Statistical Inference in Systems Biology through Gaussian Processes and Ordinary Differential Equations

Neil D. Lawrence and Magnus Rattray
Researchers: Pei Gao, Michalis Titsias, Antti Honkela, Jennifer Withers

Max Planck Institute, Ringberg Castle

7th May 2008

Last time ...

Referring to inference of TF concentrations:
One can come up with linear (or higher order) $f(t)$ approximations on each subinterval. This will introduce additional parameters, which will be impossible to infer with any certainty given limited amount of data.

Khanin et al. (2006)

Outline

1. Introduction
2. Gaussian Process Inference for Linear Activation
3. Non-linear Response Models
4. Cascaded Differential Equations
5. Discussion and Future Work
6. Acknowledgements

Outline

1. Introduction
2. Gaussian Process Inference for Linear Activation
3. Non-linear Response Models
4. Cascaded Differential Equations
5. Discussion and Future Work
6. Acknowledgements

Talk Outline

- Models of Transcriptional Regulation
- Gaussian Process Inference for Linear Activation
- Non-linear Response Models
- Cascaded Differential Equations

Talk Outline

- Models of Transcriptional Regulation
- Gaussian Process Inference for Linear Activation
- Non-linear Response Models
- Cascaded Differential Equations

Talk Outline

- Models of Transcriptional Regulation
- Gaussian Process Inference for Linear Activation
- Non-linear Response Models
- Cascaded Differential Equations

Talk Outline

- Models of Transcriptional Regulation
- Gaussian Process Inference for Linear Activation
- Non-linear Response Models
- Cascaded Differential Equations

Talk Outline

- Models of Transcriptional Regulation
- Gaussian Process Inference for Linear Activation
- Non-linear Response Models
- Cascaded Differential Equations

ODE Model of Activation

- Linear Activation Model (Barenco et al., 2006, Genome Biology)

$$\frac{\mathrm{d}x_j\left(t\right)}{\mathrm{d}t} = B_j + S_j f\left(t\right) - D_j x_j\left(t\right)$$

- $x_j(t)$ – concentration of gene j's mRNA
- $f(t)$ – concentration of active transcription factor
- Model parameters: baseline B_j, sensitivity S_j and decay D_j
- Application: identifying co-regulated genes (targets)
- Problem: how do we fit the model when $f(t)$ is not observed?

ODE Model of Activation

- Linear Activation Model (Barenco et al., 2006, Genome Biology)

$$\frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = B_j + S_j f(t) - D_j x_j(t)$$

- $x_j(t)$ – concentration of gene j's mRNA
- $f(t)$ – concentration of active transcription factor
- Model parameters: baseline B_j, sensitivity S_j and decay D_j
- Application: identifying co-regulated genes (targets)
- Problem: how do we fit the model when $f(t)$ is not observed?

ODE Model of Activation

- Linear Activation Model (Barenco et al., 2006, Genome Biology)

$$\frac{\mathrm{d}x_j\left(t\right)}{\mathrm{d}t} = B_j + S_j f\left(t\right) - D_j x_j\left(t\right)$$

- $x_j(t)$ – concentration of gene j's mRNA
- $f(t)$ – concentration of active transcription factor
- Model parameters: baseline B_j, sensitivity S_j and decay D_j
- Application: identifying co-regulated genes (targets)
- Problem: how do we fit the model when $f(t)$ is not observed?

ODE Model of Activation

- Linear Activation Model (Barenco et al., 2006, Genome Biology)

$$\frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = B_j + S_j f(t) - D_j x_j(t)$$

- $x_j(t)$ – concentration of gene j's mRNA
- $f(t)$ – concentration of active transcription factor
- Model parameters: baseline B_j, sensitivity S_j and decay D_j
- Application: identifying co-regulated genes (targets)
- Problem: how do we fit the model when $f(t)$ is not observed?

ODE Model of Activation

- Linear Activation Model (Barenco et al., 2006, Genome Biology)

$$\frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = B_j + S_j f(t) - D_j x_j(t)$$

- $x_j(t)$ – concentration of gene j's mRNA
- $f(t)$ – concentration of active transcription factor
- Model parameters: baseline B_j, sensitivity S_j and decay D_j
- Application: identifying co-regulated genes (targets)
- Problem: how do we fit the model when $f(t)$ is not observed?

ODE Model of Activation

- Linear Activation Model (Barenco et al., 2006, Genome Biology)

$$\frac{\mathrm{d}x_j\left(t\right)}{\mathrm{d}t} = B_j + S_j f\left(t\right) - D_j x_j\left(t\right)$$

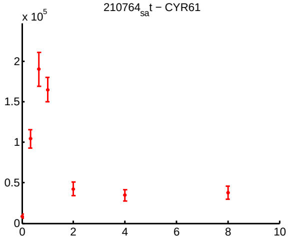
- $x_j(t)$ – concentration of gene j's mRNA
- $f(t)$ – concentration of active transcription factor
- Model parameters: baseline B_j, sensitivity S_j and decay D_j
- Application: identifying co-regulated genes (targets)
- Problem: how do we fit the model when $f(t)$ is not observed?

Why use a model-based approach?

- Co-regulated genes can differ greatly in their expression profiles

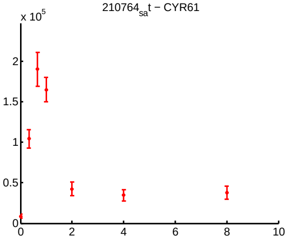
Why use a model-based approach?

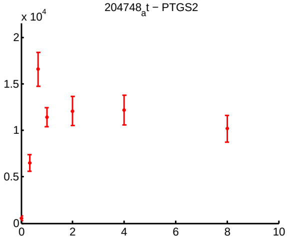
- Co-regulated genes can differ greatly in their expression profiles



Why use a model-based approach?

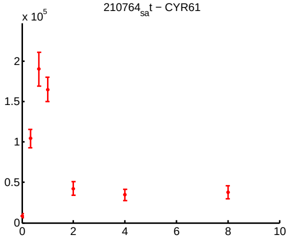
- Co-regulated genes can differ greatly in their expression profiles

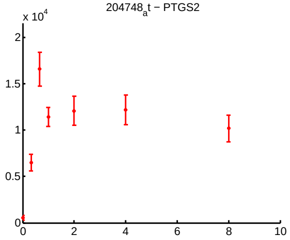




Why use a model-based approach?

- Co-regulated genes can differ greatly in their expression profiles

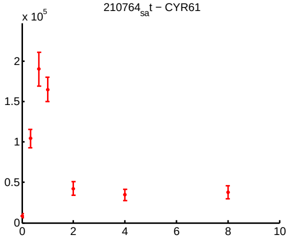


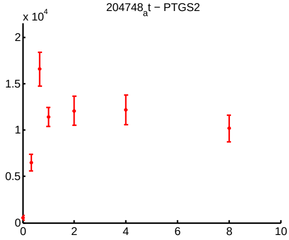


- Clustering cannot be relied on to identify co-regulated genes

Why use a model-based approach?

- Co-regulated genes can differ greatly in their expression profiles





- Clustering cannot be relied on to identify co-regulated genes
- A model-based approach is required

Models of non-linear regulation

- Non-linear Activation: Michaelis-Menten Kinetics

$$\frac{\mathrm{d}x_j\left(t\right)}{\mathrm{d}t} = B_j + \frac{S_j f\left(t\right)}{\gamma_j + f\left(t\right)} - D_j x_j\left(t\right)$$

  used by Rogers and Girolami (2006)

- Non-linear Repression

$$\frac{\mathrm{d}x_j\left(t\right)}{\mathrm{d}t} = B_j + \frac{S_j}{\gamma_j + f\left(t\right)} - D_j x_j\left(t\right)$$

  used by Khanin et al., 2006, PNAS 103

Models of non-linear regulation

- Non-linear Activation: Michaelis-Menten Kinetics

$$\frac{\mathrm{d}x_j\left(t\right)}{\mathrm{d}t} = B_j + \frac{S_j f\left(t\right)}{\gamma_j + f\left(t\right)} - D_j x_j\left(t\right)$$

used by Rogers and Girolami (2006)

- Non-linear Repression

$$\frac{\mathrm{d}x_j\left(t\right)}{\mathrm{d}t} = B_j + \frac{S_j}{\gamma_j + f\left(t\right)} - D_j x_j\left(t\right)$$

used by Khanin et al., 2006, PNAS 103

Models of non-linear regulation

- Non-linear Activation: Michaelis-Menten Kinetics

$$\frac{\mathrm{d}x_j\left(t\right)}{\mathrm{d}t} = B_j + \frac{S_j f\left(t\right)}{\gamma_j + f\left(t\right)} - D_j x_j\left(t\right)$$

  used by Rogers and Girolami (2006)

- Non-linear Repression

$$\frac{\mathrm{d}x_j\left(t\right)}{\mathrm{d}t} = B_j + \frac{S_j}{\gamma_j + f\left(t\right)} - D_j x_j\left(t\right)$$

  used by Khanin et al., 2006, PNAS 103

Standard inference approach

- Previous approaches all use similar inference methodology:
  - Represent $f(t)$ as coarse-grained piecewise continuous function $[f_1, f_2, \ldots, f_d]$
  - Often discretize where data are collected
  - Treat f_i as additional model parameters
  - Use maximum likelihood or Bayesian MCMC to estimate $\{f_i\}$ along with other model parameters of interest
- Limitations:
  - Arbitrary choice of discretization points
  - Coarse-grain gives crude approximation to $f(t)$
  - Fine-grain leads to harder inference problem

Standard inference approach

- Previous approaches all use similar inference methodology:
  - Represent $f(t)$ as coarse-grained piecewise continuous function $[f_1, f_2, \ldots, f_d]$
  - Often discretize where data are collected
  - Treat f_i as additional model parameters
  - Use maximum likelihood or Bayesian MCMC to estimate $\{f_i\}$ along with other model parameters of interest
- Limitations:
  - Arbitrary choice of discretization points
  - Coarse-grain gives crude approximation to $f(t)$
  - Fine-grain leads to harder inference problem

Standard inference approach

- Previous approaches all use similar inference methodology:
  - Represent $f(t)$ as coarse-grained piecewise continuous function $[f_1, f_2, \ldots, f_d]$
  - Often discretize where data are collected
  - Treat f_i as additional model parameters
  - Use maximum likelihood or Bayesian MCMC to estimate $\{f_i\}$ along with other model parameters of interest
- Limitations:
  - Arbitrary choice of discretization points
  - Coarse-grain gives crude approximation to $f(t)$
  - Fine-grain leads to harder inference problem

