$$dx_i = (A_i \mu_i(t) + b_i - \lambda_i x_i)\,dt + \sigma dw(t) \tag{A.1}$$

where A_i is the efficiency of promoter i to recruit polymerase when occupied, b_i the basal transcriptional-translational rate and the exponential decay constant λ_i.

The term $\sigma dw(t)$ represents a noise process with variance σ^2. This modeling framework accounts for the intrinsic and extrinsic noise in gene expression (Ocone et al., 2013).

By applying approximate inference techniques, HRM by (Ocone et al., 2013) can estimate the conditional probability of the promoter states μ, kinetic parameters A, b and λ along with the protein levels x, given a set of observations over the gene expression levels, denoted as $\mathbf{y}$.

A.1.2 JUMP3

Jump3 (Huynh-Thu and Sanguinetti, 2015) is a gene regulatory network inference method whose basic principle can also be traced to the SDE in Eq.(A.1). Supposing that we are given the states $\mu_i(t)$, the solution to Eq.(A.1) is equivalent to a Gaussian Markov process whose mean function is denoted as $m_i(t)$ and covariance function $c_i\left(t,t^{'}\right)$ are such that

$$\begin{aligned} m_i(t) &= x_i(0)e^{-\lambda_i t} + A_i\int_0^t e^{-\lambda_i(t-\tau)}\mu_i(\tau)\,d\tau + \frac{b_i}{\lambda_i}\left(1-e^{-\lambda_i t}\right) \\ c_i\left(t,t'\right) &= \frac{\sigma^2}{2\lambda_i}\left(e^{-\lambda_i|t-t'|} - e^{-\lambda_i(t+t')}\right). \end{aligned}$$

Additionally the gene expression levels are observed with a normal i.i.d error of variance $s^2_{i,k}$, where the subindex k represents the $k-th$ observed time point. Then the observed gene expression levels for gene i (denoted as $\hat{\mathbf{x}}_i$) follow a multivariate normal distribution as

$$\hat{\mathbf{x}}_i \sim \mathcal{N}(\mathbf{m}_i, C_i + D_i)$$

where $\mathbf{m}_i = [m_i(t_1), m_i(t_2), ..., m_i(t_N)]^T$, C_i is the covariance matrix with elements $c_i(t_k,t_l)$ and D_i is a diagonal matrix with values $s^2_{i,k}$.

Then Jump3 jointly solves for each target gene i:

- Identify the promoter state trajectory μ_i that maximizes the likelihood $\log p(\hat{\mathbf{x}}_i)$.
- Identify the genes that influence μ_i, these genes are the regulators of gene i and as such, are the parent nodes of gene i in the genetic regulatory network.

The regression over the gene expression levels is performed using random forests, a machine learning algorithm based on averaging a set of decision trees (Huynh-Thu and Sanguinetti, 2015).

A.1.3 DSS

DSS is a network inference method presented in Chapter 3. First the gene expression levels are transformed into a set of coefficients for sinusoidal basis functions using the DFT. The frequency spectrum is calculated for all the expression levels of the putative components of a genetic regulatory network.

The spectra are collated in the same matrix $\mathbf{X}^q$, where the q indexes an experiment. By working in Fourier domain we are able to compute the derivative of the time domain signals by simply multiplying their frequency domain counter parts by a constant factor $\mathbf{D}$. Then the linearized dynamics of the network are represented by the matrix equation

$$\dot{\mathbf{X}}^q = \mathbf{X}^q\mathbf{A}^T + \mathbf{U}^q\mathbf{C}^T + \mathbf{R}^q \tag{A.2}$$

where $\dot{\mathbf{X}}^q = \mathbf{D}\mathbf{X}^q$ is the derivative matrix, $\mathbf{U}^q$ is the frequency spectra of the systems inputs (for example a light signal), parameters $[\mathbf{A}, \mathbf{C}]$ are the interaction coefficients (interactions in the network) and $\mathbf{R}^q \in \mathbb{R}^{M\times N}$ is the residual matrix.

DSS proposes a spike and slab prior distribution for parameters $\{\mathbf{A}, \mathbf{B}\}$. This prior is used along with a likelihood function over the residuals to infer the posterior distribution over these parameters given a set of observed spectra using a Gibbs sampling scheme (Trejo Banos et al., 2015a).

A.2 Implementation and deployment

Our aim was to implement these three algorithms as part of a stand-alone application. Additionally we wish the application to be fully integrated as part of Biodare in future iterations. Biodare is mainly developed in java, and as such a java-based solution is desirable. Thus we decided to re-implement these programs into JAVA programming language.

A java-based version of HRM (which we will call jHRM) was kindly provided by its authors, for both JUMP3 and DSS only Matlab versions were available. Having these settings we faced three main challenges:

1. Decoupling the user interface from the algorithm in jHRM into an architecture that will ease its eventual integration as a Biodare service.

2. Generating java code from the Matlab sources of JUMP3 and DSS.

3. Re-engineer Jump3 and DSS in order to integrate them seamlessly into a java application.

4. Build the java based applications jJump3 and jDSS by developing wrappers for data conversion around the compiled-to-java Matlab sources.

Given that the process for Jump3 and DSS is almost identical, we will only describe into detail the former.

A.3 jJump3

We started by compiling Jump3 from its Matlab version developed by (Huynh-Thu and Sanguinetti, 2015). We used Mathworks tool Java Builder, which can compile Matlab functions into a Java library. This library contains binary files for all the compiled functions. Eventually we aim at having this program completely re implemented in java language. For this moment we are going to provide a group of classes to receive data from the graphical user interface (or Biodare) and execute the Jump3 algorithm. These classes will compose what we call jJump3, and are designed in such a way that a java re-implementation of Jump3 will be straightforward to plug in.

An illustrative view of the architecture in form of package diagram is presented in Fig.A.1. Here the same package contains the necessary Mathworks libraries for Matlab-to java compilation and execution along with the Jump3 compiled code and jJump3 classes. This package will then be used by the graphical user interface. The interface searches the Biodare repository and submit the resulting time series data to jJump3 for executing Jump3. In our case the interface will import search results from Biodare stored in a text file, in which entries are delimited by commas [1].

