

Gaussian Processes and Gene Regulation



James W J Anderson and Chris T L Choy

University of Oxford

Department of Statistics

Summer 2009

Abstract

Many factors affect the production of proteins in the cells; here we consider the regulation at the transcription level, where transcription factors (TFs) affect the level of copying from DNA to mRNA, and hence the gene expression. Being able to understand the relationship between the TFs and the corresponding gene expression is important. However, it is not the case that TFs only regulate for a single type of gene expression. Similarly, the way a gene will express itself will affect the way other genes express themselves. The regulatory network is intricate and complex.

In this project, we investigated how we can infer underlying transcription factor levels using Gaussian Process. We extended the model given by Lawrence et al. [3] to multiple transcription factors. The relevant covariance matrices and mean functions are computed for the case where the gene expression depends linearly with the transcription factor level and an algorithm is provided for non-linear case.

Contents

1	Introduction	5
1.1	Motivation	5
1.2	The Model	5
1.3	Gaussian Processes	5
1.4	Why this model?	6
1.5	Inference with a single TF	9
2	Multiple TFs	11
2.1	Introduction	11
2.2	Two Transcription Factors, Two Genes	11
2.2.1	The Easy Case: $D_1 = D_2 = D$	12
2.2.2	The Less Easy Case: $D_1 \neq D_2$	15
2.3	Generalising: m Transcription Factors, n Genes	18
2.3.1	If $m \leq n$	18
2.3.2	If $m > n$	19
3	General Network Models	20
3.1	Network Model 1	20
3.2	Network Model 2	23
3.3	Generalising Case 1: The Network is <i>closed</i>	27
3.4	Generalising Case 2: More dependent genes than independent transcription factors	27
3.5	Speeding up the computation	29
3.5.1	The Problem	29
3.5.2	Ways to increase efficiency	29
4	Non-linear model	31
4.1	Introduction	31
4.1.1	Aim	31
4.2	Derivation of $\nabla \log p(y f)$ and $\nabla \nabla \log p(y f)$	32
4.2.1	log likelihood	32
4.2.2	The partial derivatives of x	32
4.2.3	Compute functions involving f	33
4.3	Generalize this method	33
4.4	Simulation	34
4.4.1	Result	37
4.5	Non-linear case	37
4.5.1	Avoid computing K^{-1}	37
4.6	Implementation details - matrices	39
4.7	Function form of g	40
4.7.1	Simple Cases	40
4.7.2	Michaelis-Menten kinetics	40
4.7.3	Multiple transcription factors	41

5	Estimation of model parameters	42
5.1	Marginal Likelihood	42
5.1.1	Different sets of parameters explaining the data equally well	45
5.2	MCMC analysis	48
5.2.1	prior	48
5.2.2	Proposal density	48
5.3	Multiple transcription factors	49
6	Future Work	51
6.1	More efficient way to maximize the marginal likelihood	51
6.2	Better accuracy	51
6.3	Known TF concentrations	51
6.4	Binding Energy Landscapes of Transcription Factors	53
6.5	Potential Data for Analysis	53
7	Conclusion	54

Chapter 1

Introduction

1.1 Motivation

Gene regulation occurs in all cells to monitor the production of different proteins. If all cells were to produce exactly the same cocktail of proteins, each cell would have the same function and perform identical operations. Evidently, they do not. Many factors affect the production of proteins in the cells; here we consider the regulation at the transcription level, where transcription factors (TFs) affect the level of copying from DNA to mRNA, and hence the gene expression.

Naturally, being able to understand the relationship between the TFs and the corresponding gene expression is important. However, it is not the case that TFs only regulate for a single type of gene expression. Similarly, the way a gene will express itself will affect the way other genes express themselves. The regulatory network is intricate and complex.

1.2 The Model

Many mathematical models have been suggested to model the relationship between TF concentration and gene expression. We focus on the following ordinary differential equation:

$$\frac{dx_j}{dt} = B_j + S_j g(f(t)) - D_j x_j(t) \quad \text{with} \quad x_j(0) = \frac{B_j}{D_j} \quad (1.1)$$

where $x_j(t)$ is the expression of gene j at time t , B_j is the basal transcription rate of gene j , D_j the decay rate of mRNA, S_j the sensitivity of gene j to the transcription factor, g a function and $f(t)$, the concentration of the transcription factor.

By solving above differential equation, we obtain the equation for the gene expression level

$$x_j(t) = \frac{B_j}{D_j} + S_j e^{-D_j t} \int_0^t g(f(u)) e^{D_j u} du \quad (1.2)$$

1.3 Gaussian Processes

A Gaussian Process $X = (X_t)_{t \geq 0}$ on a domain D is a stochastic process such that for any finite $S \subseteq D$, S has a Multivariate Normal Distribution. In this model we have $D = \mathbb{R}$.

Gaussian Processes are uniquely characterised by their mean function $m(x)$ and their covariance function $k(x, x')$. We write $f(t) = \mathcal{GP}(m(x), k(x, x'))$. For example, Brownian Motion is a GP with mean function 0 and covariance function $k(x, x') = \min(x, x')$.

In our model, we assume that the concentration of the transcription factors $f(t)$ follows the Gaussian Process. So what mean and covariance functions do we give f ? Lawrence et al. [3] suggest $f(t) = \mathcal{GP}(0, \exp(-\frac{(t-t')^2}{\ell^2}))$; we will also use this. The covariance function suggests that time points close together have a large covariance, but time points faraway have close to zero. This reflects what we might expect in a setting such as this.

1.4 Why this model?

We would like to be able to make observations of gene expression levels and infer transcription factor concentration. GPs provide a convenient framework for this to be done.

Versatility

GPs form a flexible non-parametric regression model which will fit the data in most cases. For example, suppose we have observed the points in Figure 1.1. We can then fit a GP regression curve, with the black line indicating the mean and with the dotted lines as 95% confidence lines as in Figure 1.2. Typically our observations will have a normally distributed noise function. GP regression can be modified simply to account for this (Figure 1.3). Gaussian process can model an underlying smooth function naturally and it is more realistic than other Markov models which can only provide point estimates (such as Khanin et al. [2]).

Computability

It is computationally quick to then calculate a posterior distribution for f with mean \bar{f}_{post} and covariance function K_{ff}^{post} given observations \mathbf{x} , as set out in Lawrence et al. [3] and Rasmussen and Williams [7],

$$\bar{f}_{post} = K_{f\mathbf{x}}K_{\mathbf{x}\mathbf{x}}^{-1}(\mathbf{x} - \mathbf{m}(\mathbf{x})) \quad (1.3)$$

$$K_{ff}^{post} = K_{ff} - K_{f\mathbf{x}}K_{\mathbf{x}\mathbf{x}}^{-1}K_{\mathbf{x}f} \quad (1.4)$$

where K is the covariance matrix obtained by evaluating the covariance function on every pair of observed time points.

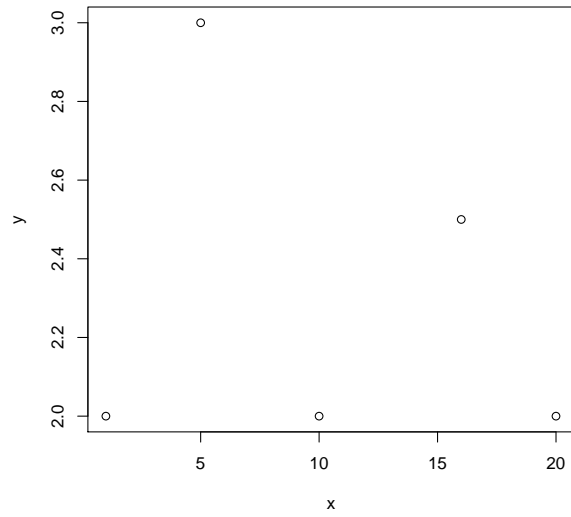


Figure 1.1: Observed Points

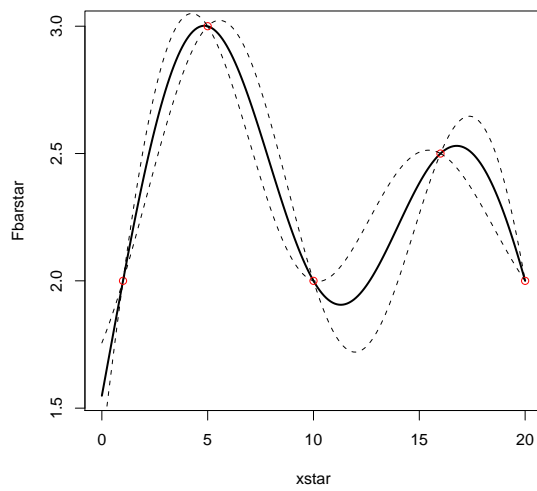


Figure 1.2: Fitting a GP regression curve

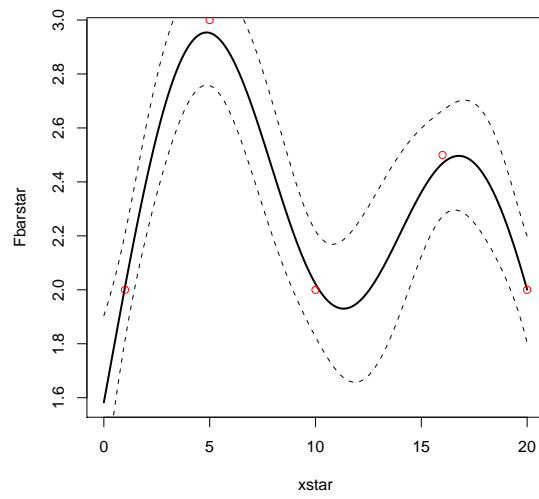


Figure 1.3: Fitting a GP regression with noise

1.5 Inference with a single TF

Lawrence et al. [3] have developed the result for a single transcription factor. Assuming we have $x_j(t)$ a GP, we can calculate the mean and covariance functions

$$m(x_j(t)) = \frac{B_j}{D_j} \quad (1.5)$$

$$\text{Cov}[z_i(t), z_j(t')] = S_i S_j \frac{\sqrt{\pi} \ell}{2} [h_{ij}(t, t') + h_{ji}(t', t)] \quad (1.6)$$

where

$$h_{ij}(t', t) = \frac{\exp(\gamma_i^2)}{D_i + D_j} \left(\exp[D_i(t' - t)] \left[\text{erf}\left(\frac{t' - t}{\ell} - \gamma_i\right) + \text{erf}\left(\frac{t}{\ell} + \gamma_i\right) \right] - \exp[-(D_i t' + D_j t)] \left[\text{erf}\left(\frac{t'}{\ell} - \gamma_i\right) + \text{erf}(\gamma_i) \right] \right)$$

for $\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-y^2) dy$ and $\gamma_i = \frac{D_i \ell}{2}$.

Also

$$\text{Cov}[z_j(t'), f_i(t)] = \frac{\sqrt{\pi} \ell_j S_j}{2} \exp(\gamma_i^2) \exp(-D_j(t' - t)) \left[\text{erf}\left(\frac{t' - t}{\ell} - \gamma_i\right) + \text{erf}\left(\frac{t}{\ell} + \gamma_i\right) \right] \quad (1.7)$$

We can then calculate the covariance matrices and infer f as in equations 1.3 and 1.4.

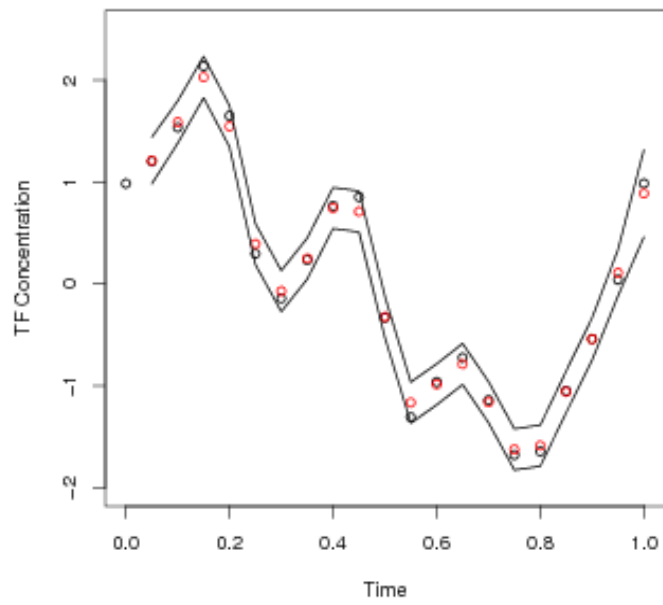


Figure 1.4: Simulated f (black) and inferred f (red) with parameters $B = 1$, $D = 1$, $S = 1$, $\ell = 0.1$, $\sigma = 0.05$. The lines indicate the 95% confidence region.

Chapter 2

Multiple TFs

2.1 Introduction

We now look at the case where we have multiple transcription factors which have a linear effect on the rate of change of the gene expression level (i.e. $g(x) = x$). In this section we will first look at a simple case of 2 transcription factors, 2 genes to get a flavour for what has changed, and then develop techniques to generalise this to provide an analytic solution for inference.

