Cogn Comput manuscript No.
(will be inserted by the editor)

A Gaussian Process Model for Inferring the Dynamic Transcription Factor Activity using Ornstein-Uhlenbeck Kernel

Muhammad Arifur Rahman · Neil D. Lawrence

Received: 12–09–2018/ Accepted: date

Abstract In molecular biology and genetics, a transcription factor is a protein that binds to specific DNA sequence. Transcription factors control the flow (or transcription) of genetic information from DNA to mRNA. To develop models of cellular processes quantitative estimation of the regulatory relationship between transcription factors and genes is an essential requirement. While modelling transcription factor activity, a Markovian assumption based state space model where any state solely depends on its previous state might not be a suitable option as a general case. The Gaussian process, a nonparametric model, can easily overcome this state dependency. However, designing a covariance function for Gaussian process model is crucial. In this paper, we justified the rationale for the choice of the Ornstein-Uhlenbeck covariance function and developed a Gaussian process model to figure out the dynamics of transcription factor activity.

1 Introduction

In the consequences of diverse internal and external stimuli, cellular life must recognise and respond appropriately. Gene expression is the converted form of an abstract code of biological information preserved inside the DNA. Gene expression is a tightly regulated process which allows cells to respond to any internal and external stimuli or change of environment. With very few minor exceptions, all cell types in a multicellular organism preserve the identical genetic information. For a specified organism, any individual cell type expresses only a unique subset of the total number of distinct genes. Differentially expressed genes are specified by unique epigenetic information. This information is present in the distinct cell and also determines the phenotype of the composite organism [1, 2].

DNA transcription is a process that controls the first step of gene expression and transcribes genetic information from DNA to RNA. The transcription process is pivotal to produce proteins, and the transcription profile is a highly preferable parameter for the recognition of a distinct cell type. In general, a complex biological regulatory network controls the differential gene expression, in which particular transcription factors relay the signals to specific target genes. Among these transcription factors, many of them are DNA-binding proteins, which bind to regulatory DNA elements located cis to the target genes [3].

Using microarray experiments systematic gene expression quantification has been introduced decades ago [4] yet the recent development of chromatin immunoprecipitation (ChIP) followed by microarray (also known as ChIP-chip) and sequencing (also known as ChIP-seq) technologies boosted the genome-wide identification of transcription factor [5, 6].

Transcription factors (TFs) are expository for the transcriptional regulation and the transcriptional regulatory system plays a pivotal role in controlling many biological processes, ranging from cell cycle progression and preservation of intracellular metabolic and physiological stability [7, 8]. It also plays a crucial role in cellular differentiation and developmental dynamics by ensuring the meticulous expression of specific genes [9].

Muhammad Arifur Rahman (🖂)
Department of Computer Science, University of Sheffield, UK
Department of Physics, Jahangirnagar University, Savar, 1342 - Dhaka, Bangladesh
E-mail: arif@juniv.edu

Neil D Lawrence
Department of Computer Science, University of Sheffield, UK
Amazon, UK.

There are two types of Transcription factors: some of them are *general* and rest are *sequence-specific*. *General* transcription factors act collaboratively with RNA polymerase II and are ubiquitously involved in the transcription of a large number of genes [10]. *Sequence-specific* transcription factors bind particular subsets of target genes, leading to well defined spatio-temporal patterns of gene expression [11].

A number of diseases emerge from a breakdown in the regulatory system: transcription factors are over-represented among oncogenes [12], and almost one-third of human developmental disorders have been ascribed to dysfunctional TFs [13]. Even alterations in the activity and regulatory specificity of transcription factors are likely to be a primary reason for evolutionary adaptation and phenotypic diversity [14]. Indeed, recent research and study have already proved that increased sophistication of the transcriptional regulatory system seems to have been a key requirement for the emergence of metazoan life [15]. So inferring the dynamics of transcription factors activities might play a significant role to obtain a deeper insight into the gene regulatory network.

To build a transcriptional regulatory network, it may appear that knowledge of particular biomedical functions of transcription factors are not important. Even with some naive assumptions, it may appear that promoters or repressors regulate the transcription process similarly under similar condition. However, these assumptions can't be considered as general rules as it already reported that transcription factor DNA binding events might not follow the exact or defined biological regulatory mechanism. Such as, in a study, [16] reported that in different mutants of yeast *HAP1* positive control could selectively affect different gene expressions. Using comparative genomics and functional scanning of transgenic mice [17] showed how the transcription factor *TBX4* plays a pivotal role in hindlimb and vascular development. They showed a group of enhancers control the gene expresses level in different tissues. Genomic analysis also showed the relationship between transcription factor binding events and transcription factors from genes affected by the mutation. In their study [18] explained even further where they reported about the role of cofactors during the transcriptional mechanism. Only understanding of condition-specific activation, noise presence in the data or transcriptional cascades is not enough to grasp the actual understanding of these phenomena. Therefore, to dissect the regulatory mechanisms and gaining a better insight the complete index of transcription factor activities and its interacting partners would be invaluable.

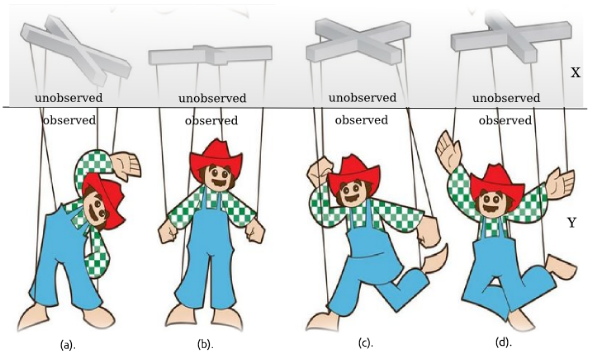

Fig. 1: Marionette analogy of latent variable model: Marionette's different dynamics are observed (later represented by **Y** in the Section 3). But these movements are controlled by the puppeteer's control bar which is unobserved (later these dynamics represented by **X** in the Section 3).

Modelling transcription factor activities can be seen as latent variable modelling [19,20]. We can observe the gene expression level, but these expression levels are regulated by protein-coding genes that bind to a specific DNA sequence and controls the production rate of mRNA. These gene expression levels are analogous to the movement of the marionette (Figure 1), but the transcription factor's activity is unobservable like the puppeteer's control bar which controls the gene expression levels.

In recent years an idea that has gained a lot of interest to infer the regulatory mechanism from the expression levels of genes. There has been a wealth of research on microarray data. A number of methods [21–23] aim to infer a matrix of transcription factor activities (TFAs). These TFAs can be summed up in a single number at a certain experimental point to find the concentration of the transcription factor and its binding affinity to its target genes. A variety of approaches has been proposed to infer these TFAs. For example, [23] developed a data decomposition technique with dimension reduction and introduced the concept of network component analysis. This method takes account of the connectivity information by imposing algebraic constraints on the factors. They argued that classical statistical methods such as principal component analysis and independent component analysis, do not consider the underlying network structure while computing low dimensional or hidden representation of a high-dimensional data sets like DNA microarray.

Later [21] used a dimension reduction technique (singular value decomposition) to figure out TFAs and also the correlation between DNA replication initiation and

RNA transcription during the yeast cell cycle. Using multivariate regression and backward variable selection to identify active transcription factors [22] targeted the same; [24] used the partial least squares (PLS) regression to infer the true TFAs from a combination of tRNA expression and DNA-protein binding measurement. A major drawback of the methods mentioned above is that transcription factor activities do not hold any information regarding the strength of the regulators' interactivity between the transcription factors and its different target genes. It is expected that depending on the experimental conditions the transcription factor activities can vary from gene to gene. It is also expected that different transcription factors may bind to the same gene. In most cases, realistic information about the intervals may not be true as they were not based on the fully probabilistic model. Moreover, false positives are always a problem for connectivity data, typically a significant portion of ChIP data suffers from it [24]. Furthermore, due to the various cellular process or changes in environmental conditions the structure of the regulatory network of the cell can change considerably. Using regression-based methods, it is difficult to track these changes. [25] built a probabilistic model, using the basic framework of dynamic Bayesian networks based on discrete random variables for protein concentrations and binding affinities. Though the model was more realistic, the computational complexity for genome-wide analysis can be expensive.

Here in this paper, we model a kernel or covariance function of the Gaussian process for reconstructing transcription factor activities where the gene expression profiles and the connectivity information of binding data (connectivity matrix) between genes and transcription factors are given. Our modelling framework initiated from the ideas of [26] where they used a linear-Gaussian state-space modelling framework to infer the TFA.

We note that the TFA model proposed by [26] is a linear Gaussian state space model with Markov property. A model with Markov property have the state dependencies; i.e. any certain state depends on the transition probability of previous state (or states for order-k Markov model). This state dependency with Markov property may deem as a restrictive assumption where the transition probability from the first state to the second state is completely different from the transition from the second state to the third state and so on. We also noticed that the linear Gaussian state space model of TFA proposed by [26] is identical to a Gaussian process model with a particular kernel or covariance function. In general, this kernel or covariance function of a Gaussian process is constructed considering the similarity measure between every possible input pairs (or states). We, therefore, started our modelling framework directly from the Gaussian process perspective to attain the same target. We also propose a computational trick, based on judicious application of singular value decomposition (SVD), to fit the Gaussian process efficiently in a reduced 'transcription factor activity' space.

In the probabilistic inference of TFAs, [27] used gene expression time series of synchronized yeast cells from the cell division cycle gene CDC-15 (Systematic name YAR019C) experiment. The second data set is from ChiP-chip [5,6] experiments performed on yeast by [28]. We constructed the binding information between transcription factors and genes. In this paper we are going to combine this binding information with the gene expression information to infer transcription factor activities.