We named the class resulting from the set of compiled Matlab functions *Jump3Functions*, each Matlab function from Jump3 gets converted into a method of this class. The main function is called *jump3*. This function is the entry point

[1] This kind of files are known as "Comma Separated Values" and have a .csv extension.

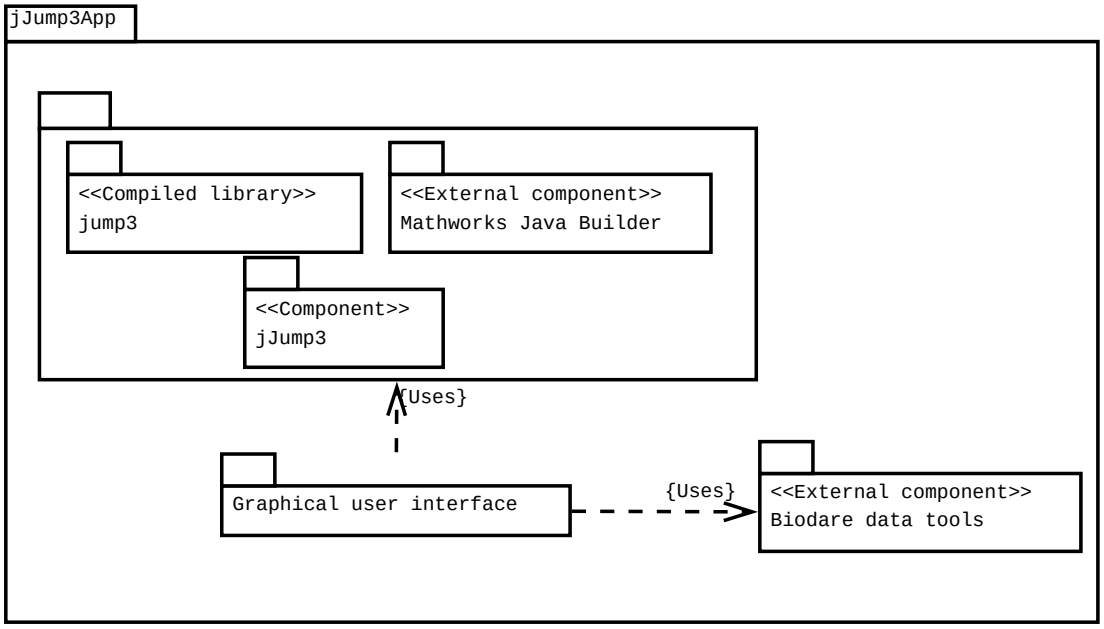


Figure A.1: Package diagram for jJump3. The application consists of a group of components for executing Jump3 and import data from Biodare. The graphical user interface will import the data from a text file (with aims at future integration into Biodare).

to the algorithm and as such our library has only to pass arguments and receive the output from this function.

The *jump3* function signature (output and input arguments) is dependent on Mathwoks Java Builder wrapper classes. These classes offer a translation from Java data structures to Matlab data structures. As these classes have to be flexible, they are "blind" to the type of data they contain. In order to adapt them for our application it is necessary to:

- Create two classes, one containing the *input* parameters and the other the *output arguments*. These classes are designed specifically for Jump3 signature, and will serve as a wrapper for the Mathworks Java Builder data structures. In this way the deployed application will not need to execute these data transformations. These classes are named *Jump3Parameters* and *Jump3Output*.

- Create a *model class*, containing the input and output parameters and the Jump3 algorithm. This will provide a flexible way to run the algorithm as an independent component from the graphical user interface. It will store

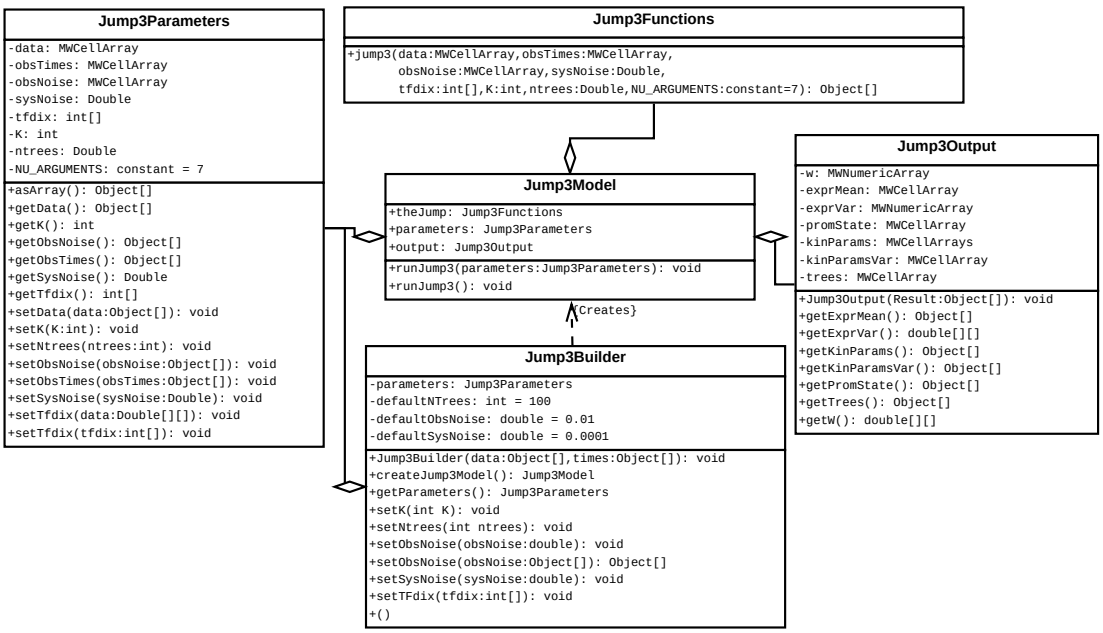


Figure A.2: Class Diagram for jJump3. A Jump3Model is parametrized by a Jump3Builder by setting the values of a Jump3Parameter object. Then the Jump3Model will use these parameters to execute the compiled-from-Matlab Jump3, and store the results into a Jump3Output object.

the parameters and output of the function Jump3. In our case we named this class as *Jump3Model*.