In each case, we are working with a model

$$\frac{dx_j}{dt} = B_j + \sum_i S_{ij} f_i(t) - D_j x_j(t), \quad \text{with } x_j(0) = \frac{B_j}{D_j} \quad (2.1)$$

2.2 Two Transcription Factors, Two Genes

Here our network looks like

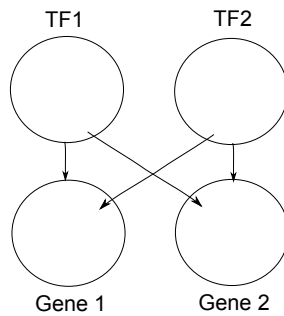


Figure 2.1: We have f_1, f_2 (unknown) acting as transcription factors for x_1, x_2 (observed).

$$x'_1(t) = B_1 + S_{11}f_1(t) + S_{12}f_2(t) - D_1x_1(t) \quad (2.2)$$

$$x'_2(t) = B_2 + S_{21}f_1(t) + S_{22}f_2(t) - D_2x_2(t) \quad (2.3)$$

These give

$$x_1(t) = \frac{B_1}{D_1} + \exp(-D_1 t) \int_0^t (S_{11} f_1(t) + S_{12} f_2(t)) \exp(D_1 u) du \quad (2.4)$$

$$x_2(t) = \frac{B_2}{D_2} + \exp(-D_2 t) \int_0^t (S_{21} f_1(t) + S_{22} f_2(t)) \exp(D_2 u) du \quad (2.5)$$

As before, we want to make inferences on f given observations of $x_1(t)$ and $x_2(t)$. The key to this inference is conditioning on the joint multivariate normal of x and f . However, we now have equations in the form where this is no longer possible. We need to rearrange so that we have a set of equations, each one involving only a single unknown transcription factor. Then we can continue as before.

2.2.1 The Easy Case: $D_1 = D_2 = D$

This is the much easier case to deal with, since we can simply take linear combinations of $x_1(t)$ and $x_2(t)$ to untangle the effect of each f :

$$S_{21}x_1(t) - S_{11}x_2(t) = \frac{S_{21}B_1 - S_{11}B_2}{D} + \exp(-Dt) \int_0^t (S_{12}S_{21} - S_{22}S_{11})e^{Du} f_2(u) du \quad (2.6)$$

$$S_{12}x_2(t) - S_{22}x_1(t) = \frac{S_{12}B_2 - S_{22}B_1}{D} + \exp(-Dt) \int_0^t (S_{12}S_{21} - S_{22}S_{11})e^{Du} f_1(u) du \quad (2.7)$$

So given observations \mathbf{x} of $x_1(t)$ and $x_2(t)$, we also have observations \mathbf{z} of $z_1(t) = S_{21}x_1(t) - S_{11}x_2(t)$ and $z_2(t) = S_{12}x_2(t) - S_{22}x_1(t)$ (we assume here that at a time of observation we observe both x_1 and x_2). $z_1(t)$ and $z_2(t)$ are also Gaussian Processes since the sum and scalar product of normal distributions is normal. We can then apply the equations in the one transcription factor case.

We again assume the prior for the $k(f_i(t), f_j(t'))$ to be $\delta_{ij} \exp\left(-\frac{(t-t')^2}{\ell_i^2}\right)$. Hence

$$\text{Cov}[z_i(t), z_j(t')] = \delta_{ij} (S_{12}S_{21} - S_{22}S_{11})^2 \frac{\sqrt{\pi}\ell}{4} [h_{ij}(t, t') + h_{ji}(t', t)] \quad (2.8)$$

where

$$h_{ij}(t', t) = \frac{\exp(\gamma_i^2)}{D} \left(\exp[D(t' - t)] \left[\text{erf}\left(\frac{t'-t}{\ell_i} - \gamma_i\right) + \text{erf}\left(\frac{t}{\ell_i} + \gamma_i\right) \right] - \exp[-(Dt' + Dt)] \left[\text{erf}\left(\frac{t'}{\ell_i} - \gamma_i\right) + \text{erf}(\gamma_i) \right] \right)$$

for $\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-y^2) dy$ and $\gamma_i = \frac{D\ell_i}{2}$.

Also

$$\text{Cov}[z_j(t'), f_i(t)] = \delta_{ij} \frac{\sqrt{\pi}\ell_j (S_{12}S_{21} - S_{22}S_{11})}{2} \exp(\gamma_i^2) \exp(-D(t' - t)) \left[\text{erf}\left(\frac{t' - t}{\ell_i} - \gamma_i\right) + \text{erf}\left(\frac{t}{\ell_i} + \gamma_i\right) \right] \quad (2.9)$$

We use

$$f_i^{post} = K_{f_i z_i} K_{z_i z_i}^{-1} (z_i - a_i) \quad (2.10)$$

$$K_{f_i f_i}^{post} = K_{f_i f_i} - K_{f_i z_i} K_{z_i z_i}^{-1} K_{z_i f_i} \quad (2.11)$$

where $a_1 = (S_{21}B_1 - S_{11}B_2)/D$, $a_2 = (S_{12}B_2 - S_{22}B_1)/D$ are the mean functions of z_1 , z_2 respectively.

The figures that follow are data simulated for this model with parameters $\ell_1 = \ell_2 = 0.1$, $D = 1$, $S_{11} = 1$, $S_{12} = 2$, $S_{21} = 3$, $S_{22} = 1.5$, $B_1 = 1$, $B_2 = 1.5$.

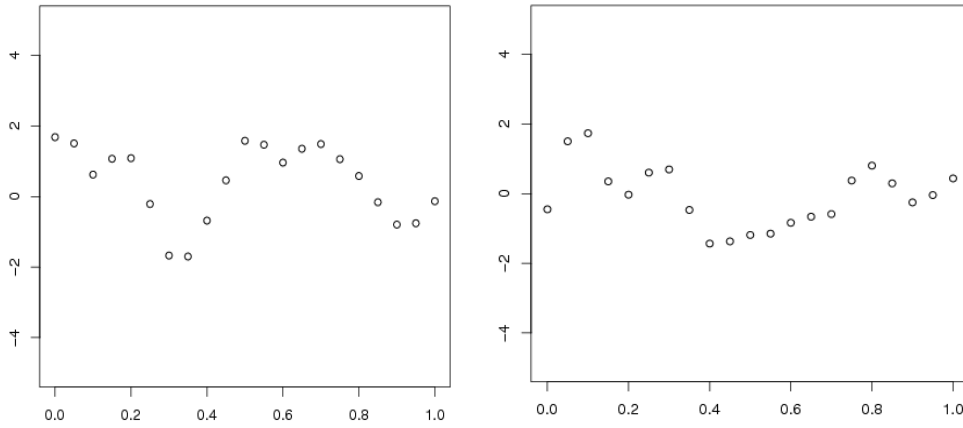


Figure 2.2: Simulated f_1 (left) and f_2 (right).

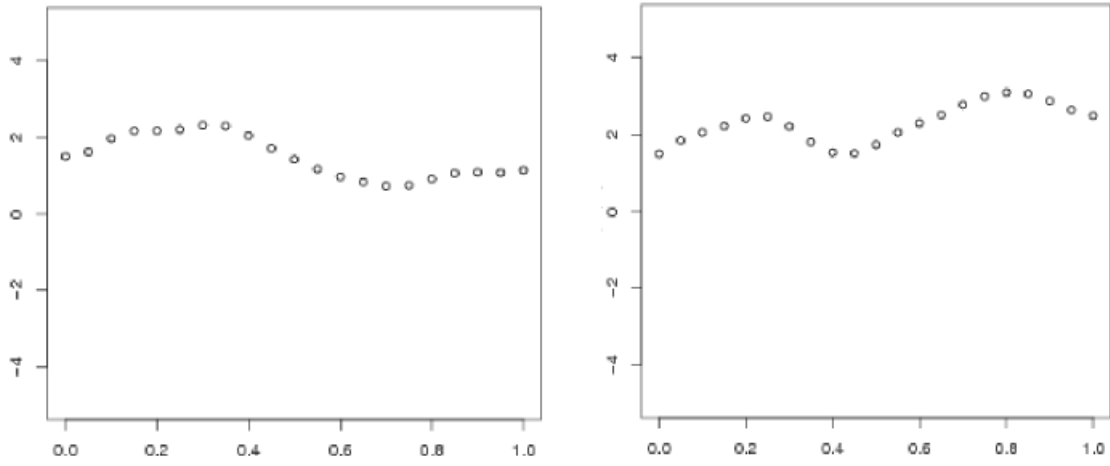


Figure 2.3: Corresponding z_1 (left) and z_2 (right).

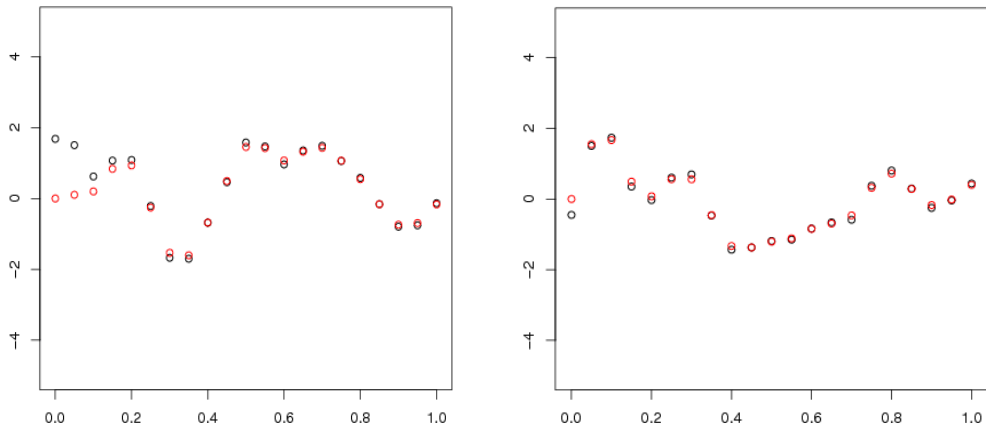


Figure 2.4: Left: Simulated f_1 (black) with inferred f_1 (red). Right: Simulated f_2 (black) with inferred f_2 (red.)

2.2.2 The Less Easy Case: $D_1 \neq D_2$

Here is the first point where we actually run into some problems and have to get creative. Given observations of x_1, x_2 we want to be able to make inferences for f , but each x_i is a function of both f_1 and f_2 . Here we can use the Laplace Transform to rearrange our equations accordingly.

$$L[x_1](p) = \frac{B_1}{D_1 p} + \frac{1}{p + D_1} (S_{11}L[f_1](p) + S_{12}L[f_2](p)) \quad (2.12)$$

$$L[x_2](p) = \frac{B_2}{D_2 p} + \frac{1}{p + D_2} (S_{21}L[f_1](p) + S_{22}L[f_2](p)) \quad (2.13)$$

Eliminating f_1 :

$$(p + D_1)S_{21}L[x_1](p) - (p + D_2)S_{11}L[x_2](p) = \frac{(p + D_1)S_{21}B_1}{D_1 p} - \frac{(p + D_2)S_{11}B_2}{D_2 p} + (S_{12}S_{21} - S_{11}S_{22})L[f_2](p) \quad (2.14)$$

then

$$\frac{S_{21}L[x_1](p)}{p + D_2} - \frac{S_{11}L[x_2](p)}{p + D_1} = \frac{S_{21}B_1}{D_1 p(p + D_2)} - \frac{S_{11}B_2}{D_2 p(p + D_1)} + \frac{S_{12}S_{21} - S_{22}S_{11}}{(p + D_2)(p + D_1)}L[f_2](p) \quad (2.15)$$

Then inverting the Laplace Transform

$$\begin{aligned} \int_0^t S_{21}e^{-D_2(t-u)}x_1(u) du - \int_0^t S_{11}e^{-D_1(t-u)}x_2(u) du &= \frac{S_{21}B_1 - S_{11}B_2}{D_1 D_2} + \frac{S_{11}B_2}{D_1 D_2}e^{-D_1 t} - \frac{S_{21}B_1}{D_1 D_2}e^{-D_2 t} \\ &+ \frac{S_{12}S_{21} - S_{22}S_{11}}{(D_1 - D_2)} \left(\int_0^t (e^{-D_2(t-u)} - e^{-D_1(t-u)}) f_2(u) du \right) \end{aligned}$$

We get a similar result for f_1 :

$$\begin{aligned} \int_0^t S_{12}e^{-D_1(t-u)}x_2(u) du - \int_0^t S_{22}e^{-D_2(t-u)}x_1(u) du &= \frac{S_{12}B_2 - S_{22}B_1}{D_1 D_2} + \frac{S_{22}B_1}{D_1 D_2}e^{-D_2 t} - \frac{S_{12}B_2}{D_1 D_2}e^{-D_1 t} \\ &+ \frac{S_{12}S_{21} - S_{22}S_{11}}{(D_1 - D_2)} \left(\int_0^t (e^{-D_2(t-u)} - e^{-D_1(t-u)}) f_1(u) du \right) \end{aligned}$$

Why is this a useful form to have equations 2.4 and 2.5 in? Given observations of $x_1(t)$ we make a numerical approximation to $\bar{z}_1(t) := \int_0^t e^{-D_2(t-u)}x_1(u) du$, resp. $\bar{z}_2(t)$ for $x_2(t)$, using these observations. We can then write our equation in the form

$$z_1(t) = \frac{S_{12}B_2 - S_{22}B_1}{D_1 D_2} - \frac{S_{12}B_2}{D_1 D_2}e^{-D_1 t} + \frac{S_{22}B_1}{D_1 D_2}e^{-D_2 t} + \frac{S_{12}S_{21} - S_{22}S_{11}}{(D_1 - D_2)} \left(\int_0^t (e^{-D_2(t-u)} - e^{-D_1(t-u)}) f_1(u) du \right) \quad (2.16)$$

With $z_1(t) := S_{12}\bar{z}_2(t) - S_{22}\bar{z}_1(t)$ and $z_2(t) := S_{21}\bar{z}_1(t) - S_{11}\bar{z}_2(t)$. Viewing these as our observations of the gene expression, we proceed with calculating the mean and covariance functions.

$$m[z_1(t)] = \frac{S_{12}B_2 - S_{22}B_1}{D_1D_2} + \frac{S_{22}B_1}{D_1D_2}e^{-D_2t} - \frac{S_{12}B_2}{D_1D_2}e^{-D_1t} \quad (2.17)$$

$$m[z_2(t)] = \frac{S_{21}B_1 - S_{11}B_2}{D_1D_2} + \frac{S_{11}B_2}{D_1D_2}e^{-D_1t} - \frac{S_{21}B_1}{D_1D_2}e^{-D_2t} \quad (2.18)$$

$$\text{Cov}[z_i(t), z_j(t')] = \left(\frac{S_{12}S_{21} - S_{22}S_{11}}{(D_1 - D_2)} \right)^2 \int_0^{t'} \int_0^t (e^{-D_2(t-u)} - e^{-D_1(t-u)}) (e^{-D_2(t'-u')} - e^{-D_1(t'-u')}) \text{Cov}[f_i(u), f_j(u')] du du' \quad (2.19)$$