Standard inference approach

- Previous approaches all use similar inference methodology:
    - Represent $f(t)$ as coarse-grained piecewise continuous function $[f_1, f_2, \ldots, f_d]$
    - Often discretize where data are collected
    - Treat f_i as additional model parameters
    - Use maximum likelihood or Bayesian MCMC to estimate $\{f_i\}$ along with other model parameters of interest
- Limitations:
    - Arbitrary choice of discretization points
    - Coarse-grain gives crude approximation to $f(t)$
    - Fine-grain leads to harder inference problem

Standard inference approach

- Previous approaches all use similar inference methodology:
  - Represent $f(t)$ as coarse-grained piecewise continuous function $[f_1, f_2, \ldots, f_d]$
  - Often discretize where data are collected
  - Treat f_i as additional model parameters
  - Use maximum likelihood or Bayesian MCMC to estimate $\{f_i\}$ along with other model parameters of interest
- Limitations:
  - Arbitrary choice of discretization points
  - Coarse-grain gives crude approximation to $f(t)$
  - Fine-grain leads to harder inference problem

Standard inference approach

- Previous approaches all use similar inference methodology:
  - Represent $f(t)$ as coarse-grained piecewise continuous function $[f_1, f_2, \ldots, f_d]$
  - Often discretize where data are collected
  - Treat f_i as additional model parameters
  - Use maximum likelihood or Bayesian MCMC to estimate $\{f_i\}$ along with other model parameters of interest
- Limitations:
  - Arbitrary choice of discretization points
  - Coarse-grain gives crude approximation to $f(t)$
  - Fine-grain leads to harder inference problem

Standard inference approach

- Previous approaches all use similar inference methodology:
  - Represent $f(t)$ as coarse-grained piecewise continuous function $[f_1, f_2, \ldots, f_d]$
  - Often discretize where data are collected
  - Treat f_i as additional model parameters
  - Use maximum likelihood or Bayesian MCMC to estimate $\{f_i\}$ along with other model parameters of interest
- Limitations:
  - Arbitrary choice of discretization points
  - Coarse-grain gives crude approximation to $f(t)$
  - Fine-grain leads to harder inference problem

Standard inference approach

- Previous approaches all use similar inference methodology:
  - Represent $f(t)$ as coarse-grained piecewise continuous function $[f_1, f_2, \ldots, f_d]$
  - Often discretize where data are collected
  - Treat f_i as additional model parameters
  - Use maximum likelihood or Bayesian MCMC to estimate $\{f_i\}$ along with other model parameters of interest
- Limitations:
  - Arbitrary choice of discretization points
  - Coarse-grain gives crude approximation to $f(t)$
  - Fine-grain leads to harder inference problem

Standard inference approach

- Previous approaches all use similar inference methodology:
  - Represent $f(t)$ as coarse-grained piecewise continuous function $[f_1, f_2, \ldots, f_d]$
  - Often discretize where data are collected
  - Treat f_i as additional model parameters
  - Use maximum likelihood or Bayesian MCMC to estimate $\{f_i\}$ along with other model parameters of interest
- Limitations:
  - Arbitrary choice of discretization points
  - Coarse-grain gives crude approximation to $f(t)$
  - Fine-grain leads to harder inference problem

Outline

1. Introduction
2. Gaussian Process Inference for Linear Activation
3. Non-linear Response Models
4. Cascaded Differential Equations
5. Discussion and Future Work
6. Acknowledgements

Gaussian Processes

- Gaussian Process

$$f(t) \backsim \mathcal{GP}\left(m(t), k\left(t, t'\right)\right)$$

where

$$\begin{aligned} m(t) &= \mathbb{E}\left[f(t)\right] = \langle f(t)\rangle \\ k\left(t, t'\right) &= \mathbb{E}\left[\left(f(t) - m(t)\right)\left(f\left(t'\right) - m\left(t'\right)\right)\right] \end{aligned}$$

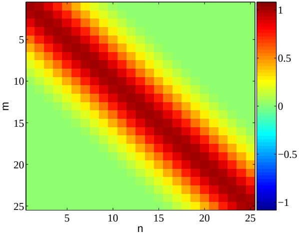
▸ Skip Covariance Functions

Covariance Functions

RBF Kernel Function

$$k\left(t, t'\right) = \alpha \exp\left(-\frac{\left(t - t'\right)^2}{2l^2}\right)$$

- Covariance matrix is built using the *inputs* to the function t.
- For the example above it was based on Euclidean distance.
- The covariance function is also know as a kernel.

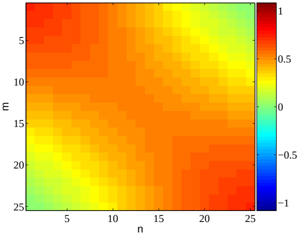


Different Covariance Functions

MLP Kernel Function

$$k\left(t,t'\right)=\alpha\sin^{-1}\left(\frac{wtt'+b}{\sqrt{wt^{2}+b+1}\sqrt{wt'^{2}+b+1}}\right)$$

- A non-stationary covariance matrix (Williams, 1997).
- Derived from a multi-layer perceptron (MLP).



Covariance Samples

`demCovFuncSample`

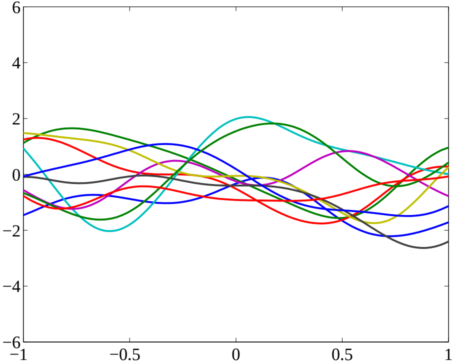


Figure: RBF kernel with $\gamma = 10^{-\frac{1}{2}}$, $\alpha = 1$

Covariance Samples

`demCovFuncSample`

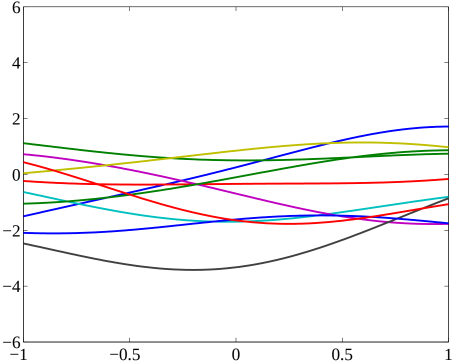


Figure: RBF kernel with $l = 1$, $\alpha = 1$

Covariance Samples

`demCovFuncSample`

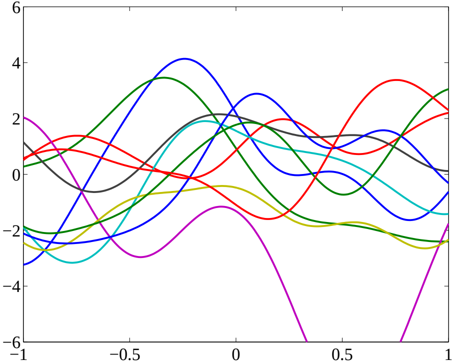


Figure: RBF kernel with $l = 0.3$, $\alpha = 4$

Covariance Samples

`demCovFuncSample`

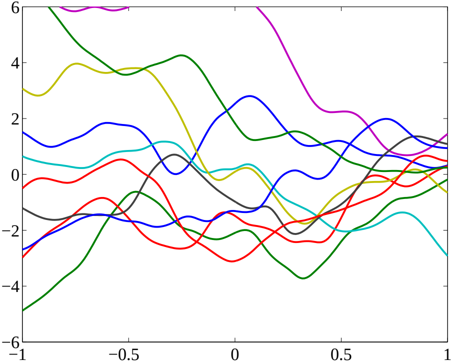


Figure: MLP kernel with $\alpha = 8$, $w = 100$ and $b = 100$

Covariance Samples

`demCovFuncSample`

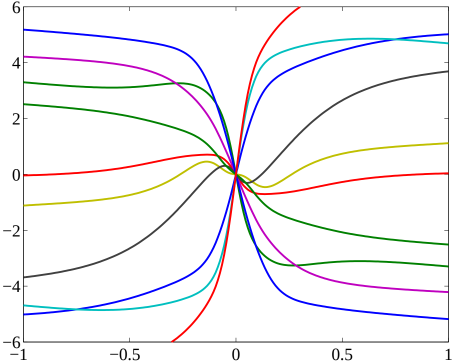


Figure: MLP kernel with $\alpha = 8$, $b = 0$ and $w = 100$

Linear Activation Model

Recall the linear model

$$\frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = B_j + S_j f(t) - D_j x_j(t) \ .$$

This differential equation can be solved for $x_j(t)$ as

$$x_j(t) = \frac{B_j}{D_j} + S_j \int_0^t e^{-D_j(t-u)} f(u) \,\mathrm{d}u \ .$$

Note: This is a linear operation on $f(t)$.

If $f(t)$ is a zero mean Gaussian process then $x_i(t)$ is also a Gaussian process with mean $\frac{B_i}{D_i}$.