- Create a *builder class* that will be in charge of parametrizing the model according to the imported data. This class will be named Jump3Builder.

These elements constitute the core logic of the application, thus the input of data and parameters, along with the presentation of results can be dealt with independently. We present the class diagram of this core elements in Fig.A.2. Here we see the class Jump3Functions with the executable function Jump3 with parameters:

- *data* A collection of time series containing the gene expression levels, each row represents the expression level of each gene at each time step ($\hat{\mathbf{x}}$).
- *obsTimes* A collection of the observation times at which measurements of the gene expression levels were taken (t).
- *obsNoise* A collection containing the observation noise variance at each time step ($s^2_{i,k}$).

- *sysNoise* A parameter containing the system noise variance (σ^2).
- *tfdix* A parameter containing the indexes of the putative transcription factors.
- K A parameter containing the number of genes against which each gene expression level will be regressed.
- *nTrees* The number of random trees used for inference.

The class Jump3Builder uses the input data and observation times to preset the parameters to its default values of 0.01 observation noise variance, 0.0001 system noise variance. Additionally it presets all the genes as putative transcription factors, and parameter K equal to the number of genes. Finally nTrees is set to 100. Finally we provide a set of methods that will allow us to change these parameters before building a Jump3Model though the method *createJump3Model()*.

The class Jump3Model contains two attributes, a set of parameters of type Jump3Parameters, and a set of algorithm outputs, of type Jump3Output. The class method *runJump3()* creates a Jump3Functions object and runs Jump3 using the parameters contained in the parameter object. The Jump3Parameter object provides data conversion from Java-types to Matlab-types. Conversely the output object provide data conversion from Matlab-type to Java-type.

The outputs are:

- w A matrix containing the importance weight of each putative edge as inferred through Jump3.
- *exprMean* The inferred mean expression levels $\mathbf{m}(\mathbf{t})$.
- *exprVar* The inferred covariance matrices $\mathbf{C}$.
- *promState* The inferred promoter states μ.
- *kinParams* The inferred kinetic parameters $\{\mathbf{A}, b, \lambda\}$.
- *kinParamsVar* The variance of the inferred kinetic parameters.
- *trees* The decision trees resulting from the inference process.

These outputs are stored and converted to the required data-type through the Jump3Output object, further data transformations can be donei by extending or modifying this class.

A.3.1 Biodare data import and graphical user interface

Having the logic behind the Jump3 algorithm we focused on the data input to the system. As mentioned earlier, we wish the integration of the software into Biodare to be as smooth as possible. We designed a small architecture for reading exported data from a Biodare query (in csv format) using the Biodare internal data representation. For this we used the already developed tools provided by the authors of (Zielinski et al., 2014).

These tools are contained in the classes *TimeSeries* and *TimeSeriesFileHandler.* The static method *readFromText()* reads data in text format and returns a list of *TimeSeries* objects. Each TimeSeries is a collection of pairs, of the form *(time,value)*, allowing for time series of different length.

Each exported file from Biodare contains time series data (easily convertible to a TimeSeries object) and some meta-data. This meta-data contains information about experimental design and data characteristics. From these fields we were only interested in the experiment specification, as these will allow us to group all the time series of an experiment into a single object. As such, it was necessary to read the column labels and save the experiment id and experimental conditions. For this purpose we developed the class *Jump3Controller.* This class contains only a static method that reads a file using the TimeSeriesFileHandler and returns an object of the class *Jump3DataAdapter.*

The object Jump3DataAdapter, has the time series grouped according to the experiment specification (experimented/experimental_cond) as represented by objects of class *ExperimentalData.* So by hashing the experiment specifications, we are able to retrieve the sets of time series belonging to that experiment.

It is important to point out, that by Biodare data representation, a TimeSeries object only represents the gene expression level of a single gene over a set of time points. The data contained in the ExperimentalData class contains the gene expression levels of all genes for a single experiment. This data is used by Jump3 as an input, as shown by the relationship between the Jump3ModelBuilder and the Jump3DataAdapter in the class diagram of Fig.A.3.

Finally we developed a simple graphical user interface. This is a prototype for testing how well the elements interact with the aim of a future web-based solution. The windows are shown in Fig.A.4 where we see the application after some data has been loaded using the “Import Biodare data” button, the algo-

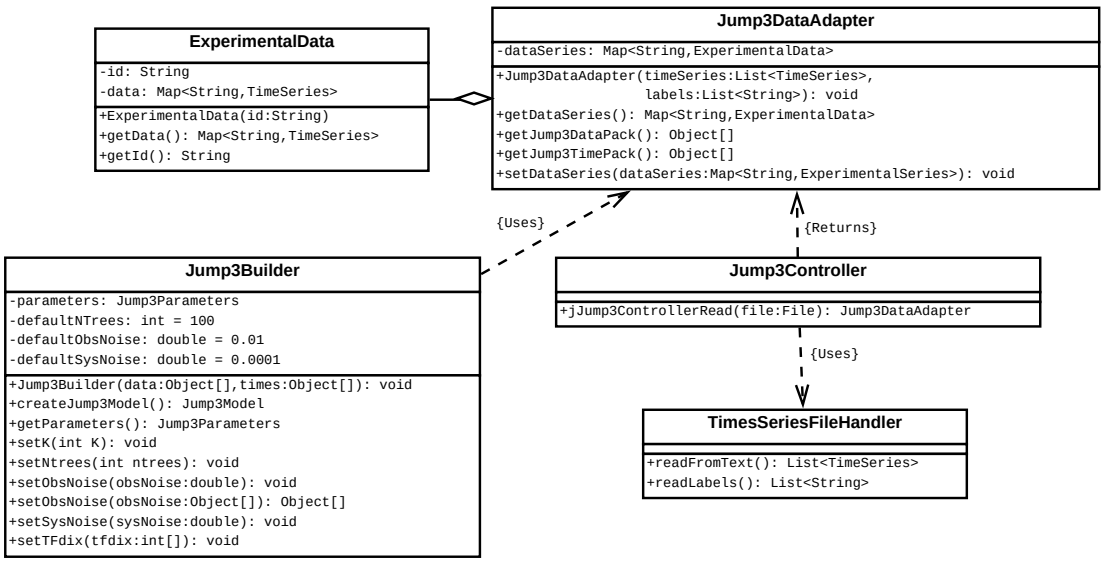


Figure A.3: Class diagram for Biodare data integration with jJump3. The Jump3Controller reads a text file and returns a Jump3DataAdapter, with the data grouped and hashed by experiment ID. This data is then passed to the Jump3Builder for model parametrization and posterior execution of Jump3.

rithm's parameters are now adjustable using standard text fields. Fig.A.5 shows the method execution after pressing the "Run Jump3" button, with a text area showing the current output of the method as it is being executed. Once the execution is finished the results are exported using the "Export results button". The results are exported in a text format, ready to be used as input to a plotter or network analysis software (such as cytoscape).