We start with the assumption that the transcription factors are independent, i.e. $\text{Cov}[f_1(t), f_2(t')] = 0$. This assumption can be easily removed, one would simply add the cross covariance terms where necessary. We have not done this here as we do not currently have a biological precedence for what this might be. Also, it makes the calculation more complicated. This gives:

$$\text{Cov}[z_i(t), z_j(t)] = \delta_{ij} \frac{\sqrt{\pi}\ell_i}{2} \left(\frac{S_{12}S_{21} - S_{22}S_{11}}{(D_1 - D_2)} \right)^2 \left(\sum_1^2 (h_{ii}(t, t') + h_{ii}(t', t)) - h_{12}(t, t') - h_{12}(t', t) - h_{21}(t, t') - h_{21}(t', t) \right) \quad (2.20)$$

$$\text{Cov}[z_i(t), f_j(t')] = \delta_{ij} \left(\frac{S_{12}S_{21} - S_{22}S_{11}}{(D_1 + D_2)} \right) \left(\int_0^t (e^{-D_2(t-u)} - e^{-D_1(t-u)}) \text{Cov}[f_i(u), f_j(t')] du \right) \quad (2.21)$$

$$\begin{aligned} \text{Cov}[z_i(t), f_j(t')] &= \delta_{ij} \frac{\sqrt{\pi}(S_{12}S_{21} - S_{22}S_{11})}{2(D_1 - D_2)} \ell_1 \exp(\gamma_1^2) \exp(-D_1(t' - t)) \left[\text{erf}\left(\frac{t' - t}{\ell_1} - \gamma_1\right) - \text{erf}\left(\frac{t}{\ell_1} + \gamma_1\right) \right] \\ &\quad + \frac{\sqrt{\pi}(S_{12}S_{21} - S_{22}S_{11})}{2(D_1 - D_2)} \ell_2 \exp(\gamma_2^2) \exp(-D_2(t' - t)) \left[\text{erf}\left(\frac{t' - t}{\ell_2} - \gamma_2\right) + \text{erf}\left(\frac{t}{\ell_2} + \gamma_2\right) \right] \end{aligned}$$

We have:

$$f_i^{post} = K_{f_i z_i} K_{z_i z_i}^{-1} (z_i - m[z_i]) \quad (2.22)$$

$$K_{f_i f_i}^{post} = K_{f_i f_i} - K_{f_i z_i} K_{z_i z_i}^{-1} K_{z_i f_i} \quad (2.23)$$

Again, we have some simulated data for this model, with parameters as in Case 1, but now with $D_2 = 1.5$.

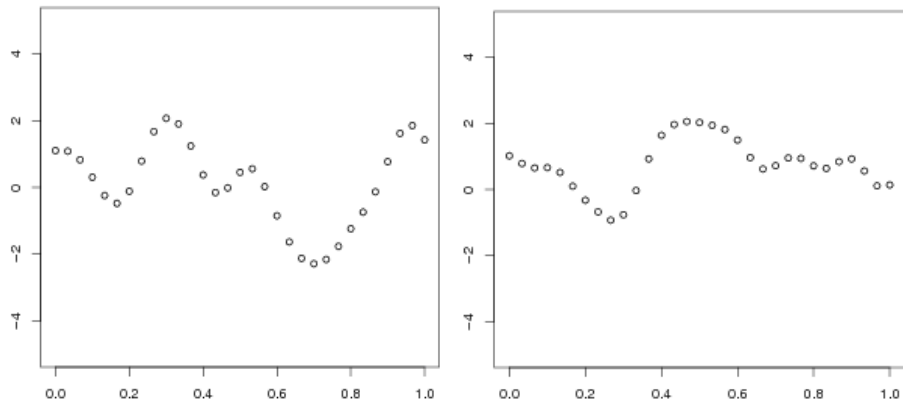


Figure 2.5: Simulated f_1 (left) and f_2 (right).

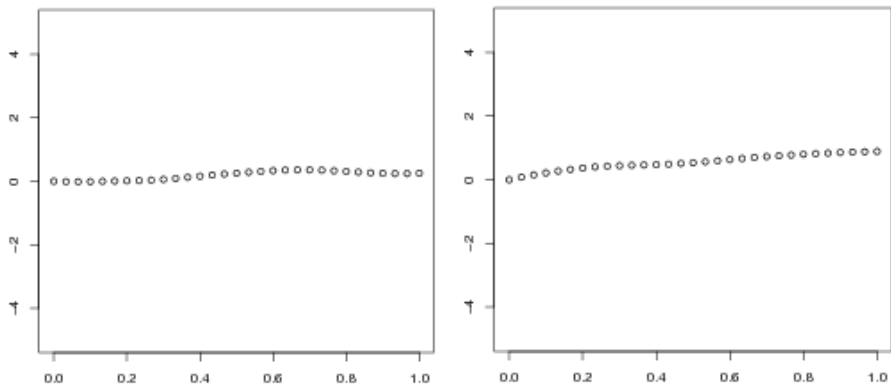


Figure 2.6: Corresponding z_1 (left) and z_2 (right).

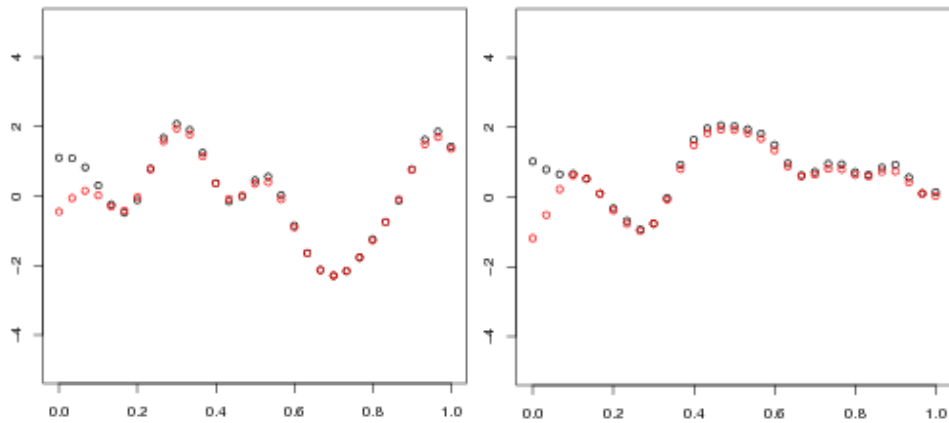


Figure 2.7: Left: Simulated f_1 (black) with inferred f_1 (red). Right: Simulated f_2 (black) with inferred f_2 (red.)

2.3 Generalising: m Transcription Factors, n Genes

2.3.1 If $m \leq n$

$$\mathbf{x}' = \mathbf{B} + \mathbf{S}\mathbf{f} - \mathbf{D}\mathbf{x} \quad (2.24)$$

where $\mathbf{x} = (x_1, \dots, x_m)^T$, $\mathbf{f} = (f_1, \dots, f_n)^T$, \mathbf{B} is a constant vector of the growth rates B_i , \mathbf{S} a constant matrix of the sensitivities S_{ij} , \mathbf{D} a diagonal matrix of decay rates D_i . We then take Laplace transforms:

$$pL[\mathbf{x}] = \frac{\mathbf{B}}{p} + \mathbf{S}L[\mathbf{f}] - \mathbf{D}L[\mathbf{x}] \quad (2.25)$$

Moving all of the $L[x]$ to the left hand side we obtain for a matrix \mathbf{K}

$$\mathbf{K}L[\mathbf{x}] = \frac{\mathbf{B}}{p} + \mathbf{S}L[\mathbf{f}] \quad (2.26)$$

Unless we have the same number of dependent genes as transcription factors we will not have that M is invertible. So we perform row and column operations via a matrix \mathbf{E} until we have

$$\mathbf{E}\mathbf{K}L[\mathbf{x}] = \mathbf{E}\frac{\mathbf{B}}{p} + \mathbf{H}L[\mathbf{f}] \quad (2.27)$$

where \mathbf{H} is a matrix of the form:

$$\begin{pmatrix} \mathbf{I}_n \\ \mathbf{0} \end{pmatrix}$$

We now think about how to invert the Laplace transform. We note that all of the entries in \mathbf{K} are constants (with respect to p), except some of the diagonals which are of the form $k_{ii} = p - D_i$. \mathbf{E} is a matrix obtained by performing row and column operations on \mathbf{S} . \mathbf{S} is a constant matrix, so \mathbf{E} will also be a constant matrix. We divide the equation by $\prod_{i=0}^{k+l} (p - D_i)$ so that we can easily invert the transform.

each term in the equation is of the form

$$\frac{\alpha}{(p - D_1)^{\delta_1^\alpha} \dots (p - D_n)^{\delta_n^\alpha}}$$

Where for each α , $\delta_i^\alpha = 0$ or 1 .

We have

$$\frac{1}{(p - D_1) \dots (p - D_n)} = \sum_{i=1}^{k+1} \frac{A_i}{p - D_i} \quad (2.28)$$

Where $A_i = \prod_{j \neq i}^{k+l} \frac{1}{D_j - D_i}$. This result enables us to use partial fractions to get (4) into a form that we can invert easily. Using

$$L^{-1} \left(\frac{L[x_i]}{p - D_j} \right) = e^{-D_j t} \int_0^t e^{D_j u} x_i(u) du \quad (2.29)$$

and defining $g_i(t) = \exp(D_i t)$ we obtain an expression for $k = 1 \dots n$ of the form

$$\sum_{i,j=1}^n b_{ijk} e^{-D_j t} \int_0^t e^{D_j u} x_i(u) du = c_k + \sum_{i=1}^n d_{ik} e^{D_i t} + \sum_{j=1}^n a_{jk} e^{-D_j t} \int_0^t e^{D_j u} f_k(u) du \quad (2.30)$$

which is now a form from which we can find the mean and covariance matrices of each f_k . From this we then numerically integrate terms on the left hand side to consider observations of the form

$$z_k(t) = \sum_{i,j=1}^n b_{ijk} e^{-D_j t} \int_0^t e^{D_j u} x_i(u) du \quad (2.31)$$

We then can calculate the mean and covariance functions of $z_k(t)$, assuming transcription factors are independent:

$$m(z_k(t)) = c_k + \sum_{i=1}^n d_{ik} e^{D_i t} \quad (2.32)$$

$$\text{Cov}(z_p(t), z_q(t')) = \delta_{pq} \sum_{i,j=1}^m a_{jp} a_{iq} e^{-D_j t - D_i t'} \int_0^t \int_0^{t'} \exp(D_j u + D_i u') \exp\left(-\left(\frac{u - u'}{l_p}\right)^2\right) du' du \quad (2.33)$$

$$\text{Cov}(z_p(t), f_q(t')) = \delta_{pq} \sum_{j=1}^m a_{jp} e^{-D_j t} \int_0^t \exp(D_j u) \exp\left(-\left(\frac{u - t'}{l_p}\right)^2\right) du \quad (2.34)$$

We can then use equations 2.22 and 2.23 to obtain inferred transcription factor concentrations. Noise can be added into the system if desired in exactly the same way as in Lawrence et al. [3].

2.3.2 If $m > n$

Unfortunately in this case we cannot untangle the effect of the TFs. We can infer the cumulative effect of the TFs as before, but it is not possible to infer each individual TF.

Chapter 3

General Network Models

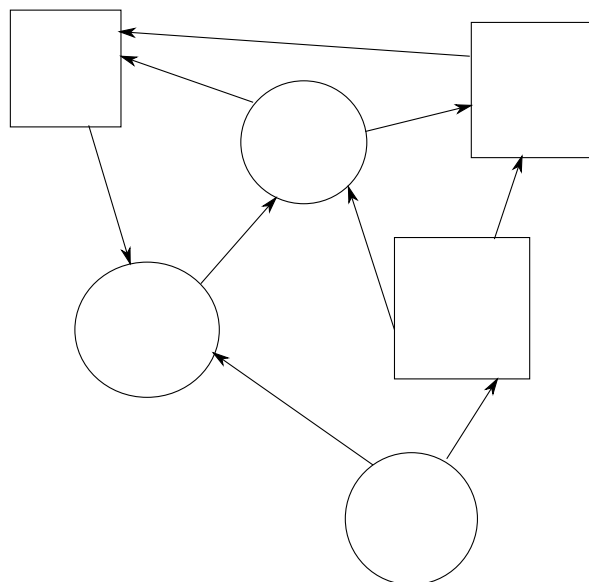


Figure 3.1: We would now like to consider more complicated networks. The circles represent the unknown transcription factor concentration, the squares the gene expression levels.

Imagine we have a (potentially very large) network made up of unknown transcription factor concentrations and observed gene expression levels. We would like to be able to pick a transcription factor, and infer its level at the time points where we observe our gene expression levels. We first examine two simple network models to as a vehicle for larger network models.

3.1 Network Model 1

The Model

If we assume, as in previous models that, for $x_0 = f$, $i = 1, 2$,

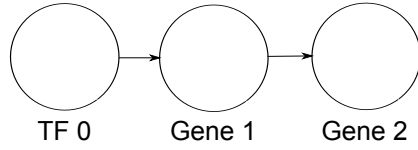


Figure 3.2: We have f (unknown) acting as a transcription factor for x_1 , which in turn acts as a transcription factor for x_2 (observed).

$$x'_i(t) = B_i + S_i x_{i-1}(t) - D_i x_i(t) \quad (3.1)$$

We obtain

$$x_i(t) = \frac{B_i}{D_i} + S_i \exp(-D_i t) \int_0^t x_{i-1}(u) \exp(D_i u) du \quad (3.2)$$

And so, substituting the expression for $x_1(t)$ into the expression for $x_2(t)$:

$$x_2(t) = \frac{B_2}{D_2} + \frac{S_2 B_1}{D_2 D_1} (1 - e^{-D_2 t}) + S_2 e^{-D_2 t} \int_0^t \int_0^u S_1 e^{D_2(u+u') - D_1 u} f(u') du' du \quad (3.3)$$

We see that we can now model x_2 solely as a function of f , just with a different relationship to the one $x_1(t)$ has with f .