2 Gaussian Process

A Gaussian process (GP) is a collection of random variables, any finite number of which have a joint Gaussian distribution [29]. It is a continuous stochastic process and defines probability distributions for functions. It can be also viewed as a collection of random variables indexed by a continuous variable. Any finite set of values from the collection can be written as a vector. Let's consider $\mathbf{f} = \{f_1, f_2, f_3, ..., f_N\}$ corresponds with indexed inputs $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_N\}$. In Gaussian processes, variables from these random functions are jointly normally distributed and as a whole can be represented as a multivariate Gaussian distribution

$$p(\mathbf{f}|\mathbf{X}) \sim \mathcal{N}\left(\mathbf{f}|\boldsymbol{\mu}, \mathbf{K}\right), \tag{1}$$

where $\boldsymbol{\mu}$ is the mean and $\mathbf{K}$ is the covariance matrix. Both potentially depends on $\mathbf{X}$. The Gaussian distribution is over vectors but the Gaussian process is over functions.

We need to define the mean function and covariance function for a Gaussian process prior. If $f(\mathbf{x})$ is a real valued process, a Gaussian process is completely defined by its mean function and covariance function given in Equation 2 and Equation 3 respectively. Usually the mean function $m(\mathbf{x})$ and the covariance function $k(\mathbf{x}, \mathbf{x}')$ are defined as

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})], \tag{2}$$

and

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))], \tag{3}$$

where $\mathbb{E}$ represents the expected value. We denote the Gaussian process as

$$f\left(\mathbf{x}\right) \sim \mathcal{GP}\left(m\left(\mathbf{x}\right), k\left(\mathbf{x}, \mathbf{x}'\right)\right). \tag{4}$$

The covariance matrix $\mathbf{K}$ is constructed from the covariance function $k(\mathbf{x}, \mathbf{x}')$ and $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, that is

$$\mathbf{K} = \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \cdots & k(\mathbf{x}_1, \mathbf{x}_n) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \cdots & k(\mathbf{x}_2, \mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & k(\mathbf{x}_n, \mathbf{x}_2) & \cdots & k(\mathbf{x}_n, \mathbf{x}_n) \end{pmatrix}. \tag{5}$$

Loosely speaking, a Gaussian process is multivariate Gaussian distribution defined over an infinite number of dimensions. A sample from a Gaussian process is a random function. While a $n-$dimensional Gaussian distribution is fully specified by mean $\boldsymbol{\mu}$, a $n \times 1$ vector of expectations and covariance matrix $\mathbf{K}$, the $n \times n$ matrix of covariances between all pair of points.

It is a common practice to consider a Gaussian process with zero mean when no prior information is available. This is not excessively restrictive as a variety of functions can be generated by a zero mean process. A second order stationary process has a constant mean and the covariance function solely depends on the distance between the inputs. Zero-mean process is a simplification just by centring the data as $\mathbf{t} = \mathbf{t} - \overline{\mathbf{t}}$, where $\overline{\mathbf{t}}$ is the data sample mean. An extra constant term with the covariance function can reflect the variation from the mean of the process [30]. So, a constant-mean or a zero-mean assumption is not overly restrictive in practice.

2.1 GP: Covariance Functions

The covariance function (also called kernel, kernel function or covariance kernel) characterises the properties or nature of the samples drawn from a Gaussian process. The covariance function encodes the modelling assumptions we wish to incorporate in our application. The mandatory requirement of a covariance matrix is to be symmetric positive semi-definite[1]. So, as long as the covariance function generates symmetric positive semi-definite matrix, we can use that function for a Gaussian process. Smoothness, periodicity, amplitude, lengthscale etc. are basic properties that can be incorporated while designing Gaussian process covariance function. Once the decision to model with a Gaussian process has been made the choice of the covariance function is a central step in modelling. The main goal of this thesis is to develop covariance functions suitable for transcription factor activity analysis and clustering gene expressions. Here we will discuss some of the very well known and widely used covariance functions. A wide choice of valid covariance functions and their detail description can be found in [29].

Any form of covariance function is acceptable, provided it satisfies the following equation

$$\sum_{i,j} a_i a_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0 \tag{6}$$

where, $a_i, a_j \ldots a_n$ are arbitrary real coefficients and $\mathbf{x}_i, \mathbf{x}_j \ldots \mathbf{x}_n$ are finite set of data points. A covariance function is termed 'stationary'when it follows

$$\text{Cov}\left[f(\mathbf{x}_i), f(\mathbf{x}_j)\right] = k(\|\mathbf{x}_i - \mathbf{x}_j\|) \tag{7}$$

for all $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^D$. In practice, a stationary covariance function gives a function that is invariant to translation and does not depend on the absolute location of the corresponding inputs; rather it depends on the distance separating them.

If the covariance does not only depend on the distance between the data points in the input space, rather the model needs to adapt to functions where smoothness varies with the inputs, a non-stationary covariance function will be required. There are many interesting non-stationary covariance functions. Depending on the nature or trend a careful selection of appropriate covariance function is essential. One of the simplest examples of non-stationary covariance function which have a linear trend can be expressed by

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sum_{d=1}^{D} a_d x_i^d x_j^d \tag{8}$$

where x_i^d is the d^{th} component of $\mathbf{x}_i \in \mathbb{R}^D$.

In this paper, as a prior, we used some stationary covariance functions and in the following section, we briefly describe some of them. Non-stationary covariance functions are beyond our scope and a detailed description is available in [29].

2.2 The Matérn Covariance Function

The Matérn class of covariance functions can be expresses by Equation 9

$$K_{Mat}(r) = a^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}r}{l}\right)^{\nu} K_{\nu}\left(\frac{\sqrt{2\nu}r}{l}\right) \tag{9}$$

where a, l, ν are positive hyperparameters, K_ν is a modified Bessel function and $\Gamma(.)$ is the Gamma function.

[1] A matrix $\mathbf{C}$ is called positive semi-definite if $\mathbf{z}^\top \mathbf{C}\mathbf{z} \geq 0$ for all $\mathbf{z}$. Where $\mathbf{z}$ is a non zero column vector of length n, $\mathbf{C}$ is a $n \times n$ symmetric real matrix and $\mathbf{z}^\top$ is the transpose of $\mathbf{z}$.

Hyperparameter ν controls the roughness of the function and like Exponentiated quadratic covariance function the parameters a and l controls the amplitude and lengthscale respectively. Though for $\nu \to \infty$ we can obtain the exponentiated quadratic kernel. For finite value of ν, the sample functions are significantly rough.

The simpler form of Matérn covariance function is obtained when ν is considered as a half-integer: $\nu = p + 1/2$, where p is a non-negative integer. The covariance function can be expressed as a product of an exponential and a polynomial of order p. In [31] Abramowitz et al. derived the general expression as follows

$$K_{\nu=p+1/2}(r) = a^2 \exp\left(-\frac{\sqrt{2\nu}r}{l}\right) \frac{\Gamma(p+1)}{\Gamma(2p+1)} \sum_{i=0}^{p} \frac{(p+i)!}{i!\,(p-i)!} \left(\frac{\sqrt{8\nu}r}{l}\right)^{p-i}. \tag{10}$$

The most interesting cases for machine learning are $\nu = 3/2$ and $\nu = 5/2$, for which we get the following equations respectively

$$K_{\nu=3/2}(r) = a^2 \left(1 + \frac{\sqrt{3}r}{l}\right) \exp\left(-\frac{\sqrt{3}r}{l}\right) \tag{11}$$

and

$$K_{\nu=5/2}(r) = a^2 \left(1 + \frac{\sqrt{5}r}{l} + \frac{5r^2}{3l^2}\right) \exp\left(-\frac{\sqrt{5}r}{l}\right). \tag{12}$$

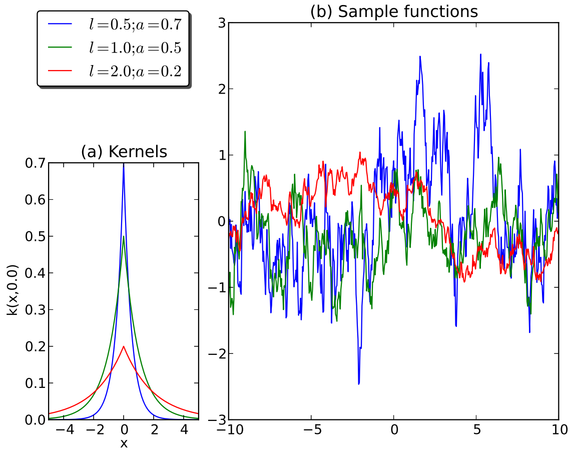

Fig. 2: The OU kernels (a) and random sample functions (b) for different hyperparameter settings shown in the top left.

2.3 The Ornstein-Uhlenbeck Process

The Ornstein-Uhlenbeck process [32] is a special case of Matérn class covariance functions. The Ornstein-Uhlenbeck (OU) process was developed as a mathematical model of the velocity of a particle moving with Brownian motion.

The Ornstein-Uhlenbeck process can be found setting up $\nu = 1/2$ and expressed as Equation 13. Figure 2(a) shows the kernel and Figure 2(b) shows the sample functions form the OU process with different hyperparameter settings as shown to the left of the figure.

$$K_{\nu=1/2}(r) = a^2 \exp\left(-\frac{r}{l}\right) \tag{13}$$

2.4 Constructing Kernels

Modelling kernel is the central step in Gaussian process modelling. A number of 'built-in' kernels (both stationary and non-stationary) are available for Gaussian process, yet we may need to model a complicated structure which is not expressed very well by any known kernel. To model such a structure, we may build our own 'customised kernel' with the required structure or, desired properties. An addition of two Gaussian variables is a Gaussian. Scaling a Gaussian also leads to a Gaussian. These two basic mathematical properties help to develop a range of kernels from a very simple one to complex a one.