A.4 jDSS

We proceeded to implement DSS by following the same guidelines as for jJump3. We compiled the DSS matlab code into a Java library. Then we designed the application architecture following the same package structure as in Fig.A.1. Here the DSS package takes the place of the Jump3 package, with the Graphical User interface being adapted to the DSS parameters and the Biodare package remains unchanged.

Analogously to jJump3 we present the class diagram of jDSS core elements in Fig.A.6. DSS will be executed with parameters

- *data* A collection of time series containing the gene expression levels, each

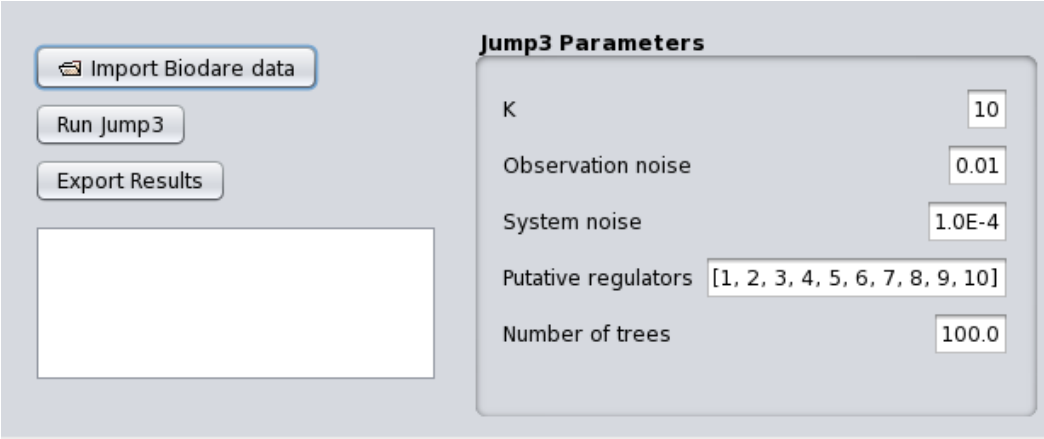


Figure A.4: Jump3 application with the adjustable parameters.

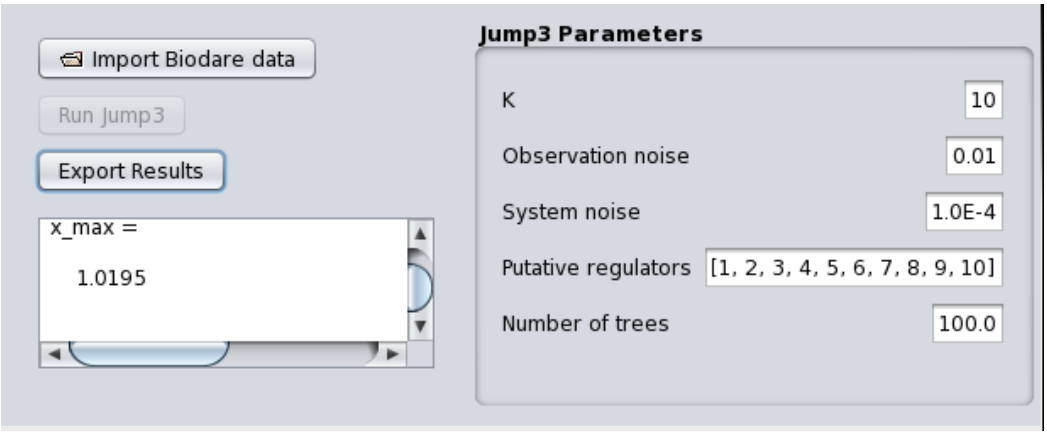


Figure A.5: Jump3 application during the execution of the Jump3 method.

column represents the expression level of at each time step (matrix $\mathbf{x}$).

- M The number of rows of data.
- N The number of genes to be used as putative members of the network.
- *Inputs* The subrange between 1 and N containing the indexes of the columns to be considered as input signals. These inputs can represent either protein levels or light inputs for example.
- S It's the N by N matrix containing the similarity scores for promoter regions. DSS will execute without the region similarity clustering in case of empty matrix S.
- *Spectra* RDFT spectra to be used, with options:
  - FFTUnorm: DFT coefficients computed by FFT and RDFT computed by stacking real and imaginary parts of the first M/2 coefficients.
  - FFTNorm: DFT computed by FFT and then normalized (divided by M). Then the RDFT is computed.
- *SamplerParameters* Structure containing the Gibbs sampling parameters of the DSS model:
  - *Samples* The number of samples, default is 5000 samples with 4000 samples of burn-in.
  - *V0* Parameter that determines the *"width"* of the spike. It must be a value close to zero, but not zero. Default value is 0.005.
  - *A1,A2* Parameters controlling the Gamma distribution of the spike and slab parameter τ^{-2}. Default values are (5,0.001).
  - *B1,B2* Parameter for the Beta distribution over the sparsity parameter w. Default values are (1,1).
  - *C1,C2* Parameters for the Gamma distribution over parameter σ^{-2}. Default values are (1,0.001).
  - D1,D2 Parameters for the Gamma distribution over the sequence similarity model parameter σ_{seq}^{-2}. Default values are (100,0.001).