Calculating the Mean and Covariance functions

We want now to calculate the mean and covariance functions so that we can try to infer f as before.

$$m(x_2(t)) = \frac{B_2}{D_2} + \frac{S_2 B_1}{D_2 D_1} (1 - e^{-D_2 t}) \quad (3.4)$$

$$\text{Cov}(x_2(t), x_2(t')) = S_2^2 S_1^2 e^{-D_2(t+t')} \int_0^t \int_0^{t'} \int_0^u \int_0^{u'} e^{D_2(u+u'+w+w') - D_1(u+u')} \exp\left(-\frac{(w-w')^2}{\ell}\right) dw' dw du' du \quad (3.5)$$

This integral can be computed analytically along the lines of Lawrence et al., but it has a very large number of terms, approximately one hundred error functions. Instead of actually doing this, we will instead compute this numerically.

It is quite disconcerting that just adding an intermediate transcription factor adds so much complexity (and cost) to the calculation. One would expect that considering a longer chain of transcription factors would compound this even more. We will return to this later.

$$\text{Cov}(x_2(t), f(t')) = S_1 S_2 e^{-D_2 t} \int_0^t \int_0^u S_1 e^{D_2(u+u') - D_1 u} \exp\left(-\frac{(u'-t')^2}{\ell}\right) du' du \quad (3.6)$$

This integral can be computed as before. We can simulate data for this model:

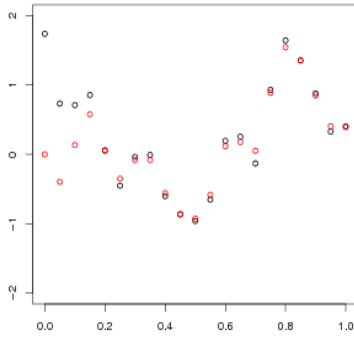
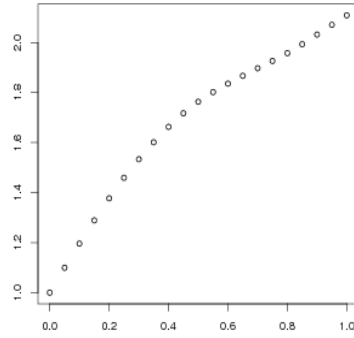
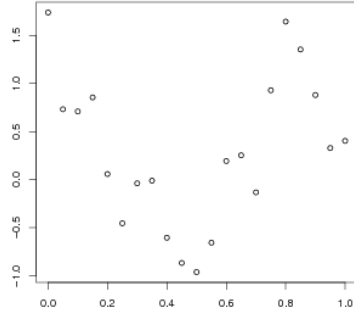


Figure 3.3: *Top*: Simulated f . *Middle*: x_2 . *Bottom*: Simulated f (black) and inferred f (red). Parameters here are $l = 0.1$, $D_1 = 1$, $D_2 = 1.5$, $B_1 = 1$, $B_2 = 1.5$, $S_1 = 1$, $S_2 = 1$.

3.2 Network Model 2

Here we have the model:

$$x_1'(t) = B_1 + S_{11}f_1(t) + S_{12}f_2(t) + R_2x_2(t) - D_1x_1(t) \quad (3.7)$$

$$x_2'(t) = B_2 + S_{21}f_1(t) + S_{22}f_2(t) + R_1x_1(t) - D_2x_2(t) \quad (3.8)$$

Solving these equations

In matrix form this is

$$\mathbf{x}' = \mathbf{B} + \mathbf{Sf} + \mathbf{R}\mathbf{x} \quad (3.9)$$

The solution to this is

$$\mathbf{x} = e^{\mathbf{R}t} \int_0^t e^{-\mathbf{R}u} (\mathbf{B} + \mathbf{Sf}) du + e^{\mathbf{R}t} \frac{\mathbf{B}}{\mathbf{D}} \quad (3.10)$$

To compute this solution we need to find out what $e^{\mathbf{R}t}$ is. We assume \mathbf{R} is diagonalisable. If

$$\mathbf{R} = \begin{pmatrix} -D_1 & R_2 \\ R_1 & -D_2 \end{pmatrix} \quad (3.11)$$

Then

$$\mathbf{R} = \frac{1}{K} \begin{pmatrix} R_2 & R_2 \\ \lambda_+ + D_1 & \lambda_- + D_1 \end{pmatrix} \begin{pmatrix} \lambda_+ & 0 \\ 0 & \lambda_- \end{pmatrix} \begin{pmatrix} \lambda_- + D_1 & -R_2 \\ -(D_1 + \lambda_+) & R_2 \end{pmatrix} \quad (3.12)$$

where $\lambda_+ = \frac{-(D_2+D_1)+\sqrt{(D_1-D_2)^2+4R_1R_2}}{2}$, $\lambda_- = \frac{-(D_2+D_1)-\sqrt{(D_1-D_2)^2+4R_1R_2}}{2}$, $K = R_2(\lambda_- - \lambda_+)$.

Explicitly, equation 3.10 goes to

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \begin{pmatrix} a_1e^{\lambda_+t} + a_3e^{\lambda_-t} + a_5 + d_1f_1*g_1 + d_2f_1*g_2 + d_5f_2*g_1 + d_6f_2*g_2 \\ a_2e^{\lambda_+t} + a_4e^{\lambda_-t} + a_6 + d_3f_1*g_1 + d_4f_1*g_2 + d_7f_2*g_1 + d_8f_2*g_2 \end{pmatrix} \quad (3.13)$$

where * denotes convolution and

$$\begin{aligned} a_1 &= \frac{1}{\lambda_+\lambda_-(\lambda_+-\lambda_-)} \left(\left(\frac{R_2B_2}{D_2} - B_1 \right) \lambda_+\lambda_- - \lambda_-^2 \lambda_+ \frac{B_1}{D_1} - (D_1B_1 - R_2B_2) \lambda_- - \lambda_-^2 B_1 \right) \\ a_2 &= \frac{1}{\lambda_+\lambda_-(\lambda_+-\lambda_-)} \left(\left(\frac{R_1B_1}{D_1} + \frac{D_1B_1}{D_2} \right) \lambda_+\lambda_- + \lambda_+^2 \lambda_- \frac{B_2}{D_2} + (R_1B_1 + B_2D_1) \lambda_- + \lambda_+\lambda_- B_2 \right) \\ a_3 &= \frac{1}{\lambda_+\lambda_-(\lambda_+-\lambda_-)} \left(\lambda_+\lambda_- \left(B_1 - \frac{R_2B_2}{D_2} \right) + \lambda_+^2 \lambda_- \frac{B_1}{D_1} - \lambda_+ (R_2B_2 - D_1B_1) + \lambda_+^2 B_1 \right) \\ a_4 &= \frac{1}{\lambda_+\lambda_-(\lambda_+-\lambda_-)} \left(-\lambda_+\lambda_- \left(\frac{R_1B_1}{D_1} + \frac{D_1B_2}{D_2} \right) - \lambda_+\lambda_-^2 B_2 D_2 - \lambda_+ (R_1B_1 + B_2D_1) - \lambda_+\lambda_- B_2 \right) \\ a_5 &= \frac{1}{\lambda_+\lambda_-} (-B_1D_1 + R_2B_2 - (\lambda_+ + \lambda_-)B_1) \\ a_6 &= \frac{1}{\lambda_+\lambda_-} (R_1B_1 + B_2D_1) \end{aligned}$$

$$\begin{aligned} d_1 &= R_2S_{12} - D_1S_{11} - \lambda_-S_{11} \\ d_2 &= -R_2S_{12} + D_1S_{11} + \lambda_+S_{11} \end{aligned}$$

$$\begin{aligned}
d_3 &= R_1 S_{11} + D_1 S_{12} + \lambda_+ S_{12} \\
d_4 &= -R_1 S_{11} - D_1 S_{11} - \lambda_- S_{12} \\
d_5 &= R_2 S_{22} - D_1 S_{21} - \lambda_- S_{21} \\
d_6 &= -R_2 S_{22} + D_1 S_{21} + \lambda_+ S_{21} \\
d_7 &= R_1 S_{21} + D_1 S_{22} + \lambda_+ S_{22} \\
d_8 &= R_1 S_{21} - D_1 S_{22} - \lambda_- S_{22}
\end{aligned}$$

We then take Laplace Transforms and rearrange as before to get

$$\begin{aligned}
& d_3 \int_0^t e^{\lambda_+(t-u)} x_1(u) du + d_4 \int_0^t e^{\lambda_-(t-u)} x_1(u) du - d_1 \int_0^t e^{\lambda_+(t-u)} x_2(u) du - d_2 \int_0^t e^{\lambda_-(t-u)} x_2(u) du \\
&= b_1 t e^{\lambda_+ t} + b_2 t e^{\lambda_- t} + b_3 e^{\lambda_+ t} + b_4 e^{\lambda_- t} + b_5 + b_6 \alpha(\lambda_+, \lambda_+) + b_7 \alpha(\lambda_-, \lambda_-) + b_8 \alpha(\lambda_+, \lambda_-)
\end{aligned}$$

where $\alpha(p, k) = e^{pt} \int_0^t e^{(k-p)s} \int_0^s e^{-ku} f_2(u) du ds$ and

$$\begin{aligned}
b_1 &= a_1 d_3 - d_1 a_2 \\
b_2 &= a_3 d_4 - d_2 a_4 \\
b_3 &= \frac{1}{\lambda_+ - \lambda_-} (a_1 d_4 + a_3 d_3 - d_1 a_4 - d_2 a_2) + \frac{d_3 a_5 - d_1 a_6}{\lambda_+} \\
b_4 &= \frac{-1}{\lambda_+ - \lambda_-} (a_1 d_4 + a_3 d_3 - d_1 a_4 - d_2 a_2) + \frac{d_4 a_5 - d_2 a_6}{\lambda_-} \\
b_5 &= \frac{d_1 a_6 - d_3 a_5}{\lambda_+} + \frac{d_2 a_6 - d_4 a_5}{\lambda_-} \\
b_6 &= d_3 d_5 - d_1 d_7 \\
b_7 &= d_4 d_6 - d_2 d_8 \\
b_8 &= d_4 d_5 + d_3 d_6 - d_1 d_8 - d_2 d_7
\end{aligned}$$

The Mean and Covariance functions

We now wish to find the mean and covariance functions, but now of

$$z_2(t) := d_3 \int_0^t e^{\lambda_+(t-u)} x_1(u) du + d_4 \int_0^t e^{\lambda_-(t-u)} x_1(u) du - d_1 \int_0^t e^{\lambda_+(t-u)} x_2(u) du - d_2 \int_0^t e^{\lambda_-(t-u)} x_2(u) du \quad (3.14)$$

$$\mathbf{m}(z_2(t)) = b_1 t e^{\lambda_+ t} + b_2 t e^{\lambda_- t} + b_3 e^{\lambda_+ t} + b_4 e^{\lambda_- t} + b_5 \quad (3.15)$$

Defining

$$\beta(k, p, m, q)[t, t'] = e^{pt+qt'} \int_0^t \int_0^{t'} \int_0^s \int_0^{s'} e^{(k-p)s+(l-q)s'} e^{-ku-mu'} \exp\left(-\left(\frac{u-u'}{l}\right)^2\right) du' du ds' ds \quad (3.16)$$

We get

$$\begin{aligned}
\mathbf{Cov}[z_2(t), z_2(t')] &= b_6^2 \beta(\lambda_+, \lambda_+, \lambda_+, \lambda_+)[t, t'] + b_7^2 \beta(\lambda_-, \lambda_-, \lambda_-, \lambda_-)[t, t'] + b_8^2 \beta(\lambda_+, \lambda_-, \lambda_+, \lambda_-)[t, t'] + \\
& b_6 b_7 (\beta(\lambda_+, \lambda_+, \lambda_-, \lambda_-)[t, t'] + \beta(\lambda_+, \lambda_+, \lambda_-, \lambda_-)[t', t]) + b_6 b_8 (\beta(\lambda_+, \lambda_+, \lambda_+, \lambda_-)[t, t'] + \beta(\lambda_+, \lambda_+, \lambda_+, \lambda_-)[t', t]) \\
& + b_7 b_8 (\beta(\lambda_+, \lambda_-, \lambda_-, \lambda_-)[t, t'] + \beta(\lambda_+, \lambda_-, \lambda_-, \lambda_-)[t', t])
\end{aligned}$$

and defining

$$\gamma(p, k)[t, t'] = e^{pt} \int_0^t e^{(k-p)s} \int_0^s e^{-ku} \exp\left(-\left(\frac{u-t'}{l}\right)^2\right) du ds \quad (3.17)$$

we get

$$\mathbf{Cov}[z_2(t), f_i(t')] = \delta_{i2} (b_6\gamma(\lambda_+, \lambda_+)[t, t'] + b_7\gamma(\lambda_-, \lambda_-)[t, t'] + b_8\gamma(\lambda_+, \lambda_-)[t, t'])$$

Again, we compute these integrals numerically.