Assume an univariate data is globally periodic and local structures governed by some random motions (Brownian motion). There are multiple choices dealing with the global structure and one of the possible solutions could be a Cosine kernel for the global structure and a Brownian kernel for local structures in an additive form. The addition of two positive semi-definite kernels together always results in another positive semi-definite kernel. Figure 3 shows the sample functions and representation of the newly constructed kernel.

2.5 Gaussian Process Regression

Gaussian process regression can be done using the marginal and conditional properties of the multivariate Gaussian distribution. Let's consider that we have the observation $\mathbf{f}$ of a function at the observation point $\mathbf{x}$. Now we wish to predict the values of that function at the observation points $\mathbf{x}_\star$, which are represented by $\mathbf{f}_\star$. Then the joint probability of $\mathbf{f}$ and $\mathbf{f}_\star$ can be obtained as

$$p\left(\begin{bmatrix}\mathbf{f}\\ \mathbf{f}_\star\end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{f}\\ \mathbf{f}_\star\end{bmatrix}\middle|\mathbf{0}, \begin{bmatrix}\mathbf{K}_{\mathbf{x},\mathbf{x}} & \mathbf{K}_{\mathbf{x},\mathbf{x}_\star}\\ \mathbf{K}_{\mathbf{x}_\star,\mathbf{x}} & \mathbf{K}_{\mathbf{x}_\star,\mathbf{x}_\star}\end{bmatrix}\right) \tag{14}$$

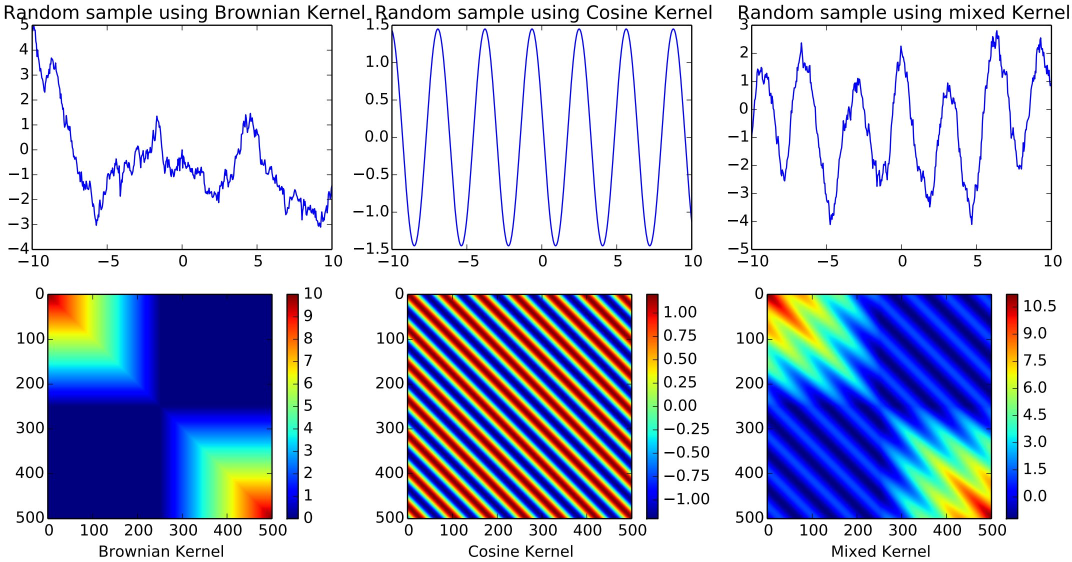

Fig. 3: Construction of 'made by order'kernel adding two basic kernels: an example of a univariate data which is globally periodic and locally governed by some random motions. (top-left) a sample taken using Brownian kernel, (top-middle) a sample is taken using Cosine kernel, (top-right) a sample taken using a newly constructed kernel by adding two kernels, (bottom-left) Brownian kernel, (bottom-middle) Cosine kernel, (bottom-right) newly constructed kernel.

where the covariance matrix $\mathbf{K}_{\mathbf{x},\mathbf{x}}$ has elements derived from the covariance function $k(x, x')$, such that the $(i,j)^{th}$ element of $\mathbf{K}_{\mathbf{x},\mathbf{x}}$ is given by $k(\mathbf{x}[i], \mathbf{x}[i])$. The conditional property of a multivariate Gaussian is used to perform regression. The conditional property can be represented by

$$p(\mathbf{f}_\star|\mathbf{f}) = \mathcal{N}\left(\mathbf{f}_\star \middle| \mathbf{K}_{\mathbf{x}_\star,\mathbf{x}}\mathbf{K}_{\mathbf{x},\mathbf{x}}^{-1}\mathbf{f}, \mathbf{K}_{\mathbf{x}_\star,\mathbf{x}_\star} - \mathbf{K}_{\mathbf{x}_\star,\mathbf{x}}\mathbf{K}_{\mathbf{x},\mathbf{x}}^{-1}\mathbf{K}_{\mathbf{x},\mathbf{x}_\star}\right). \tag{15}$$

In an ideal case, the observation $\mathbf{f}$ is noise-free, but, in practice, it is always corrupted with some noise. Let's consider $\mathbf{y}$ as a corrupted version of $\mathbf{f}$. If we consider this noise as Gaussian noise, we can write $p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2\mathbf{I})$, where σ^2 is the variance of the noise and $\mathbf{I}$ is the identity matrix with the appropriate size and marginalise the observation $\mathbf{f}$. Then, the joint probability of $\mathbf{y}$ and $\mathbf{f}_\star$ can be represented by

$$p\left(\begin{bmatrix}\mathbf{y}\\ \mathbf{f}_\star\end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{y}\\ \mathbf{f}_\star\end{bmatrix}\middle|\mathbf{0}, \begin{bmatrix}\mathbf{K}_{\mathbf{x},\mathbf{x}} + \sigma^2\mathbf{I} & \mathbf{K}_{\mathbf{x},\mathbf{x}_\star}\\ \mathbf{K}_{\mathbf{x}_\star,\mathbf{x}} & \mathbf{K}_{\mathbf{x}_\star,\mathbf{x}_\star}\end{bmatrix}\right). \tag{16}$$

Regression with Gaussian process is a Bayesian method. From the knowledge of a *prior* over a function and data, we infer a *posterior* and this happens in a closed form of Equation 15.

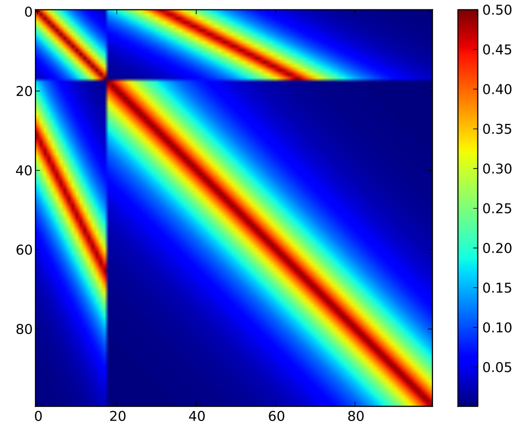

Fig. 4: Overall representation of covariances between training and test data.

Figure 4 shows the overall covariance structure between some training and test data. For this example, we choose 18 training points and 82 test points. We observe the shaded structure because some of the training data are closer to some of the test data. Observing this structure, we can also figure out the closeness between training and test data.

2.6 Making Predictions

The probability density is represented by functions. Due to consistency this density is known as a process. Also by this property, any future values of $\mathbf{f}_\star$ which are unobserved can be predicted without affecting $\mathbf{f}$. To make predictions of the test data, we use the conditional distribution. In an ideal case, the conditional distribution is $p\left(\mathbf{f}_\star|\mathbf{f}\right)$ and if we consider the noise, the conditional distribution will be $p\left(\mathbf{f}_\star|\mathbf{y}\right)$. Both of the distributions are also Gaussian

$$\mathbf{f}_\star \sim \mathcal{N}\left(\boldsymbol{\mu}_f, \mathbf{C}_f\right). \tag{17}$$

The mean of the conditional distribution in Equation 17 is

$$\boldsymbol{\mu}_f = \mathbf{K}_{\mathbf{x},\mathbf{x}_\star}^\top \left[\mathbf{K}_{\mathbf{x},\mathbf{x}} + \sigma^2\mathbf{I}\right]^{-1}\mathbf{y} \tag{18}$$

and its covariance is given by

$$\mathbf{C}_f = \mathbf{K}_{\mathbf{x}_\star,\mathbf{x}_\star} - \mathbf{K}_{\mathbf{x},\mathbf{x}_\star}^\top \left[\mathbf{K}_{\mathbf{x},\mathbf{x}} + \sigma^2\mathbf{I}\right]^{-1}\mathbf{K}_{\mathbf{x},\mathbf{x}_\star}. \tag{19}$$

These results can be calculated using the block matrix inverse rules. Figure 5 shows a visual representation of Gaussian process regression for a one-dimensional function. Coloured solid lines represent different samples from the process and the dotted line is the mean function. The shaded area is the 95% confidence interval. Figure 5(a) represents a Gaussian process not conditioned on any data points. Without any observations, the prior uncertainty about the underlying function is constant everywhere. Figure 5(b-e) show some posterior samples after conditioning on different amount of training data as shown in the figure.