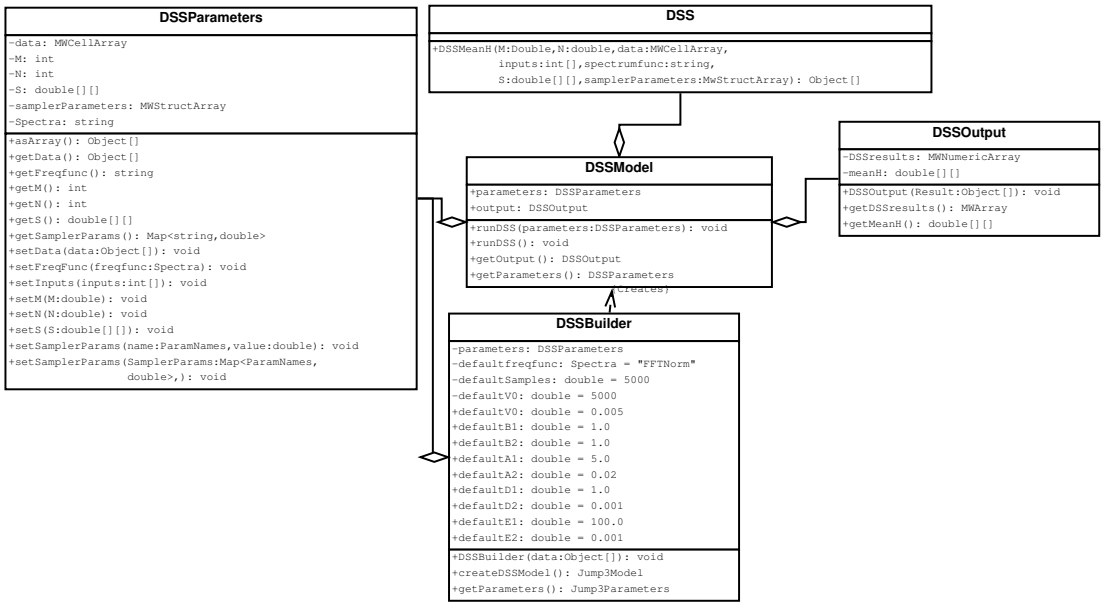


Figure A.6: Class Diagram for jDSS. A DSSModel is parametrized by a DSSBuilder by setting the values of a DSSParameter object. Then the DSSModel will use these parameters to execute the compiled-from-Matlab DSS, and store the results into a DSSOutput object.

The classDSSBuilder computes parameters M and N from the input data, and sets all other parameters to their default values. We provide accessors to these parameters, allowing them to be modified, and then we build a DSS model by invoking the method *createDSSModel().*

The class DSS contains the DSSParameters and the DSSOutput objects. The class method *runDSS()* creates a DSS object and runs DSS with the DSSParameter object as a converter between Java and Matlab data structures. The output is

- *meanH* A matrix containing the average values for each link in matrix H among the samples. These values are in the range 0-1 and represent the probability of a link existing according to the observed data and the sampler parameters.

As with jJump3, a DSSDataAdapter object will group the time series read from the Biodare data files. In case of DSS, time series are grouped in columns. The columns are stacked into matrices, each matrix representing an experiment. Now DSSBuilder will build a model according to this data pack, ready for the execution of DSS. The class diagram of this construction is shown in Fig.A.7.

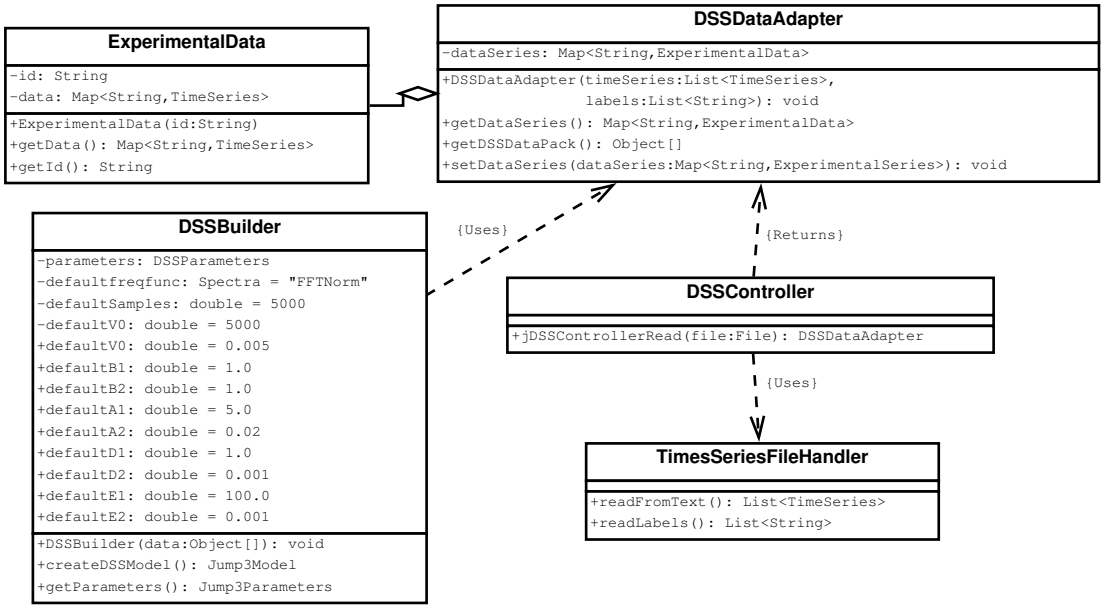


Figure A.7: Class diagram for Biodare data integration with jDSS. The DSSController reads a text file and returns a DSSDataAdapter, with the data grouped and hashed by experiment ID. This data is then passed to the DSSBuilder for model parametrization and posterior execution of DSS.

A.5 Discussion

We developed a couple applications for performing different types of inference over Biodare data sets. These applications are simple to use and depend on exported data from the Biodare repository. A direct integration with Biodare may prove to be a productive tool for researches everywhere, and thus it is a desirable outcome for further developments.

From the design point of view, we developed a fairly simple but general architecture that can be adapted to the two scenarios. Even though we modularized functions in order to allow separation of concerns, we think that a more general framework can still be designed. If more inference methods are to be deployed as part of Biodare, the use of an abstract class or an interface to encompass all inference models may hold a lot of potential. In other words, by having all the inference methods follow a same set of "rules" of execution and structure, the deployment speed of new methods may be accelerated. This would allow software engineers in charge of Biodare and the method developers to work in parallel for tool development.