We once again simulate data with parameters $l = 0.1$, $D_1 = 2$, $D_2 = 3$, $S_{11} = 6$, $S_{12} = 2$, $S_{21} = 4$, $S_{22} = 3.5$, $B_1 = 1$, $B_2 = 1.5$, $R_1 = 0.3$, $R_2 = 0.4$:

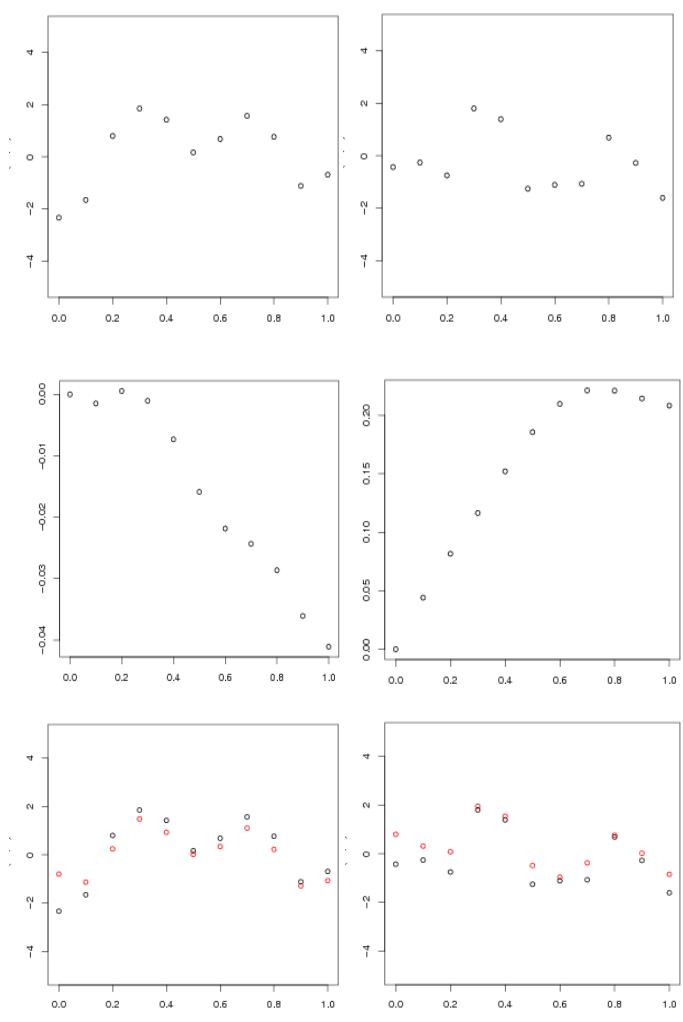


Figure 3.4: *Top*: Simulated f_1 (left) and f_2 (right). *Middle*: z_2 (left) and z_2 (right). *Bottom*: Simulated f_1 (black) and inferred f_1 (red) (left) and simulated f_2 (black) and inferred f_2 (red) (right).

3.3 Generalising Case 1: The Network is *closed*

What we mean by *closed* is that every node has an arrow going into it. If we assume that the relationship between two nodes is as before, we have

$$\mathbf{x}'(t) = \mathbf{B} + \mathbf{S}\mathbf{x} \quad (3.18)$$

where $\mathbf{x} = (x_1, x_2, \dots, x_k, f_1, f_2, \dots, f_m)^T$ for x_i gene expressions and f_i transcription factor concentrations.

We can solve this explicitly to give a deterministic equation for \mathbf{x} , with \mathbf{c} a constant vector:

$$\mathbf{x} = e^{\mathbf{S}t} \int_0^t e^{-\mathbf{S}u} \mathbf{B} du + e^{\mathbf{S}t} \mathbf{c} \quad (3.19)$$

This is a completely deterministic equation for \mathbf{x} , if we can figure out \mathbf{c} .

Computing c

Computing c requires having $k+m$ observations. If we have initial conditions $\bar{\mathbf{x}}(0) = (x_1(0), x_2(0), \dots, x_k(0))^T = \frac{\mathbf{B}}{\mathbf{D}}$ as before, we will have the first k these, seen by evaluating the above equation at 0. We then need an additional m observations.

An obvious way to get these would be by taking observations. However, we assume that we observe with noise, observing $y_i = x_i + \epsilon_i$. So assume we have made m additional observations y_1, \dots, y_m at t_1, \dots, t_m , where y_i is an observation of x_{k_i} . We then obtain m equations of the form

$$y_i - \epsilon_i = \left(e^{\mathbf{S}t_i} \int_0^{t_i} e^{-\mathbf{S}u} \mathbf{B} du \right)_{k_i} + (e^{\mathbf{S}t_i} \mathbf{c})_{k_i} \quad (3.20)$$

Putting these together we have, for \mathbf{K} a constant vector and \mathbf{M} a constant matrix, something that looks like

$$\mathbf{y} - \boldsymbol{\epsilon} = \mathbf{K} + \mathbf{M}\mathbf{c} \quad (3.21)$$

which gives

$$\mathbf{c} = \mathbf{M}^{-1}(\mathbf{y} - \mathbf{K} - \boldsymbol{\epsilon}) \quad (3.22)$$

Hence

$$\mathbf{x} = e^{\mathbf{S}t} \int_0^t e^{-\mathbf{S}u} \mathbf{B} du + e^{\mathbf{S}t} \mathbf{M}^{-1}(\mathbf{y} - \mathbf{K}) - e^{\mathbf{S}t} \mathbf{M}^{-1} \boldsymbol{\epsilon} \quad (3.23)$$

With this we then have a solution for each f_i with an adjusted noise function $e^{\mathbf{S}t} \mathbf{M}^{-1} \boldsymbol{\epsilon}$.

3.4 Generalising Case 2: More dependent genes than independent transcription factors

By an independent transcription factor, we mean a transcription factor which is not affected by any other node. By a dependent gene we mean one which is affected by an independent transcription factor. Assume we have x_1, \dots, x_k as dependent genes. The method below outlines how to infer $(f_1 \dots f_n)$.

Here we have the equations for $\bar{\mathbf{x}} = (x_1, \dots, x_k, f_1, \dots, f_l)^T$ and $\bar{\mathbf{f}} = (x_{k+1}, \dots, x_m, f_{l+1}, \dots, f_n)^T$. Note we have $k \geq n - l$ by model assumption.

$$\bar{\mathbf{x}}' = \mathbf{B} + \mathbf{S}\bar{\mathbf{f}} + \mathbf{S}^{\mathbf{x}}\bar{\mathbf{x}} \quad (3.24)$$

We then take Laplace transforms:

$$pL[\bar{\mathbf{x}}] = \frac{\mathbf{B}}{p} + \mathbf{S}L[\bar{\mathbf{f}}] + \mathbf{S}^{\mathbf{x}}L[\bar{\mathbf{x}}] \quad (3.25)$$

Moving all of the \mathbf{x} to the left hand side and all of the \mathbf{f} to the right hand side we obtain for a $(k+l) \times m$ matrix \mathbf{K} and $(k+l) \times n$ matrix \mathbf{M}

$$\mathbf{K}L[\bar{\mathbf{x}}] = \frac{\mathbf{B}}{p} + \mathbf{M}L[\bar{\mathbf{f}}] \quad (3.26)$$

Untangling the effect of the transcription factors

Unless we have the same number of dependent genes as transcription factors we will not have that \mathbf{M} is invertible. So we perform row and column operations via a $(k+l) \times (k+l)$ matrix \mathbf{E} until we have

$$\mathbf{E}\mathbf{K}L[\bar{\mathbf{x}}] = \mathbf{E}\frac{\mathbf{B}}{p} + \mathbf{D}L[\bar{\mathbf{f}}] \quad (3.27)$$

where $\bar{\mathbf{x}} = (x_1, \dots, x_m)^T$, $\bar{\mathbf{f}} = (f_1, \dots, f_n)^T$, and \mathbf{D} is a $(k+l) \times n$ matrix of the form:

$$\begin{pmatrix} \mathbf{I}_n \\ \mathbf{0} \end{pmatrix}$$

Inverting the transform

It is in general difficult to give detailed instructions on how to invert the Laplace transform since the matrix \mathbf{E} may have p s in it. In the simple case when $l = 0$, we proceed as in Section 2.3.1, as then \mathbf{E} will be a constant matrix. When $l \neq 0$, this is not the case. Using:

$$L^{-1}\left(\frac{L[x_i]}{p-D_j}\right) = e^{-D_j t} \int_0^t e^{D_j u} x_i(u) du$$

$$L^{-1}\left(\frac{L[x_i]}{(p-D_j)^2}\right) = e^{-D_j t'} \int_0^{t'} \int_0^t e^{D_j u} x_i(u) du dt$$

it is possible to invert the transform. The mean and covariance functions can then be calculated for adjusted observations as in Section 2.3.1, and we can infer our transcription factor concentration.

3.5 Speeding up the computation

3.5.1 The Problem

We need to compute the covariance matrices, and other numerical integrals. These usually involve computing many integrals of the form

$$I[t, t'] = \int_0^t \int_0^{t'} f(u, u') du' du \quad (3.28)$$

Typically f is not a function which is quick to numerically integrate. If the two nodes which we would like to find the covariance of are far away, f will contain integrals itself. For example, in Network Model 1, f is of the form

$$f(u, u') = \int_0^u \int_0^{u'} e^{D_2(u+u'+w+w')-D_1(u+u')} \exp\left(-\frac{(w-w')^2}{\ell}\right) dw' dw \quad (3.29)$$

3.5.2 Ways to increase efficiency

We use the fact that we are looking to evaluate I at a set of discrete points $t_1 < t_2 < \dots < t_n$.

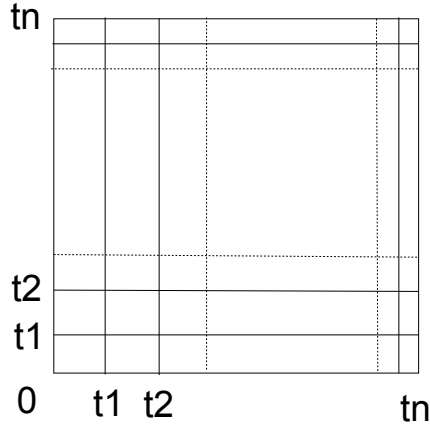


Figure 3.5: We consider a grid over which we integrate f .

We then define for $i, j \geq 1$, setting $t_0 = 0$

$$I_{ij} = \int_{t_{i-1}}^{t_i} \int_{t_{j-1}}^{t_j} f(u, u') du' du \quad (3.30)$$

Then we have

$$I[t_p, t_q] = \sum_{i=0}^p \sum_{j=0}^q I_{ij} \quad (3.31)$$

Previously, we would have computed $O(n^4)$ integrals, but now we only have to compute $O(n^2)$ integrals and sum them as appropriate. In Network model 1, we computed the covariance at 10 data points. $10^4 = 10000$, $10^2 = 100$. Already we can see a large difference in computation.

We can also notice that $\text{Cov}[x(t), x(t)]$ is a symmetric matrix since $\text{Cov}[x(t), x(t')] = \text{Cov}[x(t'), x(t)]$. This will in addition reduce the number of summations we have to do.

Chapter 4

Non-linear model

4.1 Introduction

Suppose we have observed gene expression x at time point t_1, t_2, \dots, t_T , we would like to find the posterior of the underlying transcription factor levels at those time points.

The posterior of the underlying transcription factor levels $f = f(t_1), \dots, f(t_T)$ is:

$$p(f|X, y) \propto p(y|f)p(f|X) = \exp(\Psi(f)) \quad (4.1)$$

4.1.1 Aim

Our aim is to solve for \hat{f} that maximizes the posterior distribution (equation 4.1) and use Laplace approximation around the maximum \hat{f} . The Laplace approximation to the distribution of f is

$$f \sim N(\hat{f}, (-\nabla\nabla\Psi(f))). \quad (4.2)$$

Newton method

To solve for the maximum of equation 4.1, the Newton method is utilized.

$$f^{\text{new}} = f - (\nabla\nabla\Psi)^{-1}\nabla\Psi$$

$$\nabla\Psi(f) = \nabla\log p(y|f) - K^{-1}f \quad (4.3)$$

$$\nabla\nabla\Psi(f) = \nabla\nabla\log p(y|f) - K^{-1} = -W - K^{-1} \text{ where } W = -\nabla\nabla\log p(y|f) \quad (4.4)$$

We draw f^{new} until convergence. Our implementation follows algorithm 3.1 of Rasmussen and Williams [7].

4.2 Derivation of $\nabla \log p(y|f)$ and $\nabla \nabla \log p(y|f)$

4.2.1 log likelihood

The log likelihood of $Y|f$ is the independent noise term, where $x_j(t_i)$ is the expression of gene j at time t_i :

$$\log p(Y|\cdot) = -\frac{1}{2} \sum_{i=1}^T \sum_{j=1}^N \left[\frac{(x_j(t_i) - y_j(t_i))^2}{\sigma_{j_i}^2} + \log(\sigma_{j_i}^2) \right] - \frac{NT}{2} \log(2\pi) \quad (4.5)$$

Hence, the first and second derivatives are:

$$\frac{\partial \log p(Y|\cdot)}{\partial f_k(t_p)} = - \sum_{i=1}^T \sum_{j=1}^N \frac{x_j(t_i) - y_j(t_i)}{\sigma_{j_i}^2} \frac{\partial x_j(t_i)}{\partial f_k(t_p)} \quad (4.6)$$

$$\frac{\partial^2 \log p(Y|\cdot)}{\partial f_k(t_p) \partial f_h(t_q)} = -\delta_{hk} \delta_{pq} \sum_{i=1}^T \sum_{j=1}^N \frac{x_j(t_i) - y_j(t_i)}{\sigma_{j_i}^2} \frac{\partial^2 x_j(t_i)}{\partial^2 f_k(t_p)} - \sum_{i=1}^T \sum_{j=1}^N \frac{\partial x_j(t_i)}{\partial f_k(t_p)} \frac{\partial x_j(t_i)}{\partial f_h(t_q)} \frac{1}{\sigma_{j_i}^2} \quad (4.7)$$

where δ is the Dirac delta function.