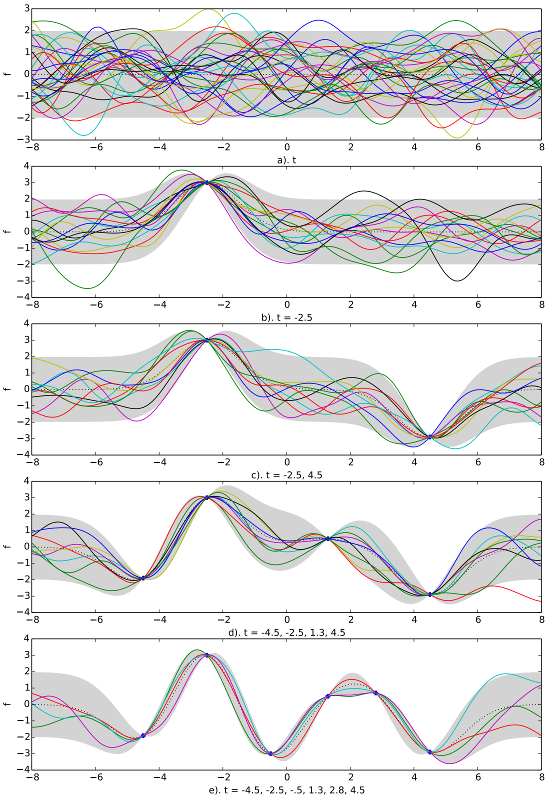

Fig. 5: A representation of Gaussian process regression: Modelling one-dimensional function using Gaussian process. Coloured solid lines represent different samples from the process and the dotted line is the mean function. The shaded area is the 95% confidence interval. (a) A Gaussian process not conditioned on any data points. Without any observations, the prior uncertainty about the underlying function is constant everywhere. (b-e) The posterior after conditioning on different amount of data.

2.7 Hyperparameter Learning

To construct the covariance function, still we need to consider the hyperparameters and optimize them. These adjustable parameters alter the distribution of the function output values obtained from a Gaussian process. The most efficient and commonly used optimization technique for hyperparameters is maximum likelihood. If we consider all the hyperparameters α (controls the amplitude), σ^2 (variance of the noise) and l (length-scale) in a vector $\boldsymbol{\theta}$, we can use gradient methods to optimize $p\left(\mathbf{y}|\boldsymbol{\theta}\right)$ with respect to $\boldsymbol{\theta}$. The Log likelihood is given by

$$\begin{aligned}\log\, p\left(\mathbf{y}|\boldsymbol{\theta}\right) = -\frac{D}{2}\log 2\pi - \frac{1}{2} \times \log\left|\mathbf{K}_{\mathbf{x},\mathbf{x}} + \sigma^2\mathbf{I}\right| \\ - \frac{1}{2}\mathbf{y}^\top\left[\mathbf{K}_{\mathbf{x},\mathbf{x}} + \sigma^2\mathbf{I}\right]^{-1}\mathbf{y}.\end{aligned} \tag{20}$$

We can have the log maximum likelihood by

$$\boldsymbol{\theta}_{max} = \arg\max_{\theta}\left(p\left(\mathbf{y}|\boldsymbol{\theta}\right)\right). \tag{21}$$

3 Mechanistic model of TFA

The logged gene expression measurements are collected in a design matrix, $\mathbf{Y} \in \mathbb{R}^{N\times d}$ where N is the number of genes and d is the time points or number of experiments. The binary matrix $\mathbf{X} \in \mathbb{R}^{N\times q}$ is the connectivity measurements, where q is the number of transcription factors. We assume that $\mathbf{X}_{i,j}$ is '1' if transcription factor j can bind gene i, '0' otherwise.

In [26] Sanguinetti et al. used a latent variable model. TFAs were obtained by regression from the gene expressions using the connectivity information, given the following linear model

$$\mathbf{y}_n = \mathbf{B}_n\mathbf{x}_n + \boldsymbol{\epsilon_n} \tag{22}$$

Here $n = 1, ..., N$ indexes the gene, $\mathbf{y}_n = \mathbf{Y}(n, :)^\top$, $\mathbf{x}_n = \mathbf{X}(n, :)^\top$ and $\boldsymbol{\epsilon_n}$ is an error term. The matrix $\mathbf{B}_n$ has d rows and q columns, and models the gene specific TFAs.

Different TFAs for every individual gene will increase number of model parameters drastically. This huge parameter space can be dealt through marginalization by prior distribution on the rows of $\mathbf{B}_n$. Yet, two physically plausible assumptions for selecting the prior distribution will be helpful to determine the gene specific TFAs.

- The first assumption: $\mathbf{b}_{nt}$ has the Markov property and hence gene specific TFA $\mathbf{b}_{nt}$ at time t depends exclusively on the gene specific TFA at time $(t-1)$.
- The second assumption: the prior distribution to be stationary in time.

In order to support these assumptions, there will be two limiting cases for prior distributions. Let's first assume all the $\mathbf{b}_{nt}$ are identical for all t. Then the first limiting case is

$$\mathbf{b}_{n1} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{23}$$

and

$$\mathbf{b}_{n(t+1)} \sim \mathcal{N}(\mathbf{b}_{nt}, \mathbf{0}). \tag{24}$$

If the experimental dataset comes by replicating a condition then this model is an appropriate one. The second limiting case appears when all the $\mathbf{b}_{nt}$ are independent and identically distributed (iid)

$$\mathbf{b}_{nt} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}). \tag{25}$$

This is the case when experimental dataset comes from independent samples drawn without any temporal order.

Sanguinetti et al. [26] expected a realistic model of time series data to be somewhere in between these two extremes (Equation 23, Equation 24 and Equation 25)

$$\mathbf{b}_{n(t+1)} \sim \mathcal{N}(\gamma\mathbf{b}_{nt} + (1-\gamma)\boldsymbol{\mu}, (1-\gamma^2)\boldsymbol{\Sigma}) \tag{26}$$

for $t = 1, ..., (d-1)$ and $\mathbf{b}_{n1} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ where γ is a parameter measuring the degree of temporal continuity of the TFAs. If genes are independent for a given TFA then the likelihood function is given by

$$p\left(\mathbf{Y}|\mathbf{B},\mathbf{X}\right) = \prod_{n=1}^{N} p\left(\mathbf{y}_n|\mathbf{B}_n, \mathbf{x}_n\right). \tag{27}$$

Considering Gaussian noise $\boldsymbol{\epsilon}_n \sim \mathcal{N}\left(0, \boldsymbol{\sigma}^2\mathbf{I}\right)$ we have

$$p\left(\mathbf{y}_n|\mathbf{B}_n, \mathbf{x}_n\right) = \mathcal{N}\left(\mathbf{y}_n|\mathbf{B}_n\mathbf{x}_n, \boldsymbol{\sigma}^2\mathbf{I}\right). \tag{28}$$

Factorizing the likelihood along the experiments with the assumption of spherical Gaussian noise distribution, we can rewrite the Equation 27 as

$$p\left(\mathbf{Y}|\mathbf{B}, \mathbf{X}\right) = \prod_{t=1}^{d}\prod_{n=1}^{N} p\left(\mathbf{y}_{nt}|\mathbf{b}_{nt}, \mathbf{x}_n\right) \tag{29}$$

where

$$p\left(\mathbf{y}_{nt}|\mathbf{b}_{nt}, \mathbf{x}_n\right) = \mathcal{N}\left(\mathbf{y}_{nt}|\mathbf{b}_{nt}^\top\mathbf{x}_n, \sigma^2\right). \tag{30}$$

4 Toward the GP model of TFA

We are interested in developing a non-parametric model of transcription factor activity using Gaussian process. Here, we want to prove that there is an analogical pathway[2] to construct a kernel function for Gaussian process model from Markovian assumption based probabilistic approach of [26]. Let's Recall the Equation 26 and then we have the probabilistic gene specific TFAs as

$$\mathbf{b}_{n(t+1)} \sim \mathcal{N}(\gamma\mathbf{b}_{nt} + (1-\gamma)\boldsymbol{\mu}, (1-\gamma^2)\boldsymbol{\Sigma}).$$

For a discrete time variable k the above equation can be rewritten as

$$\mathbf{b}_{n(k+1)} \sim \mathcal{N}\left(\gamma\mathbf{b}_{nk} + (1-\gamma)\boldsymbol{\mu}, (1-\gamma^2)\boldsymbol{\Sigma}\right), \tag{31}$$

and

$$\mathbf{b}_{n_1} \sim \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right). \tag{32}$$

Let's now form a continuous model which has the same finite-dimensional distribution. First we construct a one-dimensional process with the property

$$u_{k+1} \sim \mathcal{N}\left(\gamma u_k + (1-\gamma)\mu, (1-\gamma^2)s\right), \tag{33}$$

[2] We would like to acknowledge Simo Särkkä, Academy Research Fellow, Aalto University, Finland for his valuable suggestions and guidelines.

where μ and s are scalar.

We can now assume that u_k's are actually values u_{t_k} from a continuous process $u(t)$ and let's assume that

$$t_k = kDt. \tag{34}$$

A good candidate for this kind of model is the mean-reverting *Ornstein-Uhlenbeck* model [32]

$$du = -\lambda\,(u - \mu)\,dt + q^{1/2}dB, \tag{35}$$

where B is a standard Brownian motion (i.e., Wiener process). This equation can now be solved on the time instants t_k and the result is a recursion

$$u(t_k) = au(t_{k-1}) + b\mu + w_{k-1}, \tag{36}$$

where $w_{k-1} \sim \mathcal{N}(0, c)$ with

$$a = \exp(-\lambda Dt)$$

$$\begin{aligned} b &= \int_0^D t\exp(-\lambda(Dt - s))ds \\ &= 1 - \exp(-\lambda Dt) \end{aligned}$$

$$\begin{aligned} c &= \int_0^D t\exp(-\lambda(Dt - s))q\exp(-\lambda(Dt - s))ds \\ &= q\int_0^D t\exp(-2\lambda(Dt - s))ds \\ &= [q/(2\lambda)][1 - \exp(-2\lambda Dt)]. \end{aligned}$$

That is,

$$u_{k+1} \sim \mathcal{N}\left(au_k + b\mu, c\right). \tag{37}$$

We can now match the coefficients

$$a = \exp(-\lambda Dt) = \gamma \tag{38}$$

$$b = 1 - \exp(-\lambda Dt) = 1 - \gamma \tag{39}$$

$$c = (1 - \gamma^2)s = [q/(2\lambda)][1 - \exp(-2\lambda Dt)] \tag{40}$$

Equation 38 has a nice solution $\gamma = \exp(-\lambda Dt)$ and from Equation 40 we have another solution $s = q/(2\lambda)$, which can be inverted to give $\lambda = -[1/Dt]\log\gamma$ and $q = -[2s/Dt]\log\gamma$.