Bibliography

Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P. (2002). *Molecular biology of the cell. 4th edn.* Garland Science.

Bähler, J. (2005). Cell-cycle control of gene expression in budding and fission yeast. *Annu. Rev. Genet.*, 39:69–94.

Barber, D. (2012). *Bayesian reasoning and machine learning.* Cambridge University Press.

Bell-Pedersen, D., Cassone, V. M., Earnest, D. J., Golden, S. S., Hardin, P. E., Thomas, T. L., and Zoran, M. J. (2005). Circadian rhythms from multiple oscillators: lessons from diverse organisms. *Nature Reviews Genetics*, 6(7):544–556.

Bergersen, L. C., Glad, I. K., and Lyng, H. (2011). Weighted Lasso with Data Integration. *Statistical Applications in Genetics and Molecular Biology*, 10(1):1–29.

Bishop, C. M. (2001). *Pattern recognition and Machine learning.* Springer Verlag.

Bonneau, R., Reiss, D. J., Shannon, P., Facciotti, M., Hood, L., Baliga, N. S., and Thorsson, V. (2006). The Inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. *Genome biology*, 7(5):R36.

Callier, F. M. and Desoer, C. A. (2012). *Linear system theory.* Springer Science & Business Media.

Castillo, I., Schmidt-Hieber, J., and van der Vaart, A. W. (2014). Bayesian linear regression with sparse priors. *arXiv preprint arXiv:1403.0735*.

Chaloner, K. and Verdinelli, I. (1995). Bayesian Experimental Design: A Review. *Statist. Sci.*, (3):273–304.

Charbonnier, C., Chiquet, J., and Ambroise, C. (2010). Weighted-LASSO for structured network inference from time course data. *Statistical applications in genetics and molecular biology*, 9(1):15.

Chen, K. C., Calzone, L., Csikasz-Nagy, A., Cross, F. R., Novak, B., and Tyson, J. J. (2004). Integrative analysis of cell cycle control in budding yeast. *Molecular biology of the cell*, 15(8):3841–3862.

Chu, Y. and Corey, D. R. (2012). Rna sequencing: Platform selection, experimental design, and data interpretation. *Nucleic Acid Ther*, 22(4):271–274. 22830413[pmid].

Collas, P. (2009). The state-of-the-art of chromatin immunoprecipitation. 567.

Cooley, J. W. and Tukey, J. W. (1965). An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90):297–301.

Cowles, M. K. and Carlin, B. P. (1996). Markov chain Monte Carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association*, 91(434):883–904.

Dalchau, N. (2012). Understanding biological timing using mechanistic and black-box models. *New Phytologist*, 193(4):852–858.

De Jong, H., Gouzé, J.-L., Hernandez, C., Page, M., Sari, T., and Geiselmann, J. (2004). Qualitative simulation of genetic regulatory networks using piecewise-linear models. *Bulletin of mathematical biology*, 66(2):301–340.

de Souza, N. (2012). Single-cell methods. *Nat Meth*, 9(1):35–35. Method to Watch.

DeJesus, M. and Ioerger, T. (2013). A Hidden Markov Model for identifying essential and growth-defect regions in bacterial genomes from transposon insertion sequencing data. *BMC Bioinformatics*, 14(1):303.

Ding, Z., Doyle, M. R., Amasino, R. M., and Davis, S. J. (2007). A complex genetic interaction between arabidopsis thaliana toc1 and cca1/lhy in driving the circadian clock and in output regulation. *Genetics*, 176(3):1501–1510.

Dodd, A. N., Salathia, N., Hall, A., Kévei, E., Toth, R., Nagy, F., Hibberd, J. M., Millar, A. J., and Webb, A. A. R. (2005). Plant circadian clocks increase photosynthesis, growth, survival, and competitive advantage. *Science*, 309(5734):630–633.

Dondelinger, F., Husmeier, D., Rogers, S., and Filippone, M. (2013a). Ode parameter inference using adaptive gradient matching with Gaussian processes. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, pages 216–228.

Dondelinger, F., Lèbre, S., and Husmeier, D. (2013b). Non-homogeneous dynamic Bayesian networks with Bayesian regularization for inferring gene regulatory networks with gradually time-varying structure. *Machine Learning*, 90(2):191–230.

Dornbusch, T., Michaud, O., Xenarios, I., and Fankhauser, C. (2014). Differentially phased leaf growth and movements in arabidopsis depend on coordinated circadian and light regulation. *The Plant Cell*, 26(10):3911–3921.

Edwards, K. D., Akman, O. E., Knox, K., Lumsden, P. J., Thomson, A. W., Brown, P. E., Pokhilko, A., Kozma-Bognar, L., Nagy, F., Rand, D. A., and Millar, A. J. (2010). Quantitative analysis of regulatory flexibility under changing environmental conditions. *Molecular Systems Biology*, 6(1).

Ersoy, O. (1985). Real discrete Fourier transform. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 33(4):880–882.

Eser, P., Demel, C., Maier, K. C., Schwalb, B., Pirkl, N., Martin, D. E., Cramer, P., and Tresch, A. (2014). Periodic mrna synthesis and degradation co-operate during cell cycle gene expression. *Molecular Systems Biology*, 10(1).

Faith, J. J., Hayete, B., Thaden, J. T., Mogno, I., Wierzbowski, J., Cottarel, G., Kasif, S., Collins, J. J., and Gardner, T. S. (2007). Large-Scale Mapping and Validation of Escherichia coli Transcriptional Regulation from a Compendium of Expression Profiles. *PLoS Biology*, 5(1):e8.

Fan, F. and Wood, K. V. (2007). Bioluminescent assays for high-throughput screening. *ASSAY and Drug Development Technologies*, 5(1):127–136.

Geweke, J. (1992). Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments. In *IN BAYESIAN STATISTICS*, pages 169–193. University Press.

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361.

Gomez-Cabrero, D., Abugessaisa, I., Maier, D., Teschendorff, A., Merkenschlager, M., Gisel, A., Ballestar, E., Bongcam-Rudloff, E., Conesa, A., and Tegner, J. (2014). Data integration in the era of omics: current and future challenges. *BMC Systems Biology*, 8(Suppl 2):I1.