▸ Skip GP Properties

Two Properties of GPs

The integral of a GP is also a GP,

$$f\left(t\right) \sim N\left(\mathbf{0}, \mathbf{K}_{ff}\right)$$

and

$$g\left(t\right) = \int_0^t f\left(u\right) du$$

then

$$g\left(t\right) \sim N\left(\mathbf{0}, \mathbf{K}_{gg}\right),$$

where

$$k_{gg}\left(t, t'\right) = \int_0^t \int_0^{t'} k_{ff}\left(u, u'\right) du du'$$

Two Properties of GPs

Product with deterministic function

Product with a deterministic funciton leads to another GP,

$$f(t) \sim N(\mathbf{0}, \mathbf{K}_{ff}),$$

and

$$g(t) = f(t)\, h(t)$$

where $h(t)$ is a deterministic function then,

$$g(t) \sim N(\mathbf{0}, \mathbf{K}_{gg}),$$

where

$$k_{gg}(t, t') = h(t)\, k_{ff}(t, t')\, h(t')$$

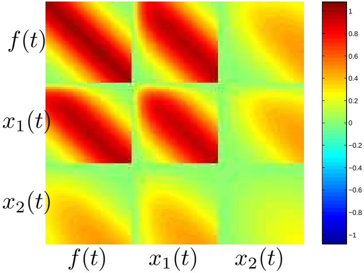
Covariance for Transcription Model

RBF covariance function for $f(t)$

$$x_i(t) = \frac{B_i}{D_i} + S_i \exp(-D_i t) \int_0^t f(u) \exp(D_i u) \, \mathrm{d}u.$$

- Joint distribution for $x_1(t)$, $x_2(t)$ and $f(t)$.
  - Here:

D_1	S_1	D_2	S_2
5	5	0.5	0.5



▸ Skip SIM Samples

Joint Sampling of $x(t)$ and $f(t)$ from Covariance

`gpsimTest`

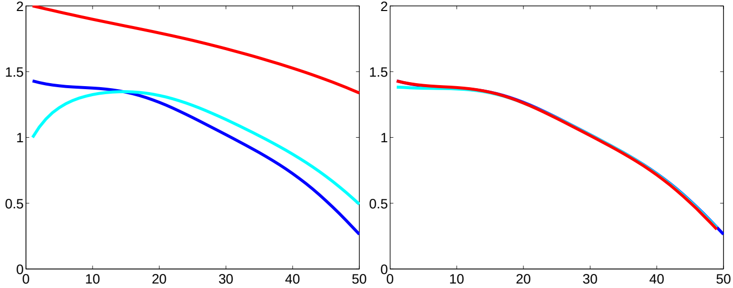


Figure: *Left*: joint samples from the transcription covariance, *blue*: $f(t)$, *cyan*: $x_1(t)$ and *red*: $x_2(t)$. *Right*: numerical solution for $f(t)$ of the differential equation from $x_1(t)$ and $x_2(t)$ (blue and cyan). True $f(t)$ included for comparison.

Joint Sampling of $x(t)$ and $f(t)$ from Covariance

```
gpsimTest
```

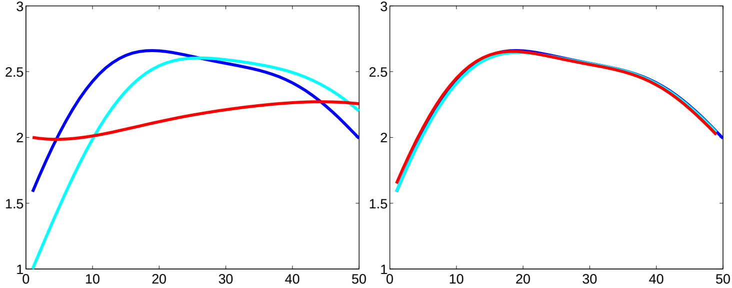


Figure: *Left*: joint samples from the transcription covariance, *blue*: $f(t)$, *cyan*: $x_1(t)$ and *red*: $x_2(t)$. *Right*: numerical solution for $f(t)$ of the differential equation from $x_1(t)$ and $x_2(t)$ (blue and cyan). True $f(t)$ included for comparison.

Joint Sampling of $x(t)$ and $f(t)$ from Covariance

`gpsimTest`

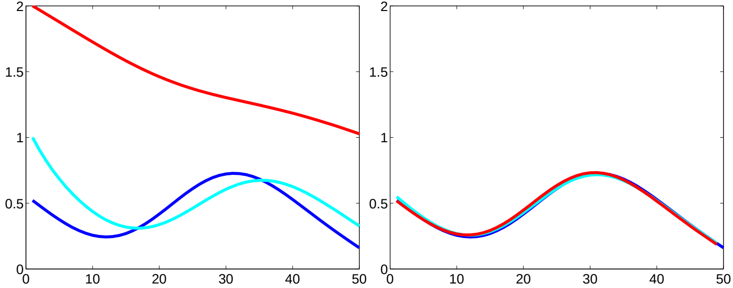


Figure: *Left*: joint samples from the transcription covariance, *blue*: $f(t)$, *cyan*: $x_1(t)$ and *red*: $x_2(t)$. *Right*: numerical solution for $f(t)$ of the differential equation from $x_1(t)$ and $x_2(t)$ (blue and cyan). True $f(t)$ included for comparison.

Covariance Function

Any linear opearation of a GP $\Longrightarrow$ Related GP

$$f\left(t\right) \backsim \mathcal{GP}\left(0, k_{ff}\left(t, t'\right)\right) \Longrightarrow x_j\left(t\right) \backsim \mathcal{GP}\left(\frac{B_j}{D_j}, k_{xx}\left(t, t'\right)\right)$$

Hence, the cross-covariances between the genes is

$$k_{x_i,x_j}\left(t, t'\right) = S_i S_j \int_0^t \int_0^{t'} e^{-D_i(t-u)-D_j(t'-u')} k_{f,f}\left(t, t'\right) \mathrm{d}u \mathrm{d}u' \ .$$

Cross-covariances between $x_j\left(t\right)$ and $f\left(t\right)$ is

$$k_{x_j,f}\left(t, t'\right) = \int_0^t e^{-D_i(t-u)} k_{f,f}\left(t, t'\right) \mathrm{d}u \ .$$

Prediction of the transcription factor concentration $f(t)$

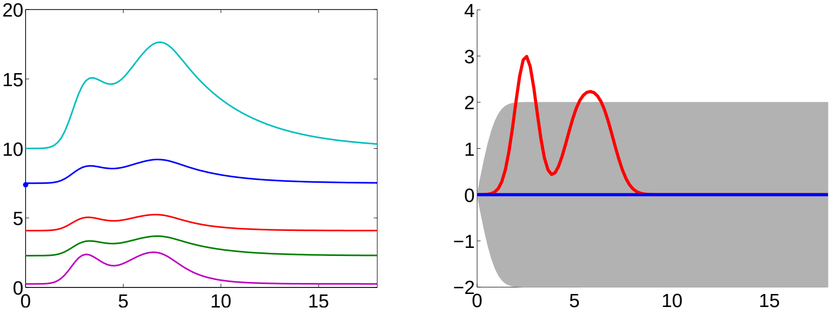
Under the linear model, we have

$$\begin{bmatrix} f \\ \mathbf{x} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ \frac{\mathbf{B}}{\mathbf{D}} \end{bmatrix}, \begin{bmatrix} K_{ff} & K_{f\mathbf{x}} \\ K_{\mathbf{x}f} & K_{\mathbf{x}\mathbf{x}} \end{bmatrix}\right)$$

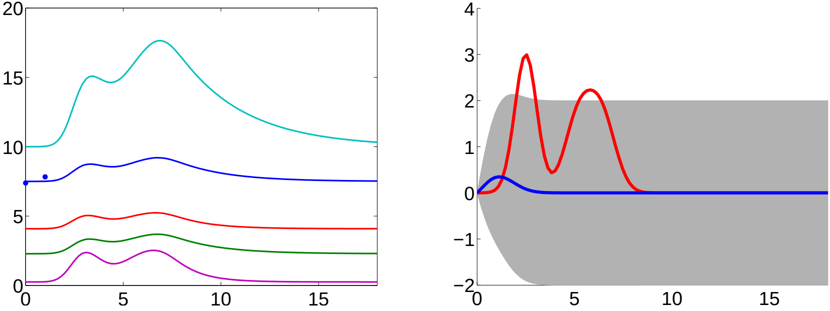
Standard GP Regression yields the mean and covariance function of the predicted process as

$$\begin{aligned} \langle f \rangle_{post} &= K_{f\mathbf{x}} K_{\mathbf{x}\mathbf{x}}^{-1} \left(\mathbf{x} - \frac{\mathbf{B}}{\mathbf{D}} \right) \\ K_{ff}^{post} &= K_{ff} - K_{f\mathbf{x}} K_{\mathbf{x}\mathbf{x}}^{-1} K_{\mathbf{x}f} \end{aligned}$$