If we fix $Dt = 1$, we get $\lambda = -\log\gamma$ and $q = -2s\log\gamma$.

We can now recall the (stationary) covariance function of the Ornstein-Uhlenbeck process

$$\begin{aligned} k_u(t, t') &= [q/(2\lambda)]\exp(-\lambda|t - t'|) \\ &= s\exp((\log\gamma)|t - t'|) \\ &= s\exp(|t - t'|(\log\gamma)) \\ &= s\exp(\log\gamma^{|t-t'|}) \\ &= s\gamma^{|t-t'|}. \end{aligned}$$

When we start from variance $s = q/[2\lambda]$, the process indeed is stationary from the beginning. Returning to the original vector valued $\mathbf{b}$, because the system is separable, we can conclude that the implied covariance function is just obtained by formally replacing s with $\mathbf{\Sigma}$ everywhere

$$\mathbf{K}_b(t, t') = \mathbf{\Sigma}\boldsymbol{\gamma}^{|t-t'|} \tag{41}$$

This is equivalent to considering the vector process of mean-reverting $Ornstein - Uhlenbeck$ model

$$\mathbf{db} = -\lambda(\mathbf{b} - \boldsymbol{\mu})\mathbf{dt} + Q^{1/2}\mathbf{dB}. \tag{42}$$

5 GP Model for TFA

Given $\mathbf{Y} \in \mathbb{R}^{n\times T}$, is the matrix of *log*-ed gene expression, where n is the number of genes, T is the time points. We assume a linear (additive) model to express the relationship between the gene expression level and the corresponding transcription factor activity. These transcription factor activities are unobserved (latent variables), and here we represent them by a matrix $\mathbf{F} \in \mathbb{R}^{q\times T}$, where q is the number of transcription factors. Our basic assumption is as follows

1. The activity of the transcription factors exists in time series, so there might have a temporal smoothness.
2. Between the transcription factors (especially simultaneously performing pair) there might have a correlation; regardless whether it is positive or negative.

Correlation Between Transcription Factors: We assumed there are q transcription factors. We can encode the correlation between different transcription factors in a covariance matrix $\mathbf{\Sigma}$. The dimension of the correlation matrix $\mathbf{\Sigma}$ will be $q \times q$..

Temporal Smoothness: We assume that the log of the transcription factors' activities is temporally smooth. The transcription factors' activities were obtained from an underlying Gaussian process. The kernel or covariance function $\mathbf{K}_t$ holds the relationship of temporal smoothness between the time points.

Intrinsic Coregionalization Model: Here we are interested to consider a joint process across all time points and transcription factor activities. The joint process across all t time points and across all q transcription factor activities can be nicely represented by an especial mathematical formulation known as intrinsic model of coregionalization (ICM). The joint covariance function (here we represent it by $\mathbf{K}_f$) across all time points and transcription factors are given by the Kronecker product.

$$\mathbf{K}_f = \mathbf{K}_t \otimes \mathbf{\Sigma} \tag{43}$$

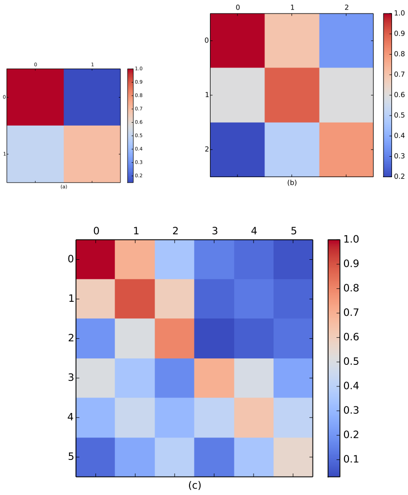

Fig. 6: Demonstration of Kronecker product by tiling. Assume (a) represents $\mathbf{K}_t$ and (b) represents $\mathbf{\Sigma}$ of Equation 43, then the representation of $\mathbf{K}_f$ will be as like (c).

This is known as an intrinsic coregionalization model [33] . In a review [34] presented the machine learning orientated approach of these methods. The matrix $\mathbf{\Sigma}$ of Equation 43 is also known as the coregionalization matrix. Figure 6 shows is a simple realization of intrinsic model of coregionalization by tiling.

5.1 Relation to Gene Expressions

Let us consider that the gene expression of j^{th} gene is obtained from the product of the transcription factors that bind to that specific gene. There exists a log-linear relationship as we are working in log space. The log of the j^{th} gene's expression at the i^{th} time point is represented by $\mathbf{y}_{i,j}$. For $\mathbf{y}_{i,j}$, there exists a linear relationship to the log of the transcription factor activities at the i^{th} time point and represented by $\mathbf{f}_{i,:}$. This relationship is given by the binding information from $\mathbf{S}$. Though the ideal process should be noise free, for a practical case, we need to consider corrupting noise for a number of reasons. Here, we are considering some additive Gaussian noise to obtain our final observation.

$$\mathbf{y}_{i,j} = \mathbf{S}\mathbf{f}_{:,i} + \boldsymbol{\epsilon}_i \tag{44}$$

where the Gaussian noise is found from

$$\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}) \tag{45}$$

5.2 Gaussian Process Model of Gene Expression

The logged gene expressions are represented by a design matrix $\mathbf{Y} \in \mathbb{R}^{n\times T}$, where n is the number of genes and T is the number of experiments or time points. To facilitate our mathematical formulation, we need to reorganize this matrix to a vector. Let us consider a new vector $\mathbf{y}$ with length $n \times T$ is formed separating the time series of $\mathbf{Y}$ and stacking them one top of another. Similarly with q transcription factors and T time points we have a $q \times T$ length vector for $\mathbf{f}$. To build the relationship between $\mathbf{y}$ and $\mathbf{f}$ we can use Kronecker products as demonstrated in Figure 6). We can determine the relationship between $\mathbf{y}$ and $\mathbf{f}$ by the equation

$$\mathbf{y} = [\mathbf{I} \otimes \mathbf{S}]\, \mathbf{f} + \boldsymbol{\epsilon}. \tag{46}$$

Using standard properties of multivariate Gaussian distributions we have

$$\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}), \tag{47}$$

where

$$\mathbf{K} = \mathbf{K}_t \otimes \mathbf{S}\mathbf{\Sigma}\mathbf{S}^\top + \sigma^2\mathbf{I}. \tag{48}$$

This results in a covariance function that has the size $n \times T$ by $n \times T$ where the variable n represent the number of genes and other variable T represent the time points (or the stage of the experiments). However, using some mathematical formulation, we can reduce the size of the covariance function. Here we used singular value decomposition (SVD) of matrix $\mathbf{S}$. The matrix $\mathbf{S}$ has the dimensionality of n by q, where as before n is the number of genes and q is the number of transcription factors. The 'connectivity matrix' $\mathbf{S}$ contains a '1' if a transcription factor binds to a gene, and a '0' if there is no evidence of transcription factor to gene bindings. The likelihood of a multivariate Gaussian is

$$L = -\frac{1}{2}\log|\mathbf{K}| - \frac{1}{2}\mathbf{y}^\top\mathbf{K}^{-1}\mathbf{y} \tag{49}$$

In general, for the worst case the computational complexity of Gaussian process is $\mathcal{O}(n^3)$. Here in our experimental formulation, our computation complexity will be $\mathcal{O}(T^3n^3)$ because the vector $\mathbf{y}$ has the length $T \times n$ (for every individual gene there are T time points and there are total n number of genes). To deal with this computational complexity efficiently and get the likelihood here we are going to use matrix rotation.

5.3 Method of Computation

Rotation of Multivariate Gaussian: Mathematical formalism allows to rotate the data set and compute a new kernel or covariance function for a multivariate Gaussian. The newly computer covariance function will be valid for the rotated data set. If we want to compute the likelihood of the multivariate Gaussian from the rotated data set mathematically we have to insert $\mathbf{R}\mathbf{R}^\top$ in the basic likelihood function of the multivariate Gaussian (Equation: 49), where $\mathbf{R}$ is the rotation matrix and $\mathbf{R}\mathbf{R}^\top = \mathbf{I}$. Inserting $\mathbf{R}\mathbf{R}^\top$ at three points of the likelihood function we can write the Equation 49 as:

$$L = -\frac{1}{2}\log|\mathbf{K}\mathbf{R}^\top\mathbf{R}| - \frac{1}{2}\mathbf{y}^\top\mathbf{R}^\top\mathbf{R}\mathbf{K}^{-1}\mathbf{R}^\top\mathbf{R}\mathbf{y} + \text{const} \tag{50}$$

Using some basic algebra, a transformation of the data and rules of determinants will allow us to revisit the Equation: 50 and rewrite the likelihood function as:

$$L = -\frac{1}{2}\log|\mathbf{R}^\top\mathbf{K}\mathbf{R}| - \frac{1}{2}\hat{\mathbf{y}}^\top\left[\mathbf{R}^\top\mathbf{K}\mathbf{R}\right]^{-1}\hat{\mathbf{y}} + \text{const} \tag{51}$$

where we have introduced the rotated data: $\hat{\mathbf{y}} = \mathbf{R}\mathbf{y}$. The mathematical explanation here is very straightforward. If we rotate our dataset by $\mathbf{R}$ then to obtain the same likelihood we need to rotate the covariance matrix as well. If we rotate any side of the covariance matrix by $\mathbf{R}$ then maintaining the properties of multivariate Gaussian we will obtain the same likelihood.