4.2.2 The partial derivatives of x

Recall the following equation which determines how gene expression depends on the transcription factor level:

$$\frac{dx_j}{dt} = B_j + \sum_k g_{k,j}(f_k(t), \theta_j) - D_j x_j \quad (4.8)$$

The gene expression x given transcription factor level f and model parameters (B, S, D) is:

$$x_j(t) = \frac{B_j}{D_j} + k_j \exp(-D_j t) + \exp(-D_j t) \sum_k \int_0^t g_{k,j}(f_k(u), \theta_j) \exp(D_j u) du \quad (4.9)$$

where k_j depends on the initial condition $x_j(0)$. Suppose we have a uniform grid on time t_1, t_2, \dots, t_n where $t_i - t_{i-1} = \Delta$. Above equation can be approximated by

$$x_j(t_i) \approx \frac{B_j}{D_j} + k_j \exp(-D_j t) + \exp(-D_j t_i) \sum_k \Delta \sum_{\alpha=1}^i g_{k,j}(f_k(t_\alpha)) \exp(D_j t_\alpha) \quad (4.10)$$

Hence, we can calculate the derivatives of x

$$\frac{\partial x_j(t_i)}{\partial f_k(t_p)} = \exp(-D_j(t_i - t_p)) \Delta g'_{k,j}(f_k(t_p)) \Theta(t_i - t_p) \quad (4.11)$$

$$\frac{\partial^2 x_j(t_i)}{\partial^2 f_k(t_p)} = \exp(-D_j(t_i - t_p)) \Delta g''_{k,j}(f_k(t_p)) \Theta(t_i - t_p) \quad (4.12)$$

where Θ is the Heaviside step function ($\Theta(x) = I(x \geq 0)$). The Heaviside step function is a consequence of the fact that $x_j(t)$ is influenced by $f_j(u)$ with $u \leq t$.

Other second derivatives equal to zero:

$$\frac{\partial^2 x_j(t_i)}{\partial f_k(t_p) \partial f_k(t_q)} = 0$$

$$\frac{\partial^2 x_j(t_i)}{\partial f_k(t_p) \partial f_h(t_q)} = 0$$

$$\frac{\partial^2 x_j(t_i)}{\partial f_k(t_p) \partial f_h(t_p)} = 0$$

4.2.3 Compute functions involving f

Equations 4.6 and 4.7 depend on x which further depends on f .

$$x = \frac{B}{D} + e^{-Dt} \int_0^t e^{Du} g(f(u)) du$$

The integral $\int_0^t e^{Du} g(f(u)) du$ is computed using composite Simpson's rule on the same uniform time grid.

Equations 4.11 and 4.12 also depend on f . However, f is a random quantity. Our solution to this problem is to replace above equations with the expectation of it. If $g(x) = x$, i.e. the linear case, then it is simple: $\mathbf{E}g(f) = \bar{f}$, $\mathbf{E}g'(f) = 1$ and $\mathbf{E}g''(f) = 0$.

When $g(x) = \exp(x)$,

$$\mathbf{E} \exp(X) = \exp\left(\mu + \frac{\sigma^2}{2}\right) \text{ where } X \sim N(\mu, \sigma^2)$$

As an alternative, we could make a simple approximation:

$$\mathbf{E}g(X) \approx g(\mathbf{E}X)$$

If g is linear on x , then the above approximation is exact. When the inferred TFs has a narrow confidence band, then the above approximation could be reasonably good. For example,

$$\mathbf{E} \exp(X) = \exp\left(\mathbf{E}X + \frac{\text{Var}X}{2}\right) \approx \exp(\mathbf{E}X)$$

4.3 Generalize this method

We have only discussed the case when the gene expression depends on a linear combination of non-linear functions of transcription factors. Recall equation 4.6 and 4.7, this method will work as long as we can simulate gene expression x from transcription factors f and compute the first two derivatives of x with respect to f .

This gives us freedom to choose a more complicated function g , e.g. function g can be chosen to depend on two or more transcription factors. It is also possible to consider a complicated network model with gene interacting with each other.

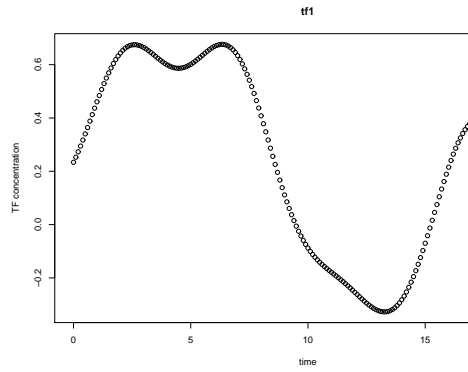


Figure 4.1: Transcription Factor 1

4.4 Simulation

We begin with a model with three genes and two transcription factors. We assume linear activation, $g(x) = x$ and the following parameters.

	B	S1	S2	D
Gene 1	1	1	1	1
Gene 2	1	2	1	1
Gene 3	1	1	2	1

The gene expressions computed using above parameters are shown below.

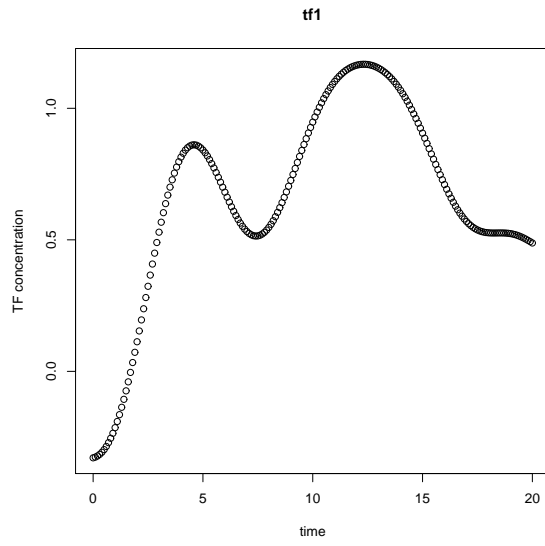


Figure 4.2: Transcription Factor 2

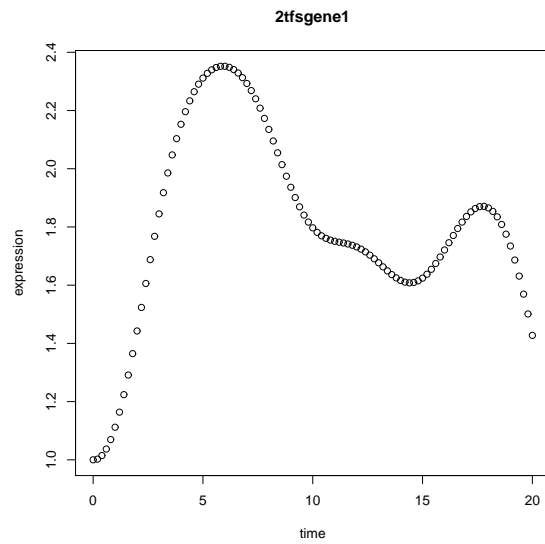


Figure 4.3: Gene expression of Gene 1

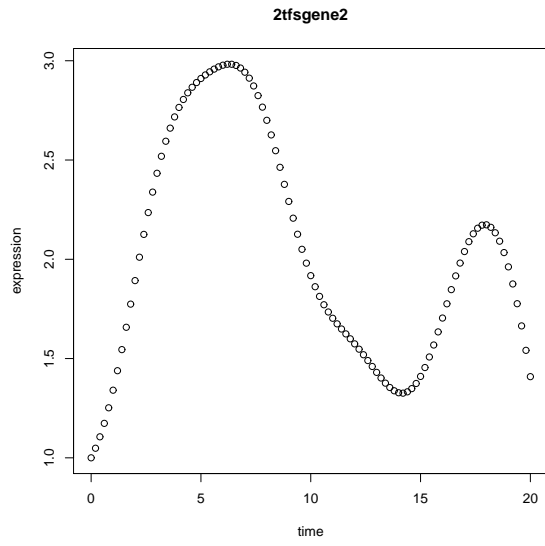


Figure 4.4: Gene expression of Gene 2

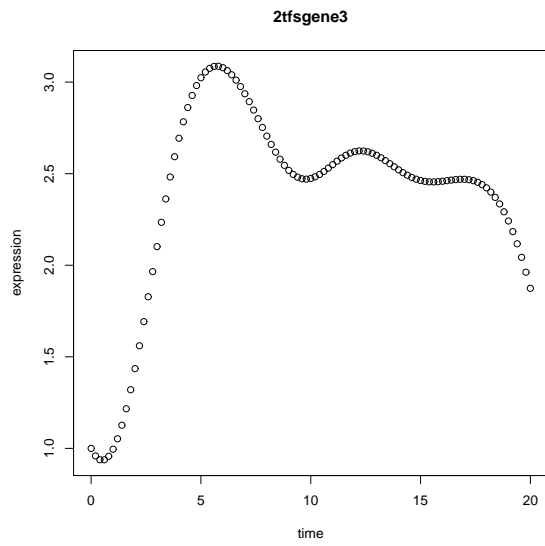


Figure 4.5: Gene expression of Gene 3

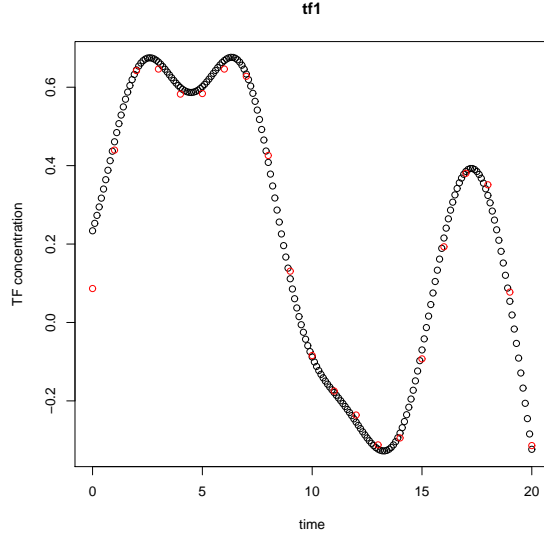


Figure 4.6: Inferred Transcription Factor 1: red dots are the inferred TF level and black dots are the original TF level

4.4.1 Result

Assume that we know the parameters, we then infer the underlying transcription factor level. Results are shown in Figure 4.6 and 4.7.

4.5 Non-linear case

We first consider g of the form $g(x) = \frac{1}{1+\exp(x)}$. The result is shown in Figure 4.5.

4.5.1 Avoid computing K^{-1}

Recall at the beginning of this chapter that we have to compute quantities $(W + K^{-1})^{-1}$ which could be numerically unstable. Algorithm 3.1 given in Rasmussen and Williams [7, P.46] assume that W is positive definite and hence $W^{\frac{1}{2}}$ can be computed. However, it appears that W in our case is not positive definite and we have to compute the quantities $(W + K^{-1})^{-1}$ directly. This limits the choice of the bandwidth ℓ to relatively small values.

Here, we outline a way to avoid computing K . However, it does not seem to be particularly numerically stable, the algorithm is included to outline the general idea involved. There are a few quantities involving K^{-1} the Newton iteration algorithm. They are $(K^{-1} + W)^{-1}$ and $K^{-1}f$:

$$f^{new} = f^{old} + (W + K^{-1})^{-1}(\nabla \log p(y|f) - K^{-1}f^{old}) \quad (4.13)$$

This could be rewritten in the form:

$$f^{new} = (W + K^{-1})^{-1}(Wf + \nabla \log p(y|f)) \quad (4.14)$$

To compute $(W + K^{-1})^{-1}$, we will use the Woodbury formula:

$$(Z + UDV^T)^{-1} = Z^{-1} - Z^{-1}U(D^{-1} + V^T Z^{-1}U)^{-1}V^T Z^{-1}$$

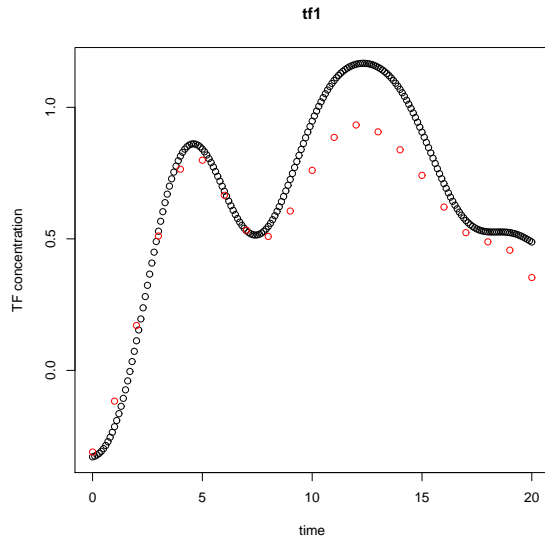


Figure 4.7: Inferred Transcription Factor 2: red dots are the inferred TF level and black dots are the original TF level

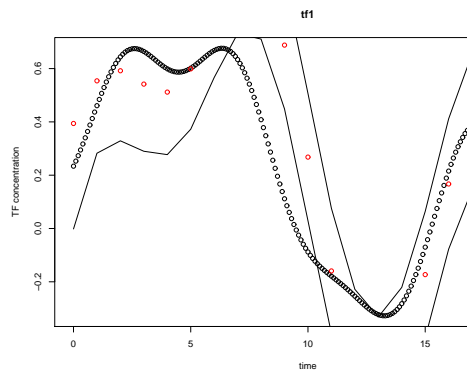


Figure 4.8: Inferred TF level

If we assume that W is diagonalizable, $W = UDV^T$, then,

$$(W + K^{-1})^{-1} = K - KU(D^{-1} + V^T KU)^{-1}V^T K$$

Hence,

$$f^{new} = K(I - U(D^{-1} + V^T KU)^{-1}V^T K)(Wf + \nabla \log p(y|f)) \quad (4.15)$$

Furthermore,

$$K^{-1}f = (I - V(D^{-1} + V^T KV)^{-1}V^T K)(Wf + \nabla \log p(y|f))$$

4.6 Implementation details - matrices

Suppose we have 2 transcription factors f_1 and f_2 . We would like to estimate TFs levels at time points on uniform grid: (t_1, t_2, \dots, t_N) .