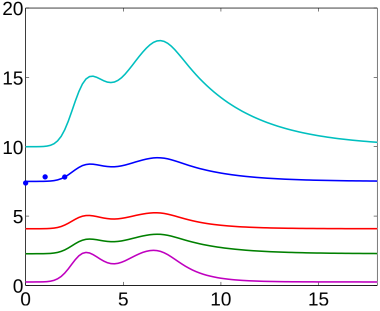
Artificial Example: Inferring $f(t)$

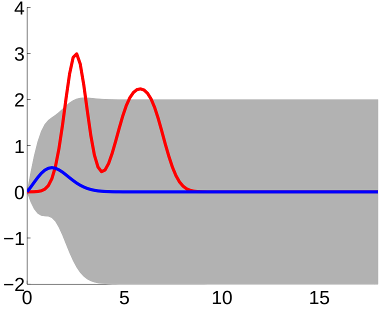


Artificial Example: Inferring $f(t)$

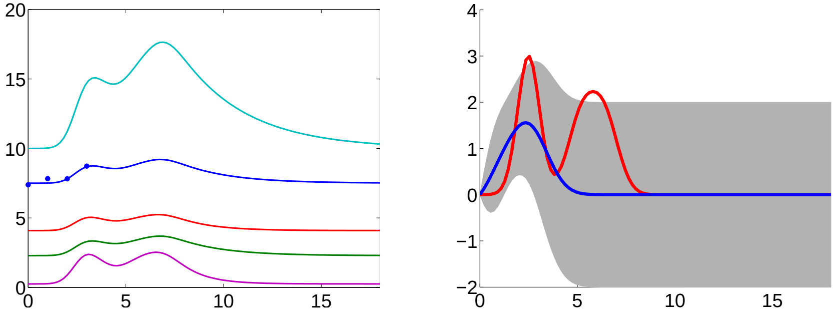


Artificial Example: Inferring $f(t)$

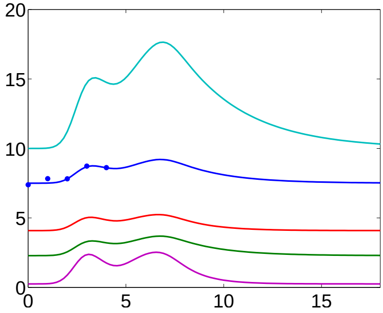


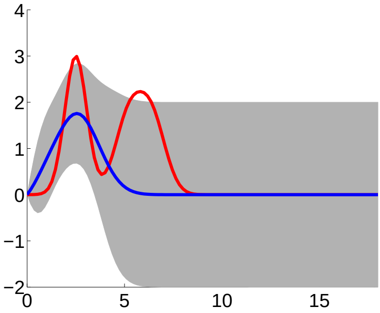


Artificial Example: Inferring $f(t)$

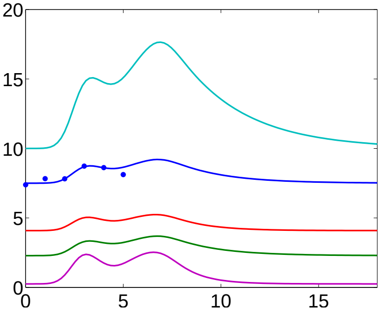


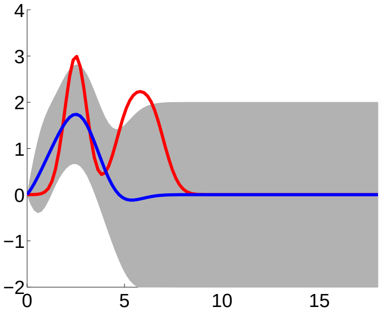
Artificial Example: Inferring $f(t)$



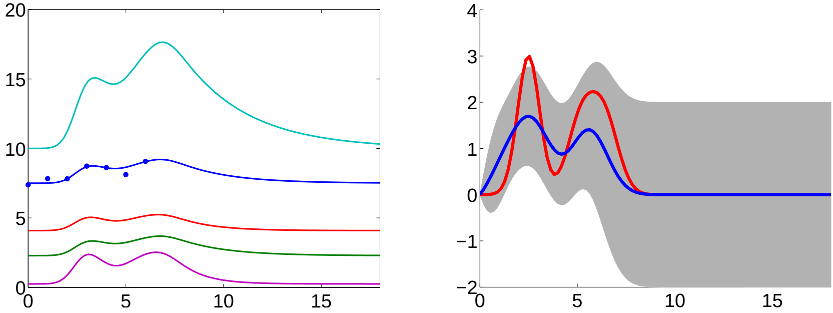


Artificial Example: Inferring $f(t)$

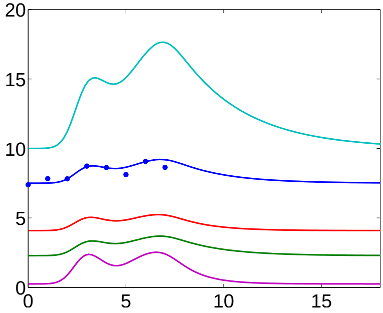


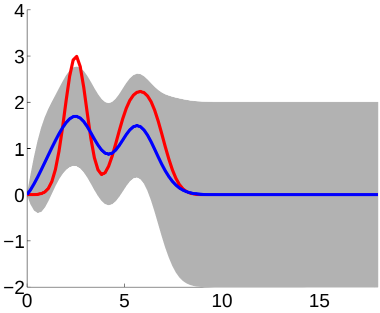


Artificial Example: Inferring $f(t)$

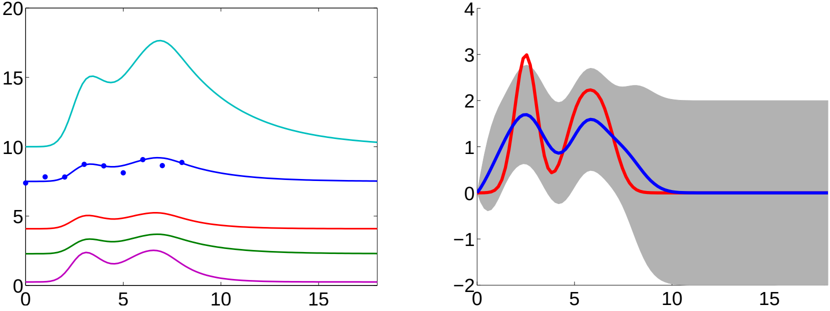


Artificial Example: Inferring $f(t)$

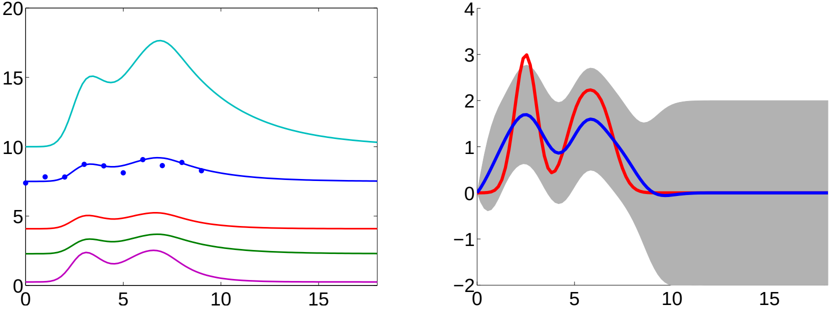




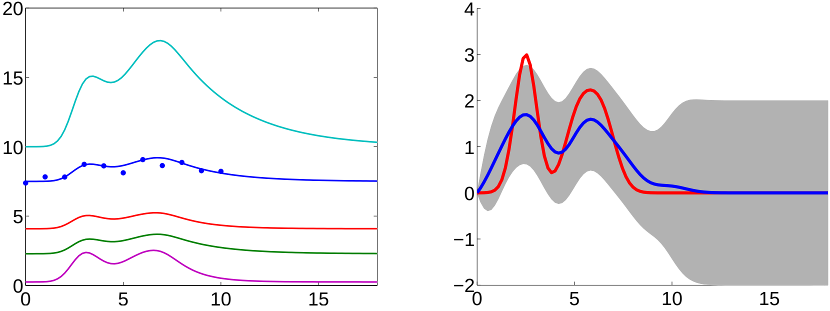
Artificial Example: Inferring $f(t)$



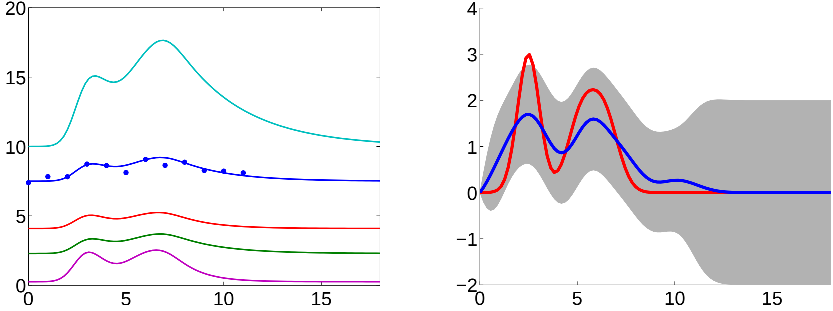
Artificial Example: Inferring $f(t)$



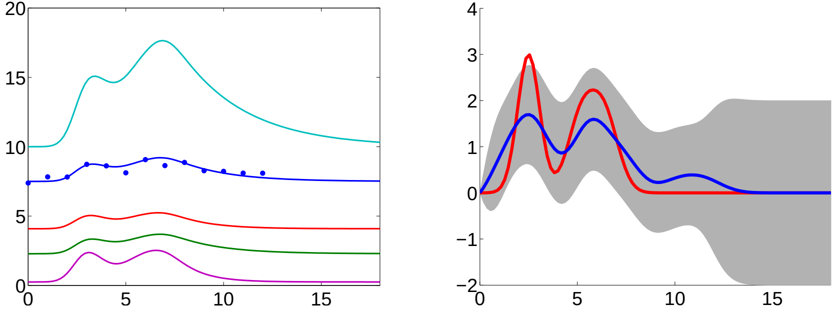
Artificial Example: Inferring $f(t)$



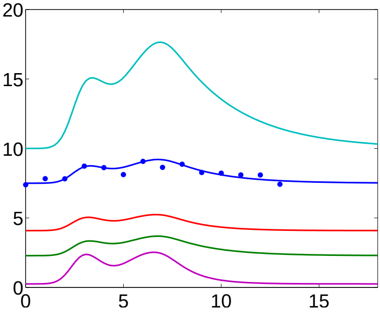
Artificial Example: Inferring $f(t)$

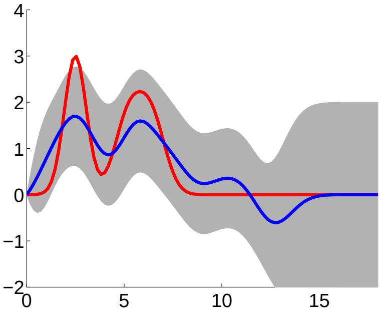


Artificial Example: Inferring $f(t)$

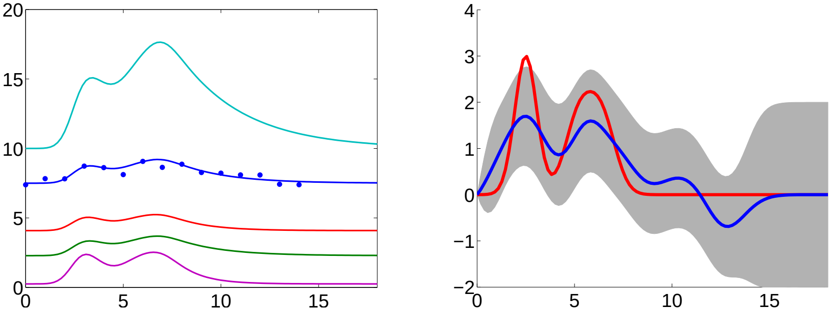


Artificial Example: Inferring $f(t)$

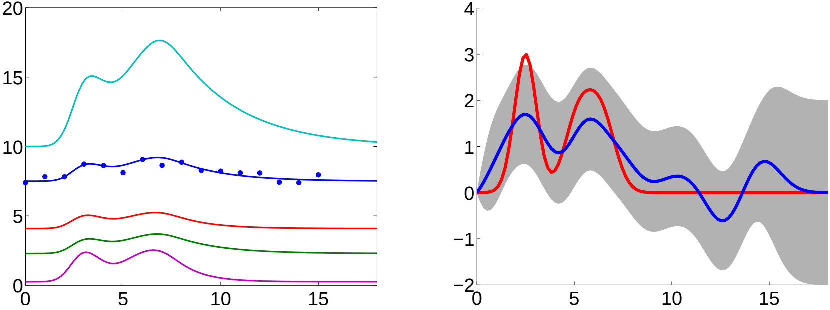