A Kronecker Rotation: Here In this paper, we considered a joint process across all time points and transcription factor activities introducing an intrinsic model of coregionalization (ICM) where we used Kronecker product (Equation: 43). To construct our rotated covariance function, we have to use Kronecker rotation as used by [35]. We will consider connectivity matrix $\mathbf{S}$ is square matrix and then matrix $\mathbf{S}\boldsymbol{\Sigma}\mathbf{S}^\top$ will not be a full rank matrix. The size of the matrix will be $n \times n$ but its rank will be q (the most). In our experiment, we had n number of genes and q number of transcription factors, and we assume that $q < n$.

Here the connectivity matrix $\mathbf{S}$ is not a full rank matrix. Use of singular value decomposition (SVD) is always preferable when ranks is involved. The singular value decomposition of connectivity matrix $\mathbf{S}$ is:

$$\mathbf{S} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{V}^\top \tag{52}$$

where $\mathbf{Q}$ matches the dimensionality of $\mathbf{S}$ and its size is $n \times q$. $\boldsymbol{\Lambda}$ is a diagonal matrix of positive values. Here, $\mathbf{Q}^\top\mathbf{Q} = \mathbf{I}$ and also $\mathbf{V}^\top\mathbf{V} = \mathbf{I}$. It is to be noted that $\mathbf{Q}$ itself is not in a rotation matrix as its size is $n \times q$. However, as we assumed $q < n$, we can consider the first q columns of the n dimensional rotation matrix.

Let us consider the missing columns of the rotation matrix. For $\mathbf{Q}$ we have considered q columns and then there were $n - q$ missing columns. If rotation matrix $\mathbf{U}$ represents the missing columns then we will have a valid rotation matrix $\mathbf{R} = \left[\mathbf{Q}\ \mathbf{U}\right]$.

The rotation matrix will only rotate across the n dimensions of the genes, leaving the additional dimensions (across time). Mathematically we are interested to keep $\mathbf{K}_t$ unrotated. To obtain this properly for our covariance function we need to design our rotation matrix as $\mathbf{R} = \mathbf{I} \otimes \left[\mathbf{Q}\ \mathbf{U}\right]$. This will allow us to rotate $\mathbf{K}_n$ leaving $\mathbf{K}_t$ from $\mathbf{K}_t \otimes \mathbf{K}_n$. Now if we want to apply the rotation matrix to $\mathbf{K}$, the rotation of $\mathbf{K}_t \otimes \mathbf{S}\boldsymbol{\Sigma}\mathbf{S}^\top$, and the rotation of $\sigma^2\mathbf{I}$ both of them have to be considered.

Rotating $\sigma^2\mathbf{I}$ is comparatively easy. Here the identity matrix is multiplied by a scalar term. So, it will be unchanged

$$\mathbf{R}^\top\mathbf{I}\sigma^2\mathbf{R} = \mathbf{I}\sigma^2 \tag{53}$$

For that term $\mathbf{K}_t \otimes \mathbf{S}\boldsymbol{\Sigma}\mathbf{S}^\top$ we have

$$\left[\mathbf{I} \otimes \left[\mathbf{Q}\ \mathbf{U}\right]^\top\right]\mathbf{K}_t \otimes \mathbf{S}\boldsymbol{\Sigma}\mathbf{S}^\top\left[\mathbf{I} \otimes \left[\mathbf{Q}\ \mathbf{U}\right]\right] =$$

$$\mathbf{K}_t \otimes \left[\mathbf{Q}\ \mathbf{U}\right]^\top\mathbf{S}\boldsymbol{\Sigma}\mathbf{S}^\top\left[\mathbf{Q}\ \mathbf{U}\right]. \tag{54}$$

Since $\mathbf{S} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{V}^\top$ then we have

$$\left[\mathbf{Q}\ \mathbf{U}\right]^\top\mathbf{S}\boldsymbol{\Sigma}\mathbf{S}^\top\left[\mathbf{Q}\ \mathbf{U}\right] = \begin{bmatrix}\boldsymbol{\Lambda}\mathbf{V}^\top\boldsymbol{\Sigma}\mathbf{V}\boldsymbol{\Lambda} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}. \tag{55}$$

This allows us to separate our vector $\hat{\mathbf{y}}$ into a q dimensional vector $\hat{\mathbf{y}}_u = \mathbf{U}^\top\mathbf{y}$ for first q transcription factor and an $n-q$ dimensional vector $\hat{\mathbf{y}}_q = \mathbf{Q}^\top\mathbf{y}$ for the rest of the genes. The Gaussian likelihood for two separate terms can be written as

$$L = L_u + L_q + \text{const} \tag{56}$$

where

$$\begin{aligned}L_q = &-\frac{1}{2}\log|\mathbf{K}_t \otimes \boldsymbol{\Lambda}\mathbf{V}^\top\boldsymbol{\Sigma}\mathbf{V}\boldsymbol{\Lambda} + \sigma^2\mathbf{I}|\\ &-\frac{1}{2}\hat{\mathbf{y}}_q^\top\left[\mathbf{K}_t \otimes \boldsymbol{\Lambda}\mathbf{V}^\top\boldsymbol{\Sigma}\mathbf{V}\boldsymbol{\Lambda} + \sigma^2\mathbf{I}\right]^{-1}\hat{\mathbf{y}}_q\end{aligned} \tag{57}$$

and

$$L_u = -\frac{T(n-q)}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\hat{\mathbf{y}}_u^\top\hat{\mathbf{y}}_u \tag{58}$$

Strictly speaking, it is expected to fit both of these models jointly. However, for the shake of illustration, we will use a simpler procedure. First, we will fit the noise variance σ^2 on $\hat{\mathbf{y}}_u$ alone using L_u. After fitting the noise variance σ^2, we will fix the value of σ^2 in L_q and then optimise with respect to the remaining other parameters.

5.4 Making Prediction

Using Kronecker product we can rewrite the Equation 47 as

$$\mathbf{y_q} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{K}_{t,t} \otimes \mathbf{\Lambda}\mathbf{V}^T\mathbf{\Sigma}\mathbf{V}\mathbf{\Lambda} + \sigma^2\mathbf{I}\right) \tag{59}$$

Standard properties of multivariate Gaussian distributions tells us can split Equation 59 into

$$\mathbf{y_q} = \mathbf{g} + \boldsymbol{\epsilon} \tag{60}$$

where $\mathbf{g}$ and $\boldsymbol{\epsilon}$ are also Gaussian distributions and can be represented by

$$\mathbf{g} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{K}_{t,t} \otimes \mathbf{\Lambda}\mathbf{V}^T\mathbf{\Sigma}\mathbf{V}\mathbf{\Lambda}\right) \tag{61}$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}\right) \tag{62}$$

Now we can represent the matrix $\mathbf{F}$ of transcription factor activity as

$$\mathbf{F} = \mathbf{I} \otimes \mathbf{V}\Lambda^{-1}\mathbf{g} \tag{63}$$

$$\mathbf{\Sigma} = \mathbf{W}\mathbf{W}^T + \operatorname{diag}(\boldsymbol{\kappa}) \tag{64}$$

where $\boldsymbol{\kappa}$ is the kappa value from coregionalization matrix.

$$\mathbf{F} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{K}_{t,t} \otimes \mathbf{\Sigma}\right) \tag{65}$$

Now we can find the conditional distribution of $\mathbf{g}$ for given $\mathbf{y}_q$ by

$$p\left(\mathbf{g}|\mathbf{y}_q\right) \sim \mathcal{N}\left(\boldsymbol{\mu}_g, \mathbf{C}_g\right) \tag{66}$$

with a mean given by

$$\begin{aligned}\boldsymbol{\mu}_g = &\left[\mathbf{K}_{t_\star,t} \otimes \mathbf{\Lambda}\mathbf{V}^T\mathbf{\Sigma}\mathbf{V}\mathbf{\Lambda}\right] \\ &\left[\mathbf{K}_{t,t} \otimes \mathbf{\Lambda}\mathbf{V}^T\mathbf{\Sigma}\mathbf{V}\mathbf{\Lambda} + \sigma^2\mathbf{I}\right]^{-1}\mathbf{y}_q\end{aligned} \tag{67}$$

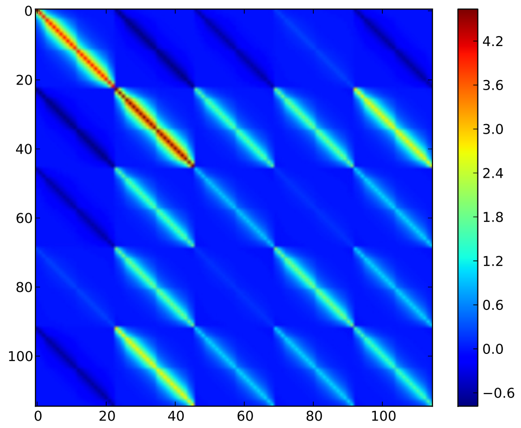

Fig. 7: Kernel of Intrinsic Coregionalization model $\mathbf{K}_f$ considering five transcription factors (a random choice for better visualization) where covariance matrix $\mathbf{\Sigma}$ of ($Equation48$) was constructed using Ornstein-Uhlenbeck kernel and White kernel in additive form.