Goodfellow, I., Courville, A., and Bengio, Y. (2012). Large-scale feature learning with spike-and-slab sparse coding. *arXiv preprint arXiv:1206.6407*.

Gottardo, R. (2009). Modeling and analysis of chip-chip experiments. In *Chromatin Immunoprecipitation Assays*, pages 133–143. Springer.

Greenfield, A., Hafemeister, C., and Bonneau, R. (2013). Robust data-driven incorporation of prior knowledge into the inference of dynamic regulatory networks. *Bioinformatics*, 29(8):1060–1067.

Haase, S. B. and Wittenberg, C. (2014). Topology and Control of the Cell-Cycle-Regulated Transcriptional Circuitry. *Genetics*, 196(1):65–90.

Haury, A.-C., Mordelet, F., Vera-Licona, P., and Vert, J.-P. (2012). TIGRESS: trustful inference of gene regulation using stability selection. *BMC systems biology*, 6(1):145.

Huynh-Thu, V. A., Irrthum, A., Wehenkel, L., and Geurts, P. (2010). Inferring regulatory networks from expression data using tree-based methods. *PLoS ONE*, 5(9):e12776.

Huynh-Thu, V. A. and Sanguinetti, G. (2015). Combining tree-based and dynamical systems for the inference of gene regulatory networks. *Bioinformatics*, 31(10):1614–1622.

Ishwaran, H. and Rao, J. S. (2005). Spike and slab variable selection frequentist and Bayesian strategies. *The Annals of Statistics*, 33(2):730–773.

Karlebach, G. and Shamir, R. (2008). Modelling and analysis of gene regulatory networks. *Nature Reviews Molecular Cell Biology*, 9(10):770–780.

Kitano, H. (2002). Systems biology a brief overview. *Science*, 295(5560):1662–1664.

Klipp, E. (2005). *ystems Biology In Practice: Concepts, Implementation And Application*. Wiley-VCH.

Kreutz, C. and Timmer, J. (2009). Systems biology: experimental design. *FEBS Journal*, 276(4):923–942.

Kuffner, R., Petri, T., Tavakkolkhah, P., Windhager, L., and Zimmer, R. (2012). Inferring gene regulatory networks by ANOVA. *Bioinformatics*, 28(10):1376–1382.

Kullback, S. (1968). *Information theory and statistics*. Courier Corporation.

Kumar, R., Reynolds, D. M., Shevchenko, A., Shevchenko, A., Goldstone, S. D., and Dalton, S. (2000). Forkhead transcription factors, Fkh1p and Fkh2p, collaborate with Mcm1p to control transcription required for M-phase. *Current Biology*, 10(15):896–906.

Lawrence, N. D., Girolami, M., Rattray, M., and Sanguinetti, G. (2009). Learning and inference in computational systems biology.

Lázaro-gredilla, M. and Titsias, M. K. (2011). Spike and slab variational inference for multi-task and multiple kernel learning. In *Advances in neural information processing systems*, pages 2339–2347.

Li, F., Long, T., Lu, Y., Ouyang, Q., and Tang, C. (2004). The yeast cell-cycle network is robustly designed. *Proceedings of the National Academy of Sciences of the United States of America*, 101(14):4781–4786.

Li, F., Yang, Y., and Xing, E. (2006). *Inferring regulatory networks using a hierarchical Bayesian graphical Gaussian model*. Carnegie Mellon University, School of Computer Science, Machine Learning Department.

Liepe, J., Filippi, S., Komorowski, M., and Stumpf, M. P. H. (2013). Maximizing the information content of experiments in systems biology. *PLoS Computational Biology*, 9(1):e1002888.

Lindley, D. V. (1956). On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics.*, (4):986–1005.

Liu, J. S. (1994). The Collapsed Gibbs Sampler in Bayesian Computations with Applications to a Gene Regulation Problem. *Journal of the American Statistical Association*, 89(427):958–966.

Logan, J. M., Edwards, K. J., and Saunders, N. A. (2009). *Real-time PCR: current technology and applications*. Horizon Scientific Press.

Loy, C. J., Lydall, D., and Surana, U. (1999). NDD1, a High-Dosage Suppressor ofcdc28-1n, Is Essential for Expression of a Subset of Late-S-Phase-Specific Genes in Saccharomyces cerevisiae. *Molecular and cellular biology*, 19(5):3312–3327.

MacIsaac, K. D., Wang, T., Gordon, D. B., Gifford, D. K., Stormo, G. D., and Fraenkel, E. (2006). An improved map of conserved regulatory sites for Saccharomyces cerevisiae. *BMC bioinformatics*, 7(1):113.

MacKay, D. J. (2003). *Information theory, inference and learning algorithms*. Cambridge university press.

Marbach, D., Prill, R. J., Schaffter, T., Mattiussi, C., Floreano, D., and Stolovitzky, G. (2010). Revealing strengths and weaknesses of methods for gene network inference. *Proceedings of the National Academy of Sciences*, 107(14):6286–6291.

Margolin, A. A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Favera, R., and Califano, A. (2006). ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context. *BMC Bioinformatics*, 7(Suppl 1):S7.

McClung, C. R. (2011). Chapter 4 - The Genetics of Plant Clocks. In Stuart Brody, editor, *Advances in Genetics*, volume Volume 74, pages 105–139. Academic Press.

Mirkin, B. G. (1987). Additive clustering and qualitative factor analysis methods for similarity matrices. *Journal of Classification*, 4(1):7–31.

Monti, S., Tamayo, P., Mesirov, J., and Golub, T. (2003). Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine learning*, 52(1-2):91–118.

Morrissey, E. R., JuÃ¡rez, M. A., Denby, K. J., and Burroughs, N. J. (2011). Inferring the time-invariant topology of a nonlinear sparse gene regulatory network using fully bayesian spline autoregression. *Biostatistics*, 12(4):682–694.

Nussinov, R. (2015). Advancements and challenges in computational biology. *PLoS computational biology*, 11(1):e1004053.

Oates, C. J. and Mukherjee, S. (2012). Network inference and biological dynamics. *The Annals of Applied Statistics*, 6(3):1209–1235.