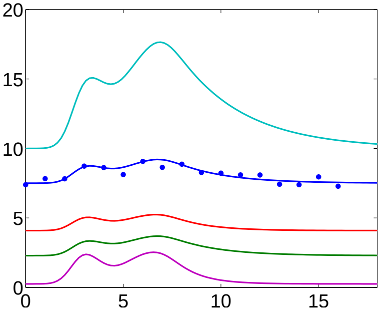
Artificial Example: Inferring $f(t)$

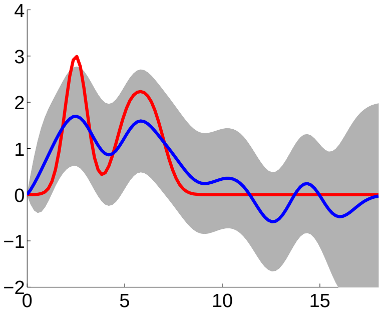


Artificial Example: Inferring $f(t)$

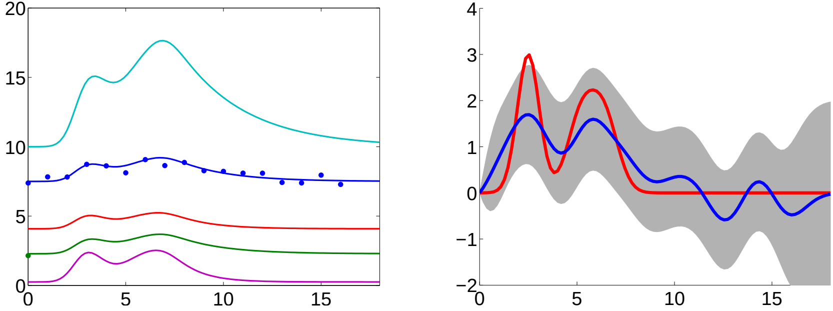


Artificial Example: Inferring $f(t)$

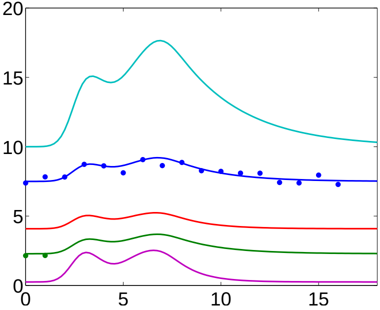


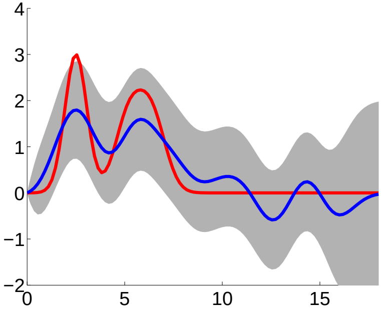


Artificial Example: Inferring $f(t)$

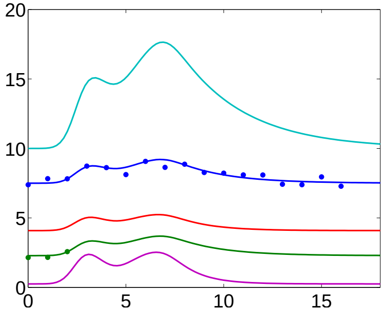


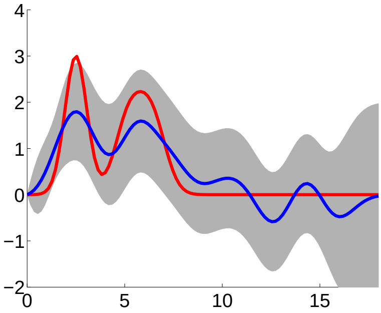
Artificial Example: Inferring $f(t)$



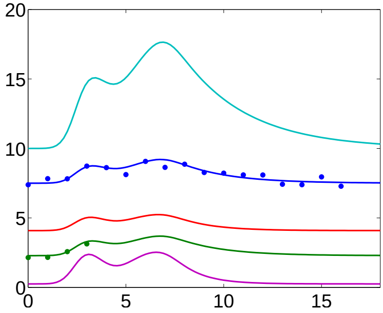


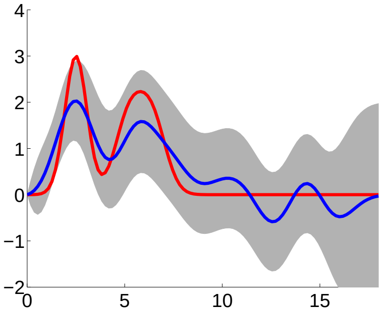
Artificial Example: Inferring $f(t)$



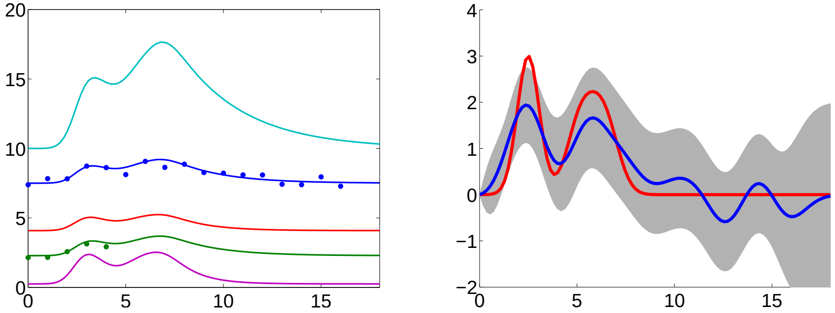


Artificial Example: Inferring $f(t)$

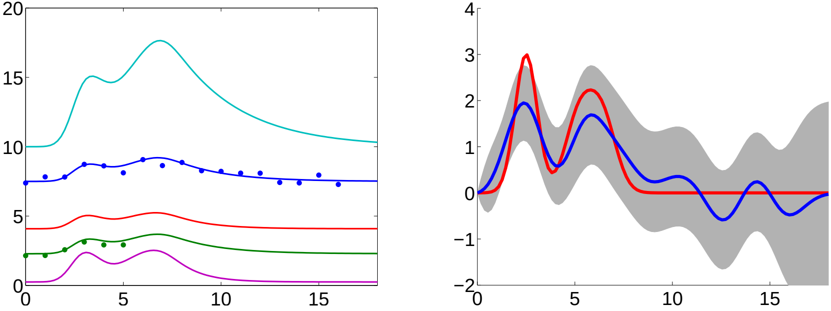




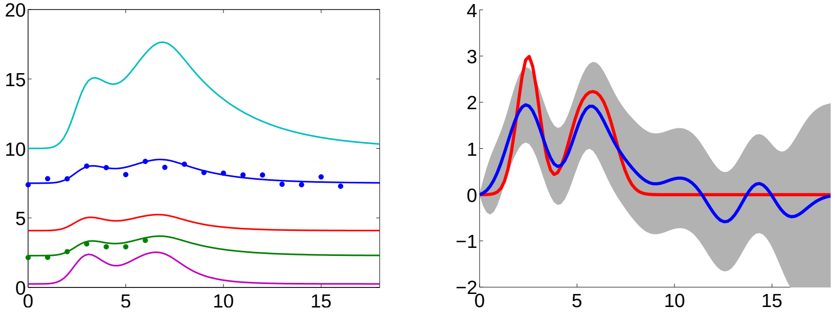
Artificial Example: Inferring $f(t)$



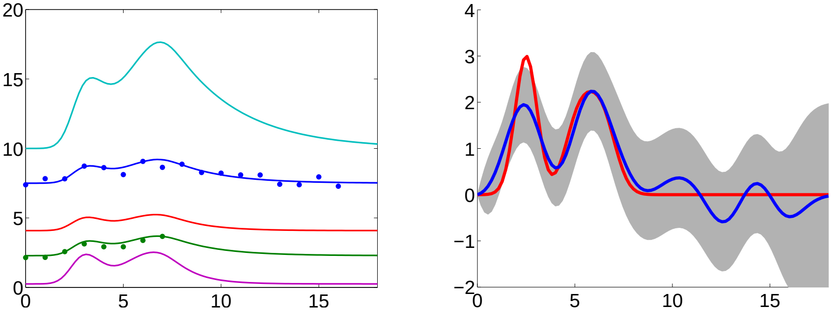
Artificial Example: Inferring $f(t)$



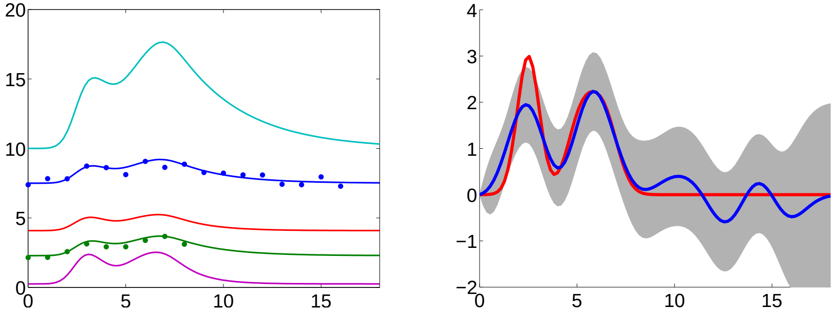
Artificial Example: Inferring $f(t)$



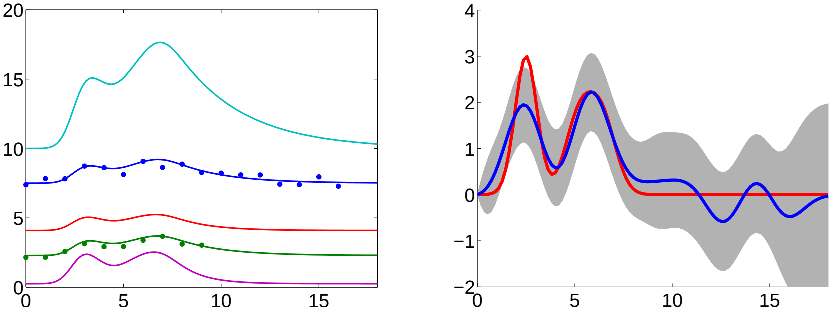
Artificial Example: Inferring $f(t)$



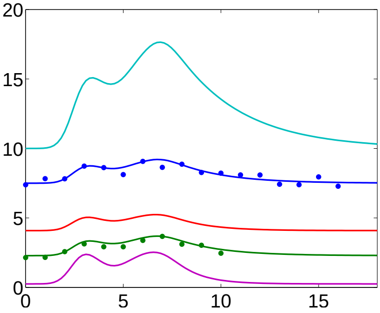
Artificial Example: Inferring $f(t)$

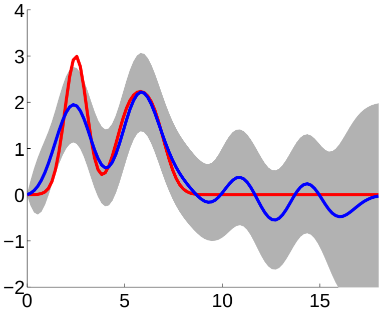


Artificial Example: Inferring $f(t)$

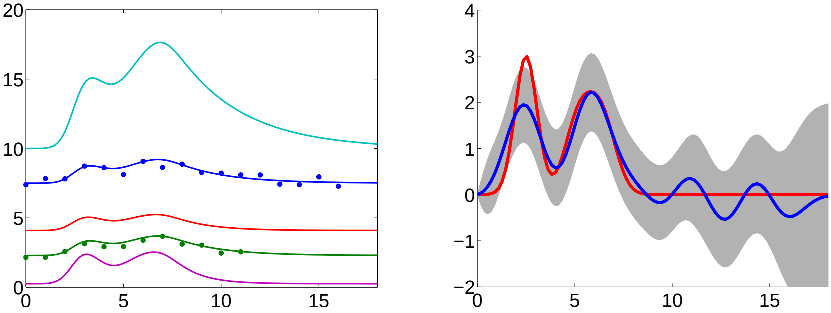


Artificial Example: Inferring $f(t)$

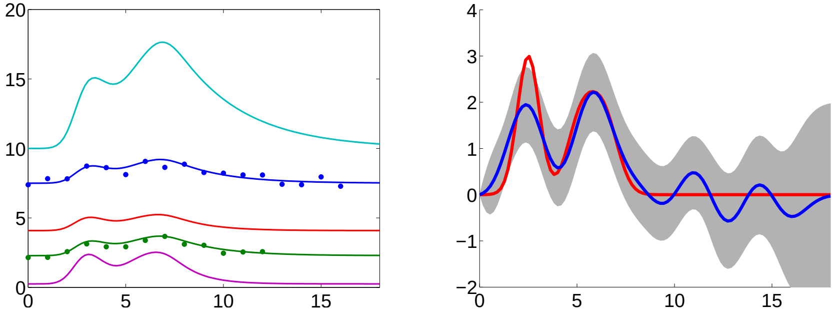




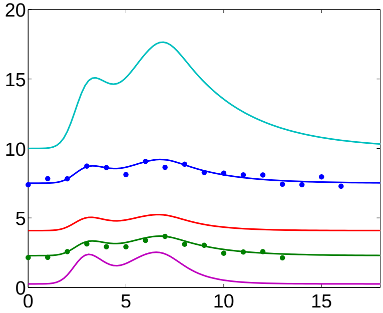