$$f = \begin{pmatrix} f_1(t_1) \\ f_1(t_2) \\ \vdots \\ f_1(t_N) \\ f_2(t_1) \\ f_2(t_2) \\ \vdots \\ f_2(t_N) \end{pmatrix} \quad (4.16)$$

The covariance matrix K describing the the covariance between f is

$$K = \begin{pmatrix} k(t_1, t_2) & k(t_1, t_2) & \dots & k(t_1, t_N) & 0 & \dots & 0 \\ k(t_2, t_1) & k(t_2, t_2) & \dots & k(t_2, t_N) & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ k(t_N, t_1) & k(t_N, t_2) & \dots & k(t_N, t_N) & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & k(t_1, t_1) & \dots & k(t_1, t_N) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & k(t_N, t_1) & \dots & k(t_N, t_N) \end{pmatrix} \quad (4.17)$$

where the covariance function is chosen to be $k(t, t') = \exp(-\frac{(t-t')^2}{2\ell^2})$. We have assumed that the covariance across different f is 0. ¹

Then, $W = \nabla \nabla \log p(y|f)$ is

$$W = \begin{pmatrix} \frac{\partial^2 \log p(y|f)}{\partial^2 f_1(t_1)} & \frac{\partial^2 \log p(y|f)}{\partial f_1(t_1) \partial f_1(t_2)} & \dots & \frac{\partial^2 \log p(y|f)}{\partial f_1(t_1) \partial f_1(t_N)} & \frac{\partial^2 \log p(y|f)}{\partial f_1(t_1) \partial f_2(t_1)} & \dots & \frac{\partial^2 \log p(y|f)}{\partial f_1(t_1) \partial f_2(t_N)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 \log p(y|f)}{\partial f_1(t_N) \partial f_1(t_1)} & \frac{\partial^2 \log p(y|f)}{\partial f_1(t_N) \partial f_1(t_2)} & \dots & \frac{\partial^2 \log p(y|f)}{\partial f_1(t_N) \partial f_1(t_N)} & \frac{\partial^2 \log p(y|f)}{\partial f_1(t_N) \partial f_2(t_1)} & \dots & \frac{\partial^2 \log p(y|f)}{\partial f_1(t_N) \partial f_2(t_N)} \\ \frac{\partial^2 \log p(y|f)}{\partial f_2(t_1) \partial f_1(t_1)} & \frac{\partial^2 \log p(y|f)}{\partial f_2(t_1) \partial f_1(t_2)} & \dots & \frac{\partial^2 \log p(y|f)}{\partial f_2(t_1) \partial f_1(t_N)} & \frac{\partial^2 \log p(y|f)}{\partial^2 f_2(t_1)} & \dots & \frac{\partial^2 \log p(y|f)}{\partial f_2(t_1) \partial f_2(t_N)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 \log p(y|f)}{\partial f_2(t_N) \partial f_1(t_1)} & \frac{\partial^2 \log p(y|f)}{\partial f_2(t_N) \partial f_1(t_2)} & \dots & \frac{\partial^2 \log p(y|f)}{\partial f_2(t_N) \partial f_1(t_N)} & \frac{\partial^2 \log p(y|f)}{\partial f_2(t_N) \partial f_2(t_1)} & \dots & \frac{\partial^2 \log p(y|f)}{\partial^2 f_2(t_N)} \end{pmatrix} \quad (4.18)$$

¹It would be interesting to assume non-zero covariance. There will be a few key questions to ask: can we define a valid covariance function? What is the physical interpretation?

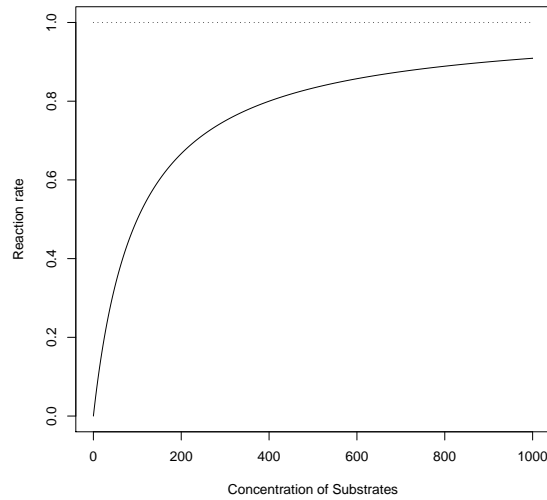


Figure 4.9: Reaction rate under Michaelis-Menten Kinetics

4.7 Function form of g

4.7.1 Simple Cases

Linear activation $g(x) = x$ The problem with this model is that it allows the transcription factor concentration falls below zero. The advantage of this model is that the solution could be found analytically.

Exponential $g(x) = \exp(x)$ The next natural step is to consider a function that maps $[-\infty, \infty]$ to a subset of $[0, \infty]$. We assume now that the transcription factor level is $\exp(x)$ which is always a positive quantity.

4.7.2 Michaelis-Menten kinetics

Michaelis-Menten kinetics is a model that describe the reaction of a catalytic reaction:

$$\text{Reaction rate} = V \frac{[S]}{\gamma + [S]} \text{ where } [S] \text{ is the substrate concentration and } V \text{ is a constant}$$

Where γ is the concentration required to achieve the reaction rate of $0.5V$. This can measure the saturation effects of the reaction(Figure 4.9).

Activation:

$$g(f) = \frac{\exp(f)}{\gamma + \exp(f)}$$

Inhibition:

$$g(f) = \frac{1}{\gamma + \exp(f)}$$

4.7.3 Multiple transcription factors

Any of the transcription factors can activate the transcription. Then, it is equivalent to having one transcription factor which the concentration is the (weighted²) sum of all relevant transcription factors.

$$g(f) = \frac{w_1 \exp(f_1) + w_2 \exp(f_2) + \dots}{\gamma + w_1 \exp(f_1) + w_2 \exp(f_2)}$$

All relevant transcription factors are required to activate the transcription. We can model it via the product of two transcription factors levels:

$$g(f) = \frac{\exp(f_1) \exp(f_2)}{\gamma + \exp(f_1) \exp(f_2)}$$

One transcription factors activate the transcription while the other inhibit the transcription. We can consider the Michaelis-Menten competitive inhibition formula:

$$g(f) = \frac{\exp(f_1)}{\exp(f_1) + \gamma(1 + \frac{\exp(f_2)}{\delta})}$$

²since different transcription factors have different affinity/sensitivities.

Chapter 5

Estimation of model parameters

In this section, we will look at how to obtain maximum likelihood estimate of model parameters (B, S, D) and bandwidth l .

5.1 Marginal Likelihood

The marginal likelihood is:

$$p(y|X) = \int p(y|f)p(f|X)df = \int \exp(\Psi(f))df$$

By using Taylor expansion of $\exp(\Psi(f))$ around \hat{f} [7, P.47],

$$\log q(y|X, \theta) = -\frac{1}{2}\hat{f}^T K^{-1}\hat{f} + \log p(y|\hat{f}) - \frac{1}{2}\log |B|$$

Given $B, S, D, l = 1$ and $\sigma = 0.1$, the marginal likelihood changes as the parameters vary and hence provide a way of parameter estimation via maximizing the likelihood or via a Bayesian MCMC procedure.

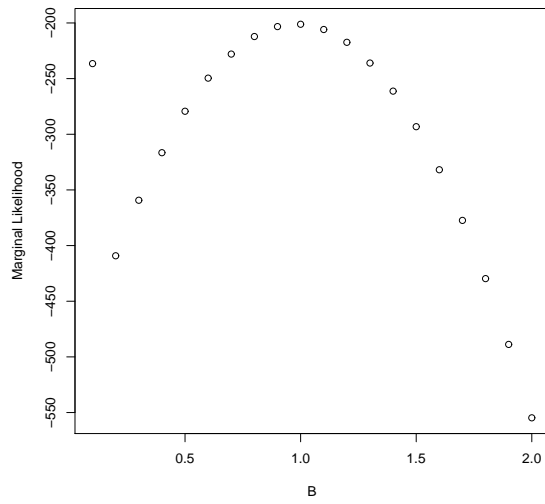


Figure 5.1: Marginal Likelihood as a function of B

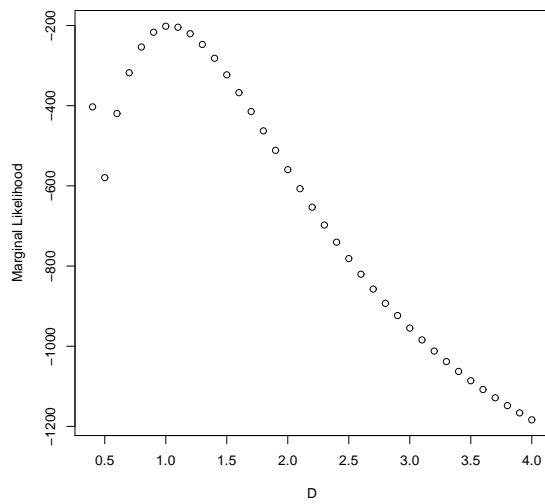


Figure 5.2: Marginal Likelihood as a function of D

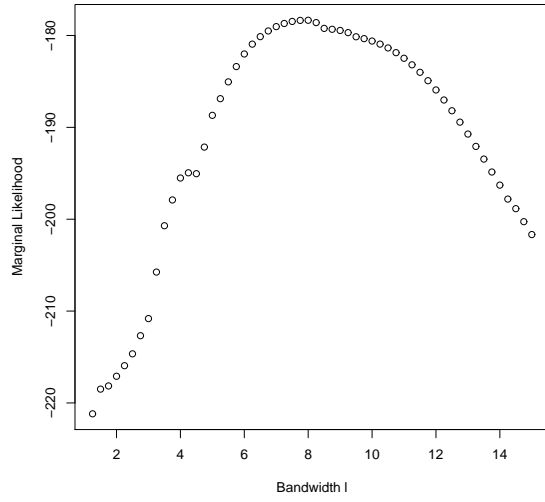


Figure 5.3: Marginal Likelihood as a function of l

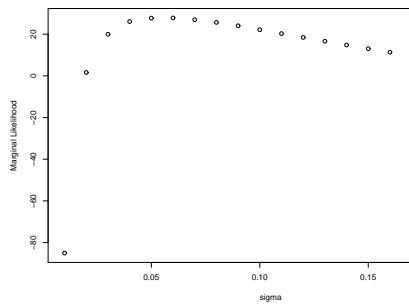


Figure 5.4: Marginal Likelihood as a function of σ

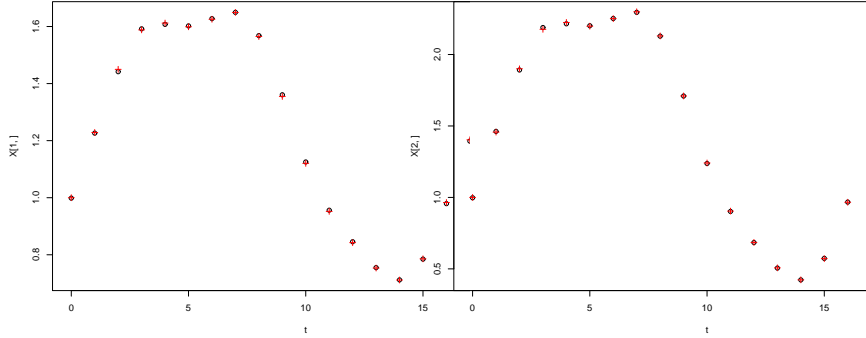


Figure 5.5: Fitted gene expression(black) and observed gene expression(red) using a “wrong” set of parameters

5.1.1 Different sets of parameters explaining the data equally well

Given a set of two gene expressions, it appears that there might be more than one set of parameters that explains the data equally well(Figure 5.5 and 5.6).

However, the inferred transcription factor levels are different(Figure 5.7 and 5.8).

The relevant parameters and the marginal likelihood are:

	B	S	D
Wrong Gene 1	0.7903106	0.979336	0.79148
Correct Gene 1	1	1	1
Wrong Gene 2	0.8102043	2	0.81216
Correct Gene 2	1	2	1
Log marginal likelihood Correct	36.40925		
Log marginal likelihood Wrong	36.54623		

Discussion

From the table above, we could see that the ratio between B and D is roughly correct. We suspected that the marginal likelihood has a ridge along the line $B/D = \text{constant}$. Hence, we will need more information about the parameters in order to obtain accurate inference. This might explain why Gao et al. [6] fixed two parameters S, D in the analysis for the linear case.

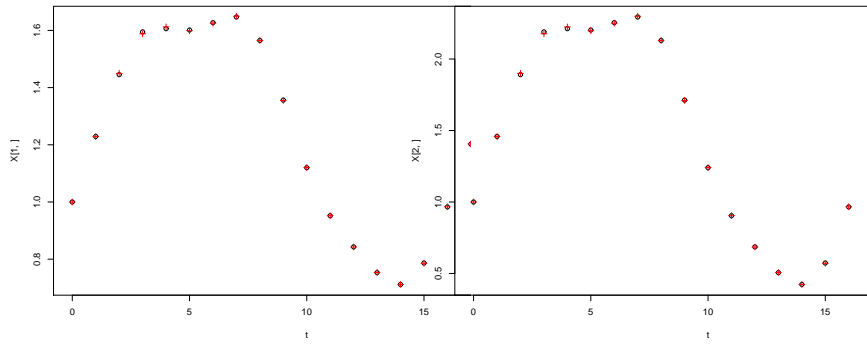


Figure 5.6: Fitted gene expression(black) and observed gene expression(red) using a “correct” set of parameters

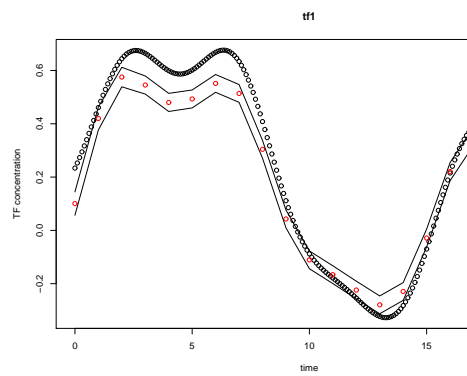


Figure 5.7: Inferred TF level(red) using “wrong” set of parameters and original TF level(black)

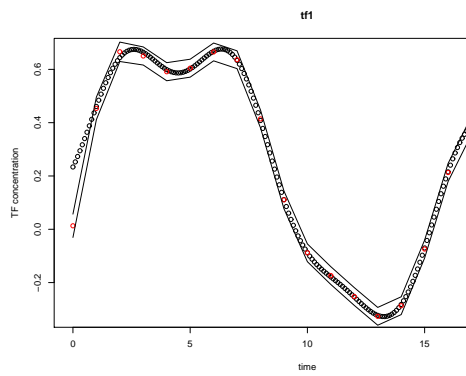


Figure 5.8: Inferred TF level(red) using “correct” set of parameters and original TF level(black)

5.2 MCMC analysis

It is possible to carry out MCMC analysis. However, it is expensive to compute the marginal likelihood using Newton's method (about 10 seconds for a relatively simple situation) and hence the corresponding Metropolis acceptance ratio is also expensive to calculate.