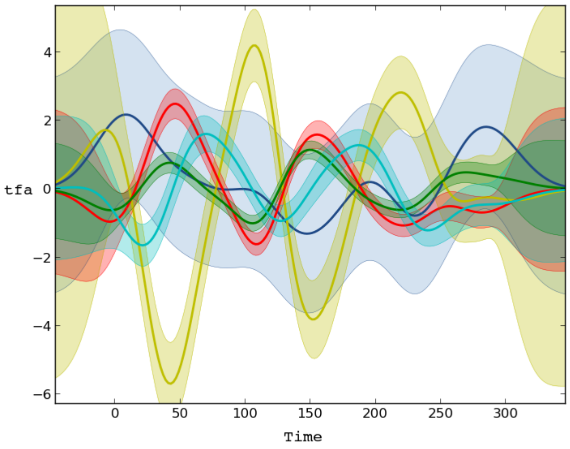

Fig. 8: Variation of activities of different transcription factors with exponentiated quadratic kernel and white kernel in additive form. A solid line represents a posterior mean, and shaded area represents the 95% confidence interval for a specific transcription factor. Different colour shows the transcription factor activity of randomly picked transcription factors. Here we have noticed that for every transcription factors the activities are extremely smooth. In practical case this behaviour is hilgly unlikely. So, a combination of exponentiated quadratic kernel and white kernel in additive form is not a very good choice to infer the transcription factor activity.

and the covariance given by

$$\boldsymbol{C}_g = \left[\mathbf{K}_{t_\star,t_\star} \otimes \mathbf{\Lambda}\mathbf{V}^T\mathbf{\Sigma}\mathbf{V}\mathbf{\Lambda}\right] - \mathbf{K}_{t_\star,t} \otimes \mathbf{\Lambda}\mathbf{V}^T\mathbf{\Sigma}\mathbf{V}\mathbf{\Lambda}\left[\mathbf{K}_{t,t} \otimes \mathbf{\Lambda}\mathbf{V}^T\mathbf{\Sigma}\mathbf{V}\mathbf{\Lambda} + \sigma^2\mathbf{I}\right]^{-1} \mathbf{K}_{t_\star,t} \otimes \mathbf{\Lambda}\mathbf{V}^T\mathbf{\Sigma}\mathbf{V}\mathbf{\Lambda} \tag{68}$$

Finally the posterior mean of the conditional distribution is is

$$\boldsymbol{\mu}_F = \mathbf{K}_{t_\star,t} \otimes \mathbf{\Sigma}\mathbf{V}\mathbf{\Lambda}\left[\mathbf{K}_{t,t} \otimes \mathbf{\Lambda}\mathbf{V}^T\mathbf{\Sigma}\mathbf{V}\mathbf{\Lambda} + \sigma^2\mathbf{I}\right]^{-1} \mathbf{y}_q \tag{69}$$

and the covariance of the conditional distribution is

$$\boldsymbol{C}_F = \mathbf{K}_{t_\star,t_\star} \otimes \mathbf{\Sigma} - \mathbf{K}_{t_\star,t} \otimes \mathbf{\Sigma}\mathbf{V}\mathbf{\Lambda}\left[\mathbf{K}_{t,t} \otimes \mathbf{\Lambda}\mathbf{V}^T\mathbf{\Sigma}\mathbf{V}\mathbf{\Lambda} + \sigma^2\mathbf{I}\right]^{-1} \left[\mathbf{K}_{t_\star,t} \otimes \mathbf{\Lambda}\mathbf{V}^T\mathbf{\Sigma}\right] \tag{70}$$

Figure 7 shows the pictorial representation of intrinsic coregionalization kernel (Equation 48) $\mathbf{K}_f$ considering 5 transcription factors where covariance matrix $\mathbf{\Sigma}$ of was constructed using Ornstein-Uhlenbeck kernel and white kernel in additive form.

The preliminary concepts of this methodology was presented in [36] and a Jupyter Notebook demo using programming language *python* is available at [37].

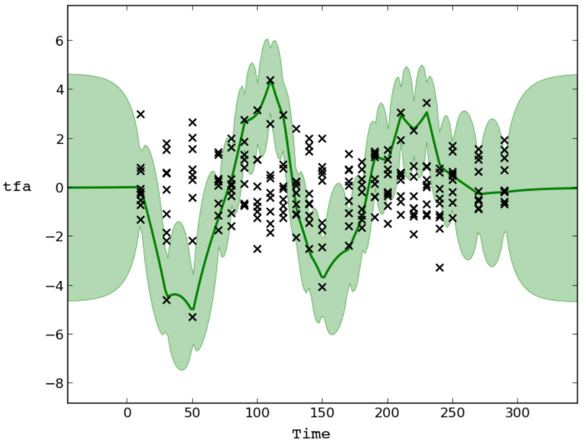

Fig. 9: Transcription factor activity of ACE2. The solid line represents a posterior mean of transcription factor activity, and shaded area represents 95% confidence interval. x-axis represents the time, and y-axis represent the transcription factor activity. Here a 'x' represents a gene expression level. At any time point, multiple gene expression levels ('x') represent the level of gene expression obtained from different genes which were regulated by transcription factor ACE2. We inferred the transcription factor activity considering all these expression levels.

6 Dataset and Result analysis

Here in this experiment we used the classic [27] yeast cell cycle dataset. The cell division cycle gene *cdc15* time series data has 23 time points.

Exponentiated quadratic kernel also known as Radial basis function (RBF) kernel is very smooth kernel compared to Ornstein-Uhlenbeck kernel and perhaps is not a very good choice for the determination of actual transcription factors activities. Still, it can figure out the basic nature of the activities with over smoothness. Figure 8 shows activities of different transcription factors while the model was developed considering exponentiated quadratic kernel with White kernel in additive form.

Figure 9 shows transcription factor activity of ACE2. While developing the model we chose *Ornstein-Uhlenbeck* kernel and White kernel in additive form. In Figure 9, we noticed at any time point, multiple gene expression levels (represented by black 'x') were present, which shows the level of gene expression at that time point from different genes. A transcription factor can regulate multiple genes. To infer the activity of a transcription factor, we have to consider all these expressions. Here, we inferred the transcription factor activity of ACE2 considering all these expression levels. Earlier, we described how Ornstein-Uhlenbeck kernel can be used to model the transcription factor activities. So, here we believe, the Ornstein-Uhlenbeck kernel will consider the basic nature of the transcription factors activity while White kernel will deal the noise associated the collected gene expression data.

Figure 10 shows transcription factor activity of different (arbitrarily chosen) transcription factors where we used our newly developed model. Here, as a prior we used Ornstein-Uhlenbeck kernel and White kernel. In the plot, a solid line represents a posterior mean of transcription factors activity and shaded area represents the 95% confidence interval.

7 Conclusion

The choice of the covariance function is the central step in modelling with a Gaussian process. In this paper, we developed a covariance function suitable for modelling the dynamics of transcription factor activity. We also

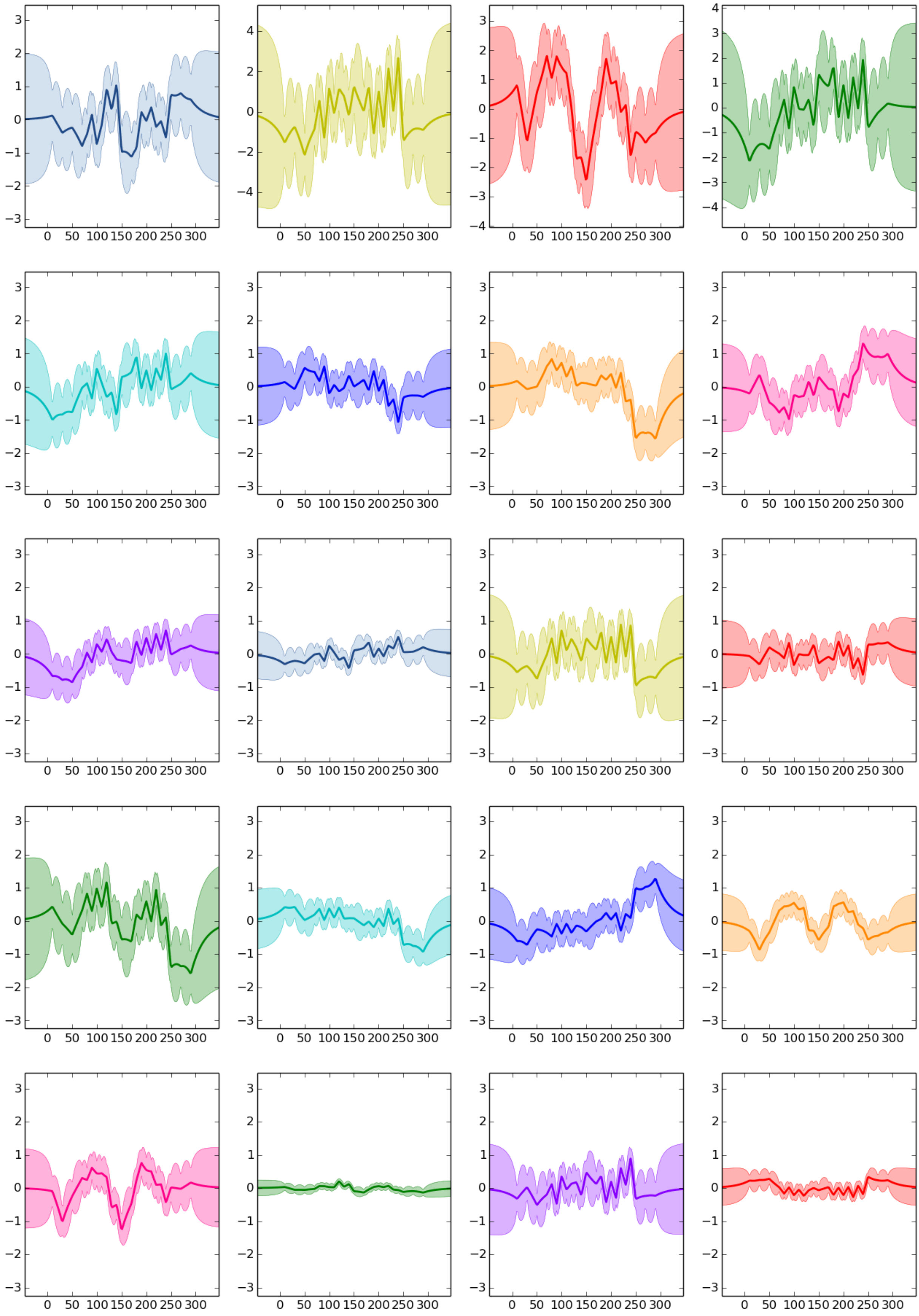

Fig. 10: Transcription factor activity of different transcription factors: individual plots shows the activity of the transcription factor. Here, the solid line represents a posterior mean and shaded area represents the 95% confidence interval. x-axis represents the time and y-axis represent the TFA.

justified the rationale behind choosing the Ornstein-Uhlenbeck kernel to achieve the goal.