Artificial Example: Inferring $f(t)$

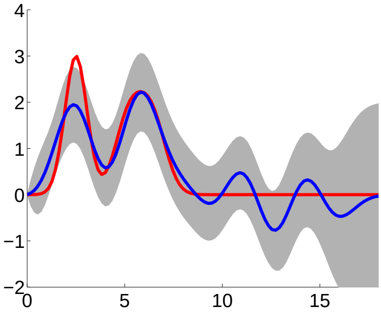


Artificial Example: Inferring $f(t)$

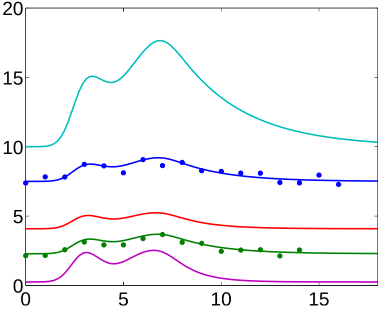


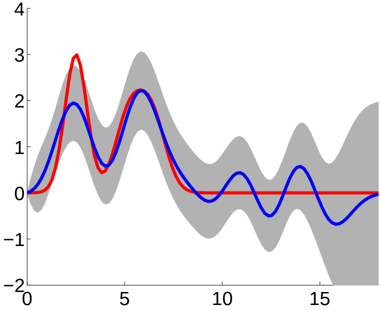
Artificial Example: Inferring $f(t)$



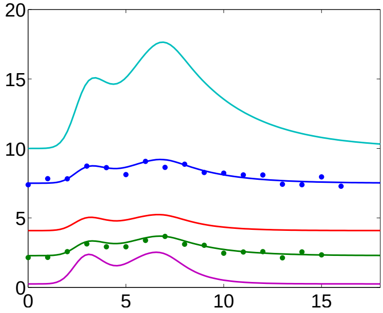


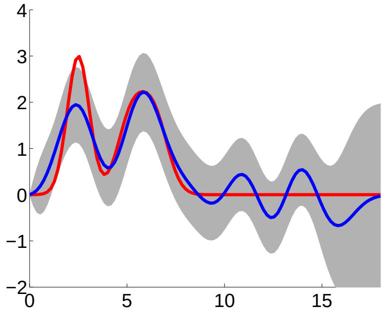
Artificial Example: Inferring $f(t)$



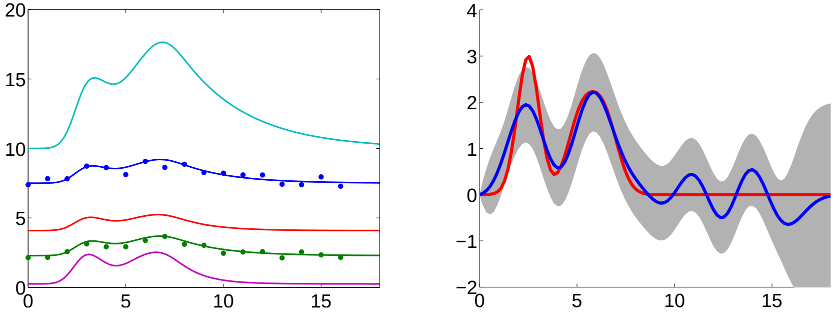


Artificial Example: Inferring $f(t)$

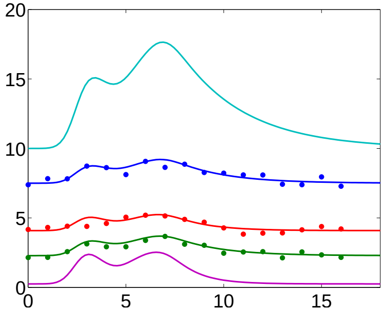


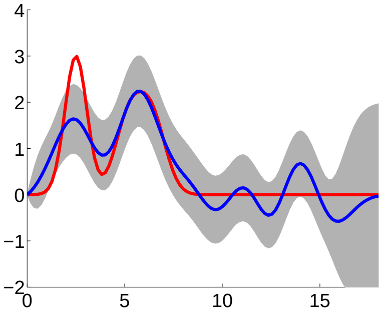


Artificial Example: Inferring $f(t)$

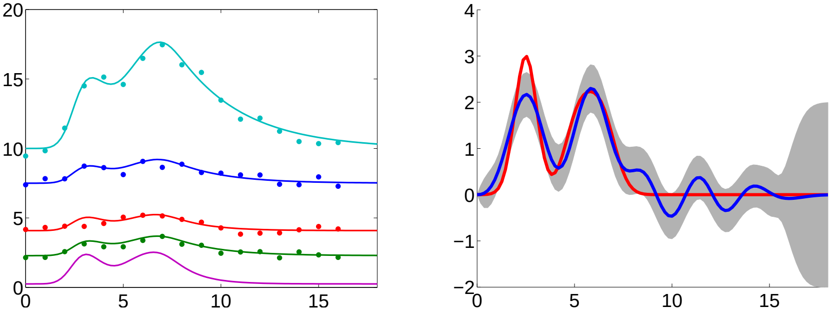


Artificial Example: Inferring $f(t)$

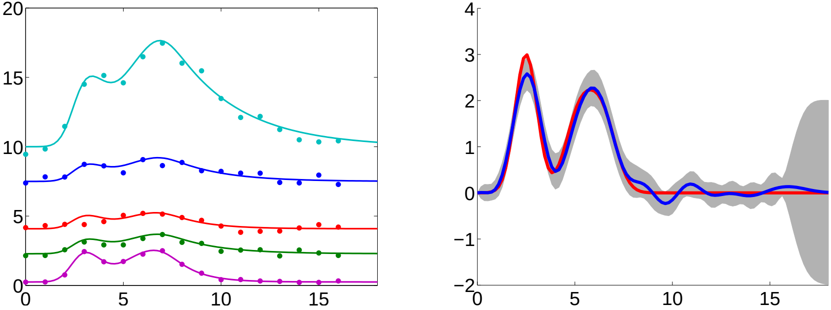




Artificial Example: Inferring $f(t)$



Artificial Example: Inferring $f(t)$



Parameter Estimation for the Linear Model

A likelihood function for the model parameters $\theta = \{B_j,\, S_j,\, D_j\}_{j=1}^{N}$ and GP length scale l is obtained by *integrating out* the latent function $f(t)$

$$L(\theta, l) = \int \left(\prod_j p(x_j|\theta, f(t)) \right) p(f(t)|l)\, \mathrm{d}f(t)$$

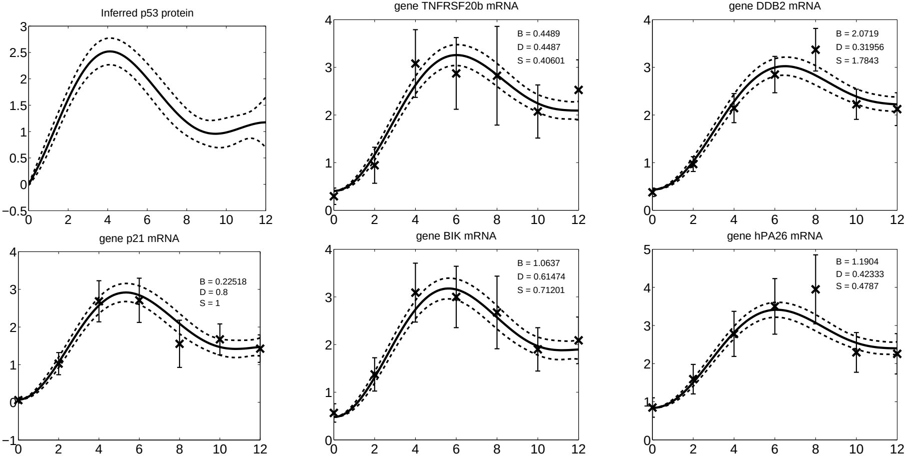
Under the GP model, the log marginal likelihood is then given by

$$\log L(\theta, l) = -\frac{1}{2} x^T \left(K + \sigma_n^2 \mathrm{I}\right)^{-1} x - \frac{1}{2} \log \left|K + \sigma_n^2 \mathrm{I}\right| - \frac{n}{2} \log 2\pi$$