5.2.1 prior

$$\pi(B) \sim \text{Gamma}(\alpha_B, \beta_B)$$

$$\pi(S) \sim \text{Gamma}(\alpha_S, \beta_S)$$

$$\pi(D) \sim \text{Gamma}(\alpha_D, \beta_D)$$

5.2.2 Proposal density

We update the parameters individually by Metropolis algorithm. The proposal of the Metropolis step is "proportional shrinking and expanding" described on page 170 of Yang [11].

Let r be a uniform(0,1) random variables.

$$c = e^{\epsilon(r - \frac{1}{2})}$$

The proposed parameter is

$$\text{new parameter} = c \times \text{old parameter}$$

The proposal ratio is simply c (See the derivation in Yang [11]).

Initial result

Given a random starting point, the Markov chain is allowed to run for 1000 iterations. The chain stayed in a relatively low density region. We suspected that this might be due to high correlation between different parameters and we rerun the analysis by updating all parameters at the same time. The chain can reach higher density region, however, it is still far from the optimal solution we expected to obtain.

Suggestion

In our previous analysis, it appears that B and D are highly correlated. One suggestion is to reparametrise as D and $\frac{B}{D}$.

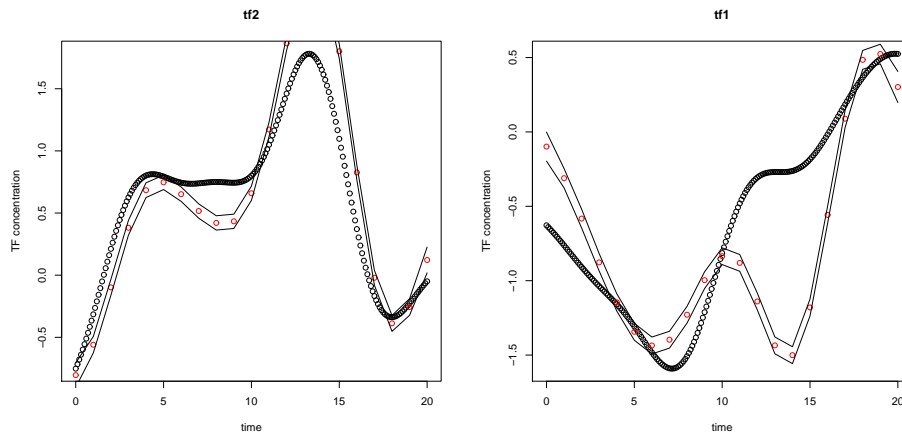


Figure 5.9: Inferred TF level

5.3 Multiple transcription factors

Suppose we have 3 genes expression and they depend on 2 TFs. Using Nelder-Mead optimization method to maximize the marginal likelihood, we obtain the following result.

	B	S1	S2	D
Gene 1	1	1	1	1
Gene 2	1	2	1	1
Gene 3	1	1	2	1

Result after running for 5 hours:

	B	S1	S2	D
Gene 1	1.208	1.535	1.606	1.218
Gene 2	0.799	1.689	2.242	0.795
Gene 3	1.235	2.138	1.467	1.226

This set of parameters explain the data reasonably well(Figure 5.10). However, the inferred transcription factors levels do not coincide with the the original level. This highlights the problem that there might not be enough information to infer both TFs levels and parameters simply by looking at the gene expression data.

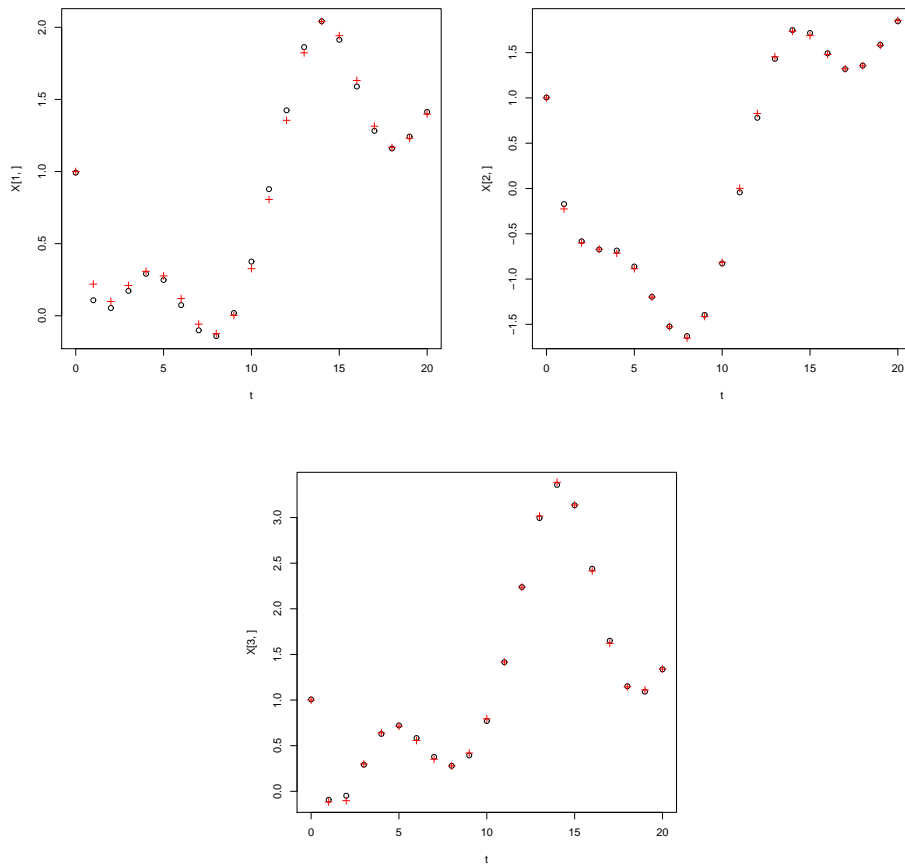


Figure 5.10: Observed gene expression(black circles) and fitted gene expression(red crosses)

Chapter 6

Future Work

6.1 More efficient way to maximize the marginal likelihood

We currently rely on an optimization method “Nelder-Mead” [5]. More efficient method based on the gradient of the marginal likelihood can be used as well.

6.2 Better accuracy

In the numerical algorithm, we have to simulate x from f . We have only used relatively simple integration technique: the composite Simpson’s rule. In order to improve the accuracy of the method, the Gaussian Process allows us to obtain estimates of f in-between observed time points(Figure 6.1). This interpolation would be useful in utilizing more accurate numerical scheme.

On the other hand, more sophisticated method in integrating the ordinary differential equations could be used, such as Runge-Kutta method. It becomes more important when the system of ordinary differential equations becomes more complicated, for example, the genes interact with each other.

6.3 Known TF concentrations

We have only considered the case when the transcription factors level is unknown. Our model can handle the case when the TF level(with noise) is known. It provides a stochastic model that take noise into account as opposed to other deterministic model.

One very interesting application would be to identify potential connections between gene expression and transcription factors(Figure 6.2). If we have mRNA levels of a gene and a list of potential transcription factors, then we can try to estimate the parameters of the model and hence the structure of the regulatory network. One idea is to compute every possible combinations and compare the fit(which is the method used in [9]). Another idea is to use an annealing procedure: first consider a weighted sum of transcription factors and gradually place more weights to only a small number of transcription factors.

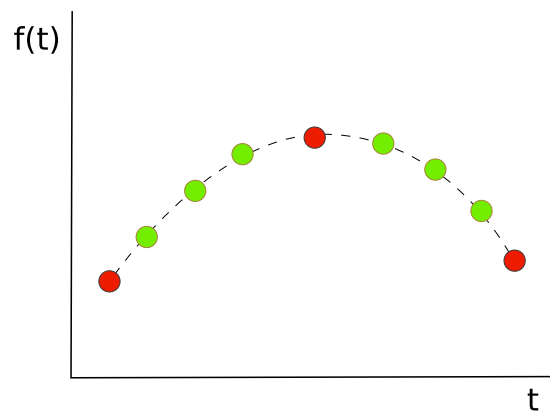


Figure 6.1: Given the function value at certain time point (red dots), the distribution of f enables us to interpolate function values (green dots) and hence obtain better numerical accuracy in the integration.

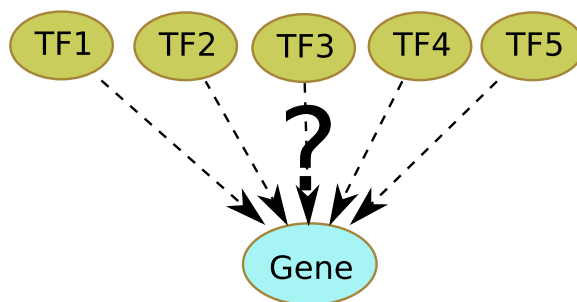


Figure 6.2: We can deduce potential link between TFs and gene expression if we have observed both gene expression and transcription factors concentration

6.4 Binding Energy Landscapes of Transcription Factors

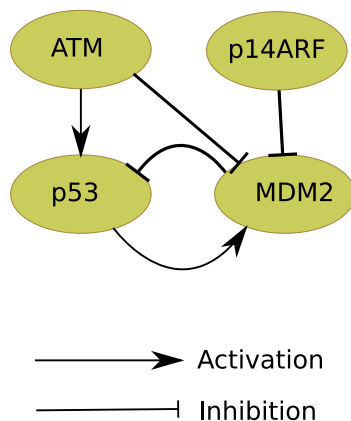
It is possible to determine how strong a transcription factor can bind to the DNA sequence experimentally [4]. One interesting application of our method is to combine gene expression data with the binding energy landscapes obtained experimentally. The binding energy landscapes information would provide a prior on the sensitivity parameters.

The potential of this model is the ability to combine different information(binding energy landscapes information and transcription factor levels), infer hidden chemical species(missing data) and handle noise. This could be a very flexible Bayesian framework in drawing together information from these two different types of experiment.

6.5 Potential Data for Analysis

It requires time-course data of gene expression with a uniform time grid. This restricts the use of many existing data. One challenge would be to generalize the method to utilize existing data with a different experimental design.

The regulatory pathway involving p53 is studied by a large number of researchers due to its connections with cancer and DNA repairs. One multiple transcription factors example is the p53 regulation described in Wilkinson [10]:



Relevant data are collected by Barenco et al. [1] and can be downloaded from the European Bioinformatics Institute Array Express website.

The yeast cycle data(Spellman et al. [8]) can provide a more complicated set of data. This is studied in a recent paper by Vu and Vohradasky [9] which investigates the use of gene expression time course data for inferring network.

In the Wilkinson [10] paper, there is also an example where a bacterium called *Bacillus Subtilis* uses transcription noise to provide stochastic switching between vegetative and competent states. The noises occurred at earlier times might lead to entirely different regulatory pathway at later time. We should be aware of such situation when considering a more complex network.

Chapter 7

Conclusion

We investigated how we can infer underlying transcription factor levels using Gaussian Process. We extended the model given by Lawrence et al. [3] to multiple transcription factors. The relevant covariance matrices and mean functions are computed for the case where the gene expression depends linearly with the transcription factor level and an algorithm is provided for non-linear case.

Our method can recover the transcription factors levels given the correct parameters. However, such parameters are unknown for real dataset. Simulation experiment suggested that the parameters are highly correlated. Extra information in fixing the parameters would help even in the simplest case of one transcription factor.

Bibliography

- [1] Martino Barenco, Daniela Tomescu, Daniel Brewer, Robin Callard, Jaroslav Stark, and Michael Hubank. Ranked prediction of p53 targets using hidden variable dynamic modeling. *Genome Biology*, 7(3), 2006.
- [2] Wit E. Khanin R, Vinciotti V. Reconstructing repressor protein levels from expression of gene targets in escherichia coli. *Proceedings of the National Academy of Sciences*, 2006.
- [3] Rattray Lawrence, Sanguinetti. Modelling transcriptional regulation using gaussian processes. 2007.
- [4] Sebastian J. Maerkl and Stephen R. Quake. A systems approach to measuring the binding energy landscapes of transcription factors. *Science*, 315(5809):233–237, January 2007.
- [5] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, January 1965.
- [6] Magnus Rattray Pei Gao, Antti Honkela and Neil D. Lawrence. Gaussian process modelling of latent chemical species: applications to inferring transcription factor activities. *Bioinformatics*, 2008.
- [7] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.
- [8] 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. *Mol Biol Cell*, 9(12):3273–3297, December 1998.
- [9] Tra T. Vu and Jiri Vohradsky. Inference of active transcriptional networks by integration of gene expression kinetics modeling and multisource data. *Genomics*, 93(5):426–433, May 2009.
- [10] Darren J. Wilkinson. Stochastic modelling for quantitative description of heterogeneous biological systems. 2009.
- [11] Ziheng Yang. *Computational Molecular Evolution (Oxford Series in Ecology and Evolution)*. Oxford University Press, USA, December 2006.