Ocone, A., Millar, A. J., and Sanguinetti, G. (2013). Hybrid regulatory models: a statistically tractable approach to model regulatory network dynamics. *Bioinformatics*, 29(7):910–916.

Orlando, D. A., Lin, C. Y., Bernard, A., Wang, J. Y., Socolar, J. E. S., Iversen, E. S., Hartemink, A. J., and Haase, S. B. (2008). Global control of cell-cycle transcription by coupled CDK and network oscillators. *Nature*, 453(7197):944–947.

Ostrow, A. Z., Nellimoottil, T., Knott, S. R., Fox, C. A., Tavare, S., and Aparicio, O. M. (2014). Fkh1 and fkh2 bind multiple chromosomal elements in the s. cerevisiae genome with distinct specificities and cell cycle dynamics.

Pavlopoulos, G. A., Secrier, M., Moschopoulos, C. N., Soldatos, T. G., Kossida, S., Aerts, J., Schneider, R., Bagos, P. G., et al. (2011). Using graph theory to analyze biological networks. *BioData mining*, 4(10):1–27.

Pintelon, R. and Schoukens, J. (2012). *System identification a frequency domain approach.* Hoboken, N.J. : Wiley ; Piscataway, NJ : IEEE Press, c2012.

Pokhilko, A., Fernandez, A. P. a., Edwards, K. D., Southern, M. M., Halliday, K. J., and Millar, A. J. (2012). The clock gene circuit in Arabidopsis includes a repressilator with additional feedback loops. *Molecular Systems Biology*, 8.

Pokhilko, A., Hodge, S. K., Stratford, K., Knox, K., Edwards, K. D., Thomson, A. W., Mizuno, T., and Millar, A. J. (2010). Data assimilation constrains new

connections and components in a complex, eukaryotic circadian clock model. *Molecular Systems Biology*, 6.

Polynikis, A., Hogan, S., and di Bernardo, M. (2009). Comparing different ode modelling approaches for gene regulatory networks. *Journal of theoretical biology*, 261(4):511–530.

Pramila, T. (2006). The Forkhead transcription factor Hcm1 regulates chromosome segregation genes and fills the S-phase gap in the transcriptional circuitry of the cell cycle. *Genes & Development*, 20(16):2266–2278.

Pukelsheim, F. (2006). *Optimal Design of Experiments*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics.

Shim, J. S. and Imaizumi, T. (2015). Circadian clock and photoperiodic response in arabidopsis: From seasonal flowering to redox homeostasis. *Biochemistry*, 54(2):157–170. PMID: 25346271.

Sievers, F., Wilm, A., Dineen, D., Gibson, T. J., Karplus, K., Li, W., Lopez, R., McWilliam, H., Remmert, M., Söding, J., Thompson, J. D., and Higgins, D. G. (2011). Fast, scalable generation of high quality protein multiple sequence alignments using Clustal Omega. *Molecular Systems Biology*, 7(1).

Sims, G. E., Jun, S.-R., Wu, G. A., and Kim, S.-H. (2009). Alignment-free genome comparison with feature frequency profiles (FFP) and optimal resolutions. *Proceedings of the National Academy of Sciences of the United States of America*, 106(8):2677–2682.

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Brown, P. O., Botstein, D., and Futcher, B. (1998). Comprehensive identification of cell cycle regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Molecular biology of the cell*, 9(12):3273–3297.

Stillman, D. J. (2013). Dancing the cell cycle two-step: regulation of yeast g1-cell-cycle genes by chromatin structure. *Trends in biochemical sciences*, 38(9):467–475.

Stone, L. and He, D. (2007). Chaotic oscillations and cycles in multi-trophic ecological systems. *Journal of Theoretical Biology*, 248(2):382–390.

Tibshirani, R. (1994). Regression Shrinkage and Selection Via the Lasso. volume 58, pages 267–288.

Trejo Banos, D., Millar, A. J., and Sanguinetti, G. (2015a). A Bayesian approach for structure learning in oscillating regulatory networks. *Bioinformatics*.

Trejo Banos, D., Millar, A. J., and Sanguinetti, G. (2015b). Experimental design for inference over the a. thaliana circadian clock network. In *Computational Methods in Systems Biology*. Springer.

Tu, Y., Stolovitzky, G., and Klein, U. (2002). Quantitative noise analysis for gene expression microarray experiments. *Proceedings of the National Academy of Sciences*, 99(22):14031–14036.

Turner, J., Ewald, J., and Skotheim, J. (2012). Cell Size Control in Yeast. *Current Biology*, 22(9):R350–R359.

Van Leeuwen, W., Hagendoorn, M. J., Ruttink, T., Van Poecke, R., Van Der Plas, L. H., and Van Der Krol, A. R. (2000). The use of the luciferase reporter system for in planta gene expression studies. *Plant Molecular Biology Reporter*, 18(2):143–144.

Vandepoele, K., Quimbaya, M., Casneuf, T., De Veylder, L., and Van de Peer, Y. (2009). Unraveling Transcriptional Control in Arabidopsis Using cis-Regulatory Elements and Coexpression Networks. *PLANT PHYSIOLOGY*, 150(2):535–546.

Venters, B. J., Wachi, S., Mavrich, T. N., Andersen, B. E., Jena, P., Sinnamon, A. J., Jain, P., Rolleri, N. S., Jiang, C., Hemeryck-Walsh, C., and Pugh, B. F. (2011). A comprehensive genomic binding map of gene and chromatin regulatory proteins in Saccharomyces. *Molecular Cell*, 41(4):480–492.

Wasserman, L. (2013). *All of statistics: a concise course in statistical inference*. Springer Science & Business Media.

Weiss, E. L. (2012). Mitotic exit and separation of mother and daughter cells. *Genetics*, 192(4):1165–1202.

Zhu, J. and Zhang, M. Q. (1999). Scpd: a promoter database of the yeast saccharomyces cerevisiae. *Bioinformatics*, 15(7):607–611.

Zielinski, T., Moore, A. M., Troup, E., Halliday, K. J., and Millar, A. J. (2014). Strengths and limitations of period estimation methods for circadian data. *PLoS ONE*, 9(5):e96462.