Maximise to find model parameters.

p53 (RBF covariance)

Pei Gao

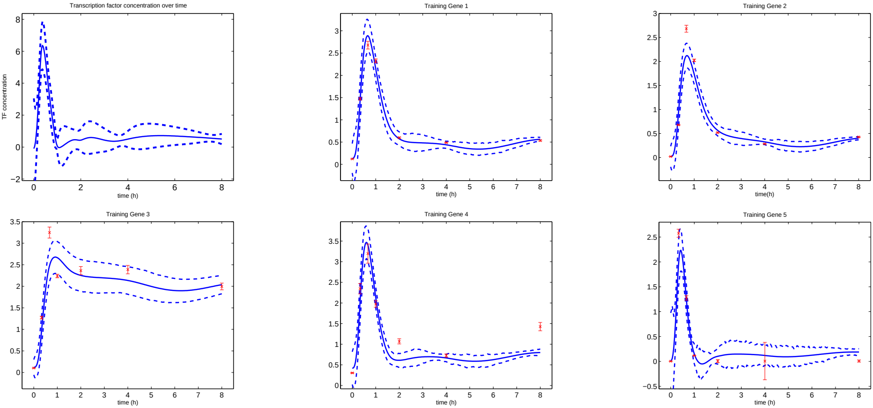


Ranking with ERK Signalling

- Target Ranking for Elk-1.
- Elk-1 is phosphorylated by ERK from the EGF signalling pathway.
- Predict concentration of Elk-1 from known targets.
- Rank other targets of Elk-1.

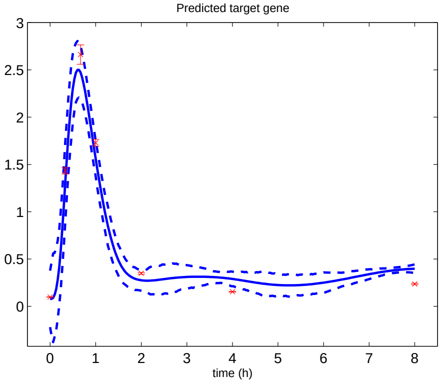
Elk-1 (MLP covariance)

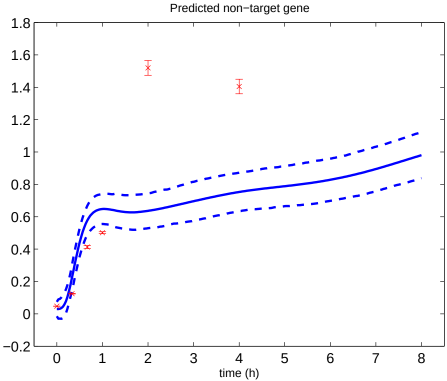
Jennifer Withers



Elk-1 target selection

Fitted model used to rank potential targets of Elk-1





Outline

1. Introduction
2. Gaussian Process Inference for Linear Activation
3. Non-linear Response Models
4. Cascaded Differential Equations
5. Discussion and Future Work
6. Acknowledgements

Nonlinear Response Models

Consider the following modification to the model,

$$\frac{\mathrm{d}x_j\left(t\right)}{\mathrm{d}t} = B_j + S_j g\left(f\left(t\right)\right) - D_j x_j\left(t\right),$$

where $g\left(\cdot\right)$ is a non-linear function. The differential equation can still be solved,

$$x_j\left(t\right) = \frac{B_j}{D_j} + S_j \int_0^t e^{-D_j(t-u)} g_j\left(f\left(u\right)\right) \mathrm{d}u$$

MAP-Laplace Approximation

Based on Laplace's method,

$$p\left(\mathbf{f} \mid \mathbf{x}\right) = N\left(\hat{\mathbf{f}}, \mathbf{A}^{-1}\right) \propto \exp\left(-\frac{1}{2}\left(\mathbf{f} - \hat{\mathbf{f}}\right)^{T}\mathbf{A}\left(\mathbf{f} - \hat{\mathbf{f}}\right)\right)$$

where $\hat{\mathbf{f}} = \mathrm{argmax} p(\mathbf{f} \mid \mathbf{x})$ and $\mathbf{A} = -\nabla\nabla \log p\left(\mathbf{f} \mid \mathbf{y}\right)|_{\mathbf{f}=\hat{\mathbf{f}}}$ is the Hessian of the negative posterior at that point. To obtain $\hat{\mathbf{f}}$ and $\mathbf{A}$, we define the

following function $\psi\left(\mathbf{f}\right)$ as:

$$\log p(\mathbf{f}|\mathbf{x}) \propto \psi(\mathbf{f}) = \log p\left(\mathbf{x} \mid \mathbf{f}\right) + \log p\left(\mathbf{f}\right)$$

MAP-Laplace Approximation

Assigning a GP prior distribution to $f(t)$, it then follows that

$$\log p(\mathbf{f}) = -\frac{1}{2}\mathbf{f}^T\mathbf{K}^{-1}\mathbf{f} - \frac{1}{2}\log|\mathbf{K}| - \frac{n}{2}\log 2\pi$$

where $\mathbf{K}$ is the covariance matrix of $f(t)$. Hence,

$$\begin{aligned}\nabla\psi(\mathbf{f}) &= \nabla\log p(\mathbf{x}|\mathbf{f}) - \mathbf{K}^{-1}\mathbf{f}\\ \nabla\nabla\psi(\mathbf{f}) &= \nabla\nabla\log p(\mathbf{x}|\mathbf{f}) - \mathbf{K}^{-1} = -\mathbf{W} - \mathbf{K}^{-1}\end{aligned}$$

Estimation of $\psi(\mathbf{f})$

Newton's method is applied to find the maximum of $\psi(\mathbf{f})$ as

$$\begin{aligned}\mathbf{f}^{new} &= \mathbf{f} - (\nabla\nabla\psi(\mathbf{f}))^{-1}\nabla\psi(\mathbf{f}) \\ &= (\mathbf{W} + \mathbf{K}^{-1})^{-1}\left(\mathbf{W}\mathbf{f} - \nabla \log p(\mathbf{x}|\mathbf{f})\right)\end{aligned}$$

In addition, $\mathbf{A} = -\nabla\nabla\psi(\hat{f}) = \mathbf{W} + \mathbf{K}^{-1}$ where $\mathbf{W}$ is the negative Hessian matrix. Hence, the Laplace approximation to the posterior is a Gaussian with mean $\hat{\mathbf{f}}$ and covariance matrix $\mathbf{A}^{-1}$as

$$p(\mathbf{f} \mid \mathbf{x}) \simeq N(\hat{\mathbf{f}}, \mathbf{A}^{-1}) = N(\hat{\mathbf{f}}, (\mathbf{W} + \mathbf{K}^{-1})^{-1})$$

Model Parameter Estimation

The marginal likelihood is useful for estimating the model parameters θ and covariance parameters l

$$p\left(\mathbf{x}|\boldsymbol{\theta},\phi\right) = \int p\left(\mathbf{x}|\mathbf{f},\boldsymbol{\theta}\right) p\left(\mathbf{f}|\phi\right) df = \int \exp\left(\psi\left(\mathbf{f}\right)\right) df$$

Using Taylor expansion of $\psi(\mathbf{f})$,

$$\log p(\mathbf{x}|\boldsymbol{\theta},\phi) = \log p\left(\mathbf{x}|\hat{\mathbf{f}},\boldsymbol{\theta},\phi\right) - \frac{1}{2}\mathbf{f}^{T}\mathbf{K}^{-1}\mathbf{f} - \frac{1}{2}\log\left|\mathbf{I}+\mathbf{K}\mathbf{W}\right|$$

The parameters $\boldsymbol{\eta} = \{\boldsymbol{\theta},\boldsymbol{\phi}\}$ can be then estimated by using

$$\frac{\partial \log p\left(\mathbf{x}|\boldsymbol{\eta}\right)}{\partial \boldsymbol{\eta}} = \frac{\partial \log p\left(\mathbf{x}|\boldsymbol{\eta}\right)}{\partial \boldsymbol{\eta}} \,|_{\text{explicit}} + \frac{\partial \log p\left(\mathbf{x}|\boldsymbol{\eta}\right)}{\partial \hat{\mathbf{f}}} \frac{\partial \hat{\mathbf{f}}}{\partial \boldsymbol{\eta}}$$

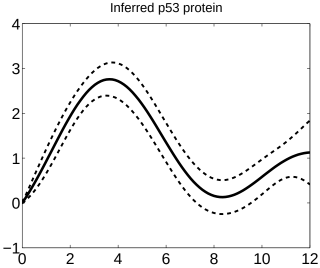
Michaelis-Menten Kinetics

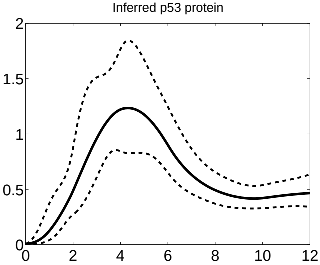
Pei Gao

- The Michaelis-Menten activation model uses the following non-linearity

$$g_j\left(f\left(t\right)\right) = \frac{e^{f(t)}}{\gamma_j + e^{f(t)}},$$

where we are using a GP $f\left(t\right)$ to model the log of the TF activity.





(a)

Valdiation of Laplace Approximation

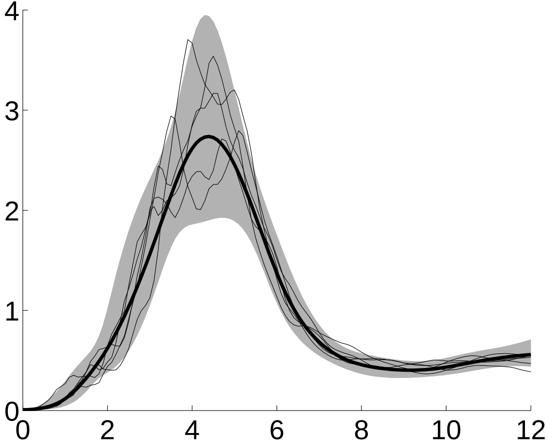


Figure: Laplace approximation error bars along with samples from the true posterior distribution.