We showed that the linear Gaussian model proposed by [26] is equivalent to a Gaussian process with a particular covariance function. In this paper, we, therefore, build a transcription factor activity model straight from the Gaussian process perspective. For given gene expression profiles and a connectivity information between genes and transcription factors, here we designed a covariance function for reconstructing transcription factor activities and overcome the restrictive state dependencies of Markovian assumptions. In the beginning, we assumed that the joint process across all transcription factor activities and across all time points might have some correlation, hence we introduced an intrinsic model of coregionalization for the joint process. We also introduced a mathematical formalism, based on a judicious application of singular value decomposition, to enable us to efficiently fit the Gaussian process in a reduced 'transcription factor activity' space. Finally, we prepared a *jupyter notebook* and made it available online.

Authors and Contributors: This work was carried out in close collaboration between both of the coauthors. NDL preliminarily defined the research theme. MAR contributed an early design of the system and implemented it. NDL refined the system development. MAR wrote the manuscript and NLD edited it. Both of the authors have contributed to, seen and approved the final manuscript.

References

1. A. D. Keller, "Specifying epigenetic states with autoregulatory transcription factors," *Journal of Theoretical Biology*, vol. 170, pp. 175–181, September 1994.
2. B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Molecular Biology of the Cell.* Garland Science, New York, 2002.
3. C. F. Calkhoven and G. AB, "Multiple steps in the regulation of transcription-factor level and activity," *Journal of Biochem.*, vol. 317, pp. 329–342, 1996.
4. M. Schena, D. Shalon, R. W. Davis, and P. O. Brown, "Quantitative monitoring of gene expression patterns with a complementary DNA microarray," *Science*, vol. 270, pp. 467–70, October 1995.
5. B. Ren, F. Robert, J. J. Wyrick, O. Aparicio, E. G. Jennings, I. Simon, J. Zeitlinger, J. Schreiber, N. Hannett, E. Kanin, T. L. Volkert, C. J. Wilson, S. P. Bell, and R. A. Young, "Genome-wide location and function of DNA binding proteins.," *Science*, vol. 290, pp. 2306–2309, 2000.
6. C. E. Horak and M. Snyder, "Chip-chip: A genomic approach for identifying transcription factor binding sites," *Methods in Enzymology*, vol. 350, pp. 469 – 483, 2002. Guide to Yeast Genetics and Molecular and Cell Biology - Part B.
7. I. Simon, J. Barnett, N. Hannett, C. T. Harbison, N. J. Rinaldi, T. L. Volkert, J. J. Wyrick, J. Zeitlinger, D. K. Gifford, T. S. Jaakkola, and R. A. Young, "Serial regulation of transcriptional regulators in the yeast cell cycle," *Cell*, vol. 106, no. 6, pp. 697 – 708, 2001.
8. K. Takahashi and S. Yamanaka, "Induction of pluripotent stem cells from mouse embryonic and adult fibroblast cultures by defined factors," *Cell*, vol. 126, no. 4, pp. 663 – 676, 2006.
9. B. D. Dynlacht, "Regulation of transcription by proteins that control the cell cycle," *Nature*, vol. 389, pp. 149–152, September 1997.
10. T. I. Lee and R. A. Young, "Transcription of eukaryotic protein-coding genes," *Annual Review of Genetics*, vol. 34, pp. 77–137, 2000.
11. J. T. Kadonaga, "Regulation of RNA polymerase II transcription by sequence-specific DNA binding factors," *Cell*, vol. 116, no. 2, pp. 247 – 257, 2004.
12. S. J. Furney, D. G. Higgins, C. A. Ouzounis, and N. López-Bigas, "Structural and functional properties of genes involved in human cancer," *BMC Genomics*, vol. 7, no. 1, p. 3, 2006.
13. S. Boyadjiev and E. Jabs, "Online mendelian inheritance in man (OMIM) as a knowledgebase for human developmental disorders," *Clinical Genetics*, vol. 57, no. 4, pp. 253–266, 2000.
14. S. De, N. Lopez-Bigas, and S. A. Teichmann, "Patterns of evolutionary constraints on genes in humans," *BMC Evolutionary Biology*, vol. 8, no. 1, p. 275, 2008.
15. M. Levine and R. Tjian, "Transcription regulation and animal diversity," *Nature*, pp. 147–151, July 2003.
16. B. Turcotte and L. Guarente, "Hap1 positive control mutants specific for one of two binding sites," *Gene and Development*, vol. 6, pp. 2001–2009, December 1992.
17. D. B. Menke, C. Guenther, and D. M. Kingsley, "Dual hindlimb control elements in the tbx4 gene and region-specific control of bone size in vertebrate limbs," *Development*, vol. 135, no. 15, pp. 2543–2553, 2008.
18. T. R. Hughes and C. G. de Boer, "Mapping yeast transcriptional networks," *Genetics*, vol. 195, no. 1, pp. 9–36, 2013.
19. C. M. Bishop, "Latent variable models," in *Learning in Graphical Models*, pp. 371–403, MIT Press, 1999.
20. N. Lawrence, "Probabilistic non-linear principal component analysis with Gaussian process latent variable models," *Journal of Machine Learning Research*, vol. 6, pp. 1783–1816, Dec. 2005.
21. O. Alter and G. H. Golub, "Integrative analysis of genome-scale data by using pseudoinverse projection predicts novel correlation between DNA replication and RNA transcription," *Proceedings of the National Academy of Sciences USA*, vol. 101, pp. 16577–16582, November 2004.
22. F. Gao, B. C. Foat, and H. J. Bussemaker, "Defining transcriptional networks through integrative modeling of mRNA expression and transcription factor binding data," *BMC Bioinformatics*, vol. 5, March 2004.
23. J. C. Liao, R. Boscolo, Y.-L. Yang, L. M. Tran, C. Sabatti, and V. P. Roychowdhury, "Network component analysis: Reconstruction of regulatory signals in biological systems," *Proceedings of the National Academy of Sciences*, vol. 100, December 2003.
24. A.-L. Boulesteix and K. Strimmer, "Predicting transcription factor activities from combined analysis of microarray and chip data: a partial least squares approach," *Theoretical Biology and Medical Modelling*, vol. 2, June 2005.

25. I. Nachman, A. Regev, and N. Friedman, "Inferring quantitative models of regulatory networks from expression data," *Bioinformatics*, vol. 20, March 2004.
26. G. Sanguinetti, M. Rattray, and N. D. Lawrence, "A probabilistic dynamical model for quantitative inference of the regulatory mechanism of transcription," *Bioinformatics, Oxford University Press*, vol. 22, no. 14, pp. 1753–1759, 2006.
27. P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher, "Comprehensive identification of cell cycle regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization," *Molecular Biology of the Cell*, vol. 9, pp. 3273–3297, December 1998.
28. T. I. Lee, N. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. T. Harbison, C. M. Thompson, I. Simon, J. Zeitlinger, E. G. Jennings, H. L. Murray, D. B. Gordon, B. Ren, J. J. Wyrick, J.-B. Tagne, T. L. Volkert, E. Fraenkel, D. K. Gifford, and R. A. Young, "Transcriptional regulatory networks in saccharomyces cerevisiae," *Science*, vol. 298, no. 5594, pp. 799–804, 2002.
29. C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning.* MIT Press, 2006.
30. D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.
31. M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions.* Dover, New York, 1965.
32. G. E. Uhlenbeck and L. S. Ornstein, "On the theory of the brownian motion," *Phys. Rev.*, vol. 36, pp. 823–841, 1930.
33. H. Wackernagel, *Multivariate Geostatistics An Introduction with Applications.* Springer-Verlag Heidelberg New York, 2003.
34. M. A. Alvarez, L. Rosasco, and N. D. Lawrence, "Kernels for vector-valued functions: A review," *Foundations and Trends in Machine Learning*, vol. 4, June 2012.
35. O. Stegle, C. Lippert, J. M. Mooij, N. D. Lawrence, and K. M. Borgwardt, "Efficient inference in matrix-variate Gaussian models with iid observation noise," in *Advances in Neural Information Processing Systems 24* (J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, eds.), pp. 630–638, Curran Associates, Inc., 2011.
36. M. A. Rahman and N. D. Lawrence, "A gaussian process model for inferring the dynamic transcription factor activity," in *Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, BCB '16, (New York, NY, USA), pp. 495–496, ACM, 2016.
37. M. A. Rahman and N. D. Lawrence, "Inferring transcription factor activities by combining binding information with gene expression profiles." `http://nbviewer.jupyter.org/github/SheffieldML/notebook/blob/master/compbio/transcriptionFactorActivities.ipynb`, 2014.