SOS Response

- DNA damage may occur as a result of activity of antibiotics.
- LexA is bound to the genome preventing transcription of the SOS genes.
- RecA protein is stimulated by single stranded DNA, inactivates the LexA repessor.
- This allows several of the LexA targets to transcribe.
- The SOS pathway may be essential in antibiotic resistance Cirz et al. (2005).
- Aim is to target these proteins to produce drugs to increase efficacy of antibiotics Lee et al. (2005).

LexA Experimental Description

- Data from Courcelle et al. (2001)
- UV irradiation of *E. coli.* in both wild-type cells and lexA1 mutants, which are unable to induce genes under LexA control.
- Response measured with two color hybridization to cDNA arrays.

Their Model

Given measurements of gene expression at N time points $(t_0, t_1, \ldots, t_{N-1})$, the temporal profile of a gene i, $x_i(t)$, that solves the ODE in Eq. 1 can be approximated by

$$x_i(t) = x_i^0 e^{-\delta_i t} + \frac{B_i}{D_i} + S_i e^{-\delta_i t} \frac{1}{D_i} \sum_{j=0}^{N-2} \left(e^{D_i t_j + 1} - e^{D_i t_j}\right) \frac{1}{\gamma_i + \bar{f}_j}$$

where $\bar{f}_j = \frac{\left(f\left(t_j\right) + f\left(t_j + 1\right)\right)}{2}$ on each subinterval $(t_j, t_j + 1)$, $j = 0, \ldots, N-2$. This is under the simplifying assumption that $f(t)$ is a piece-wise constant function on each subinterval $(t_j, t_j + 1)$. **One can come up with linear (or higher order) $f(t)$ approximations on each subinterval. This will introduce additional parameters, which will be impossible to infer with any certainty given limited amount of data.**

Khanin et al. (2006)

Their Results

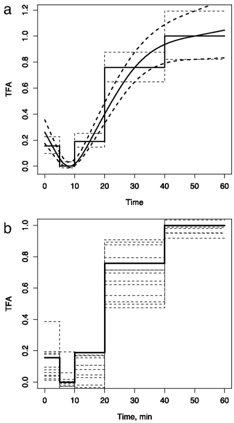


Figure: Fig. 2 from Khanin et al. (2006): Reconstructed activity level of master repressor LexA, following a UV dose of 40 J/m2.

Their Results

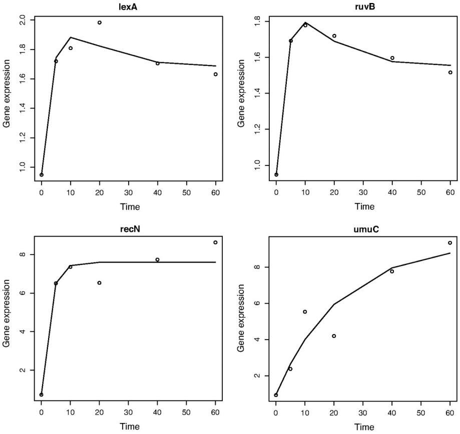


Figure: Fig. 3 from Khanin et al. (2006): Reconstructed profiles for four genes in the LexA SIM.

Repression Model

Pei Gao

- We can use the same model of repression,

$$g_j\left(f\left(t\right)\right) = \frac{1}{\gamma_j + e^{f(t)}}$$

In the case of repression we have to include the transient term,

$$x_j\left(t\right) = \alpha_j e^{-D_j t} + \frac{B_j}{D_j} + S_j \int_0^t e^{-D_j(t-u)} g_j(f\left(u\right))\mathrm{d}u$$

Results for the repressor LexA

Pei Gao

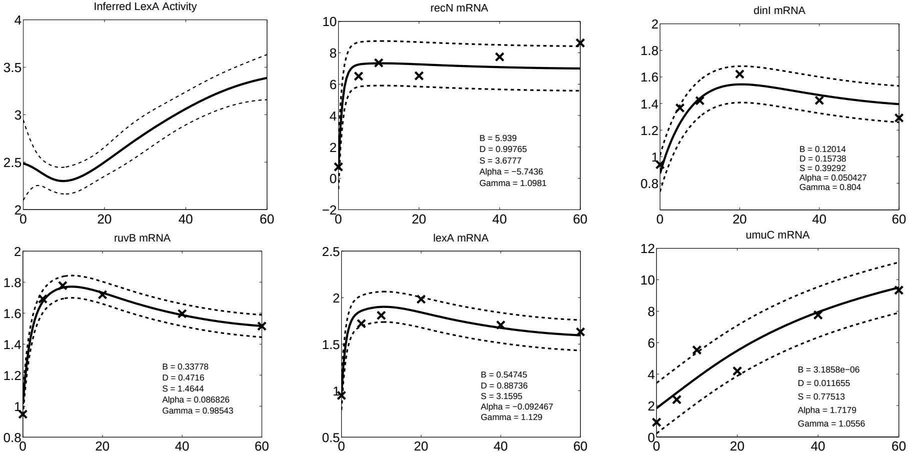


Figure: Our results using an MLP kernel. To apear at ECCB08 Gao et al. (2008).

Use Samples to Represent Posterior

Michalis Titsias

- Sample in Gaussian processes

$$p\left(\mathbf{f}|\mathbf{x}\right) \propto p\left(\mathbf{x}|\mathbf{f}\right) p\left(\mathbf{f}\right)$$

- Likelihood relates GP to data through

$$x_j\left(t\right) = \alpha_j e^{-D_j t} + \frac{B_j}{D_j} + S_j \int_0^t e^{-D_j(t-u)} g_j(f\left(u\right))\mathrm{d}u$$

- We use *control points* for fast sampling.

MCMC for Non Linear Response

The Metropolis-Hastings algorithm

- Initialize $\mathbf{f}^{(0)}$
- Form a Markov chain. Use a proposal distribution $Q(\mathbf{f}^{(t+1)}|\mathbf{f}^{(t)})$ and accept with the M-H step

$$\min\left(1, \frac{p(\mathbf{x}|\mathbf{f}^{(t+1)})p(\mathbf{f}^{(t+1)})}{p(\mathbf{x}|\mathbf{f}^{(t)})p(\mathbf{f}^{(t)})}\frac{Q(\mathbf{f}^{(t)}|\mathbf{f}^{(t+1)})}{Q(\mathbf{f}^{(t+1)}|\mathbf{f}^{(t)})}\right)$$

- $\mathbf{f}$ can be very *high dimensional* (hundreds of points)
- How do we choose the proposal $Q(\mathbf{f}^{(t+1)}|\mathbf{f}^{(t)})$?
  - Can we use the GP prior $p(\mathbf{f})$ as the proposal?

Sampling using control points

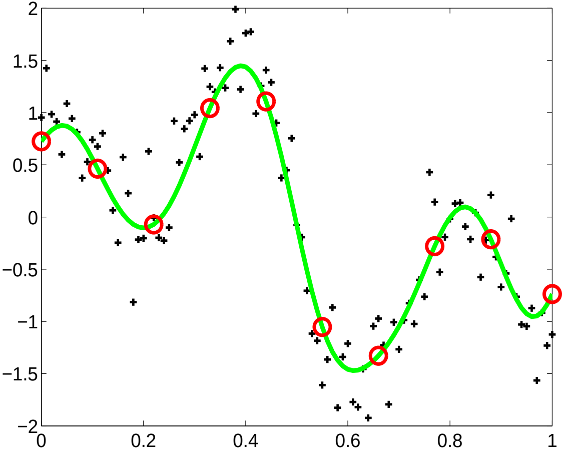
- Separate the points in $\mathbf{f}$ into two groups:
  - few control points $\mathbf{f}_c$
  - and the large majority of the remaining points $\mathbf{f}_\rho = \mathbf{f} \setminus \mathbf{f}_c$
- Sample the control points $\mathbf{f}_c$ using a proposal $q\left(\mathbf{f}_c^{(t+1)}|\mathbf{f}_c^{(t)}\right)$
- Sample the remaining points $\mathbf{f}_\rho$ using the conditional GP prior $p\left(\mathbf{f}_\rho^{(t+1)}|\mathbf{f}_c^{(t+1)}\right)$
- The whole proposal is

$$Q\left(\mathbf{f}^{(t+1)}|\mathbf{f}^{(t)}\right) = p\left(\mathbf{f}_\rho^{(t+1)}|\mathbf{f}_c^{(t+1)}\right) q\left(\mathbf{f}_c^{(t+1)}|\mathbf{f}_c^{(t)}\right)$$

- Its like sampling from the prior $p(\mathbf{f})$ but imposing random walk behaviour through the control points

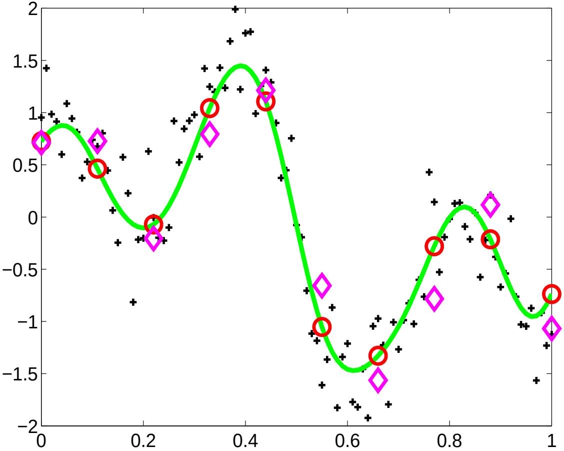
Sampling using control points: Regression-Examples

Sample 121 points using 10 control points



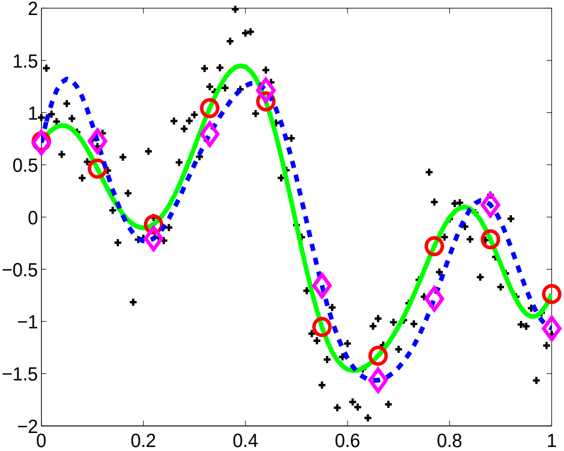
Sampling using control points: Regression-Examples

Sample 121 points using 10 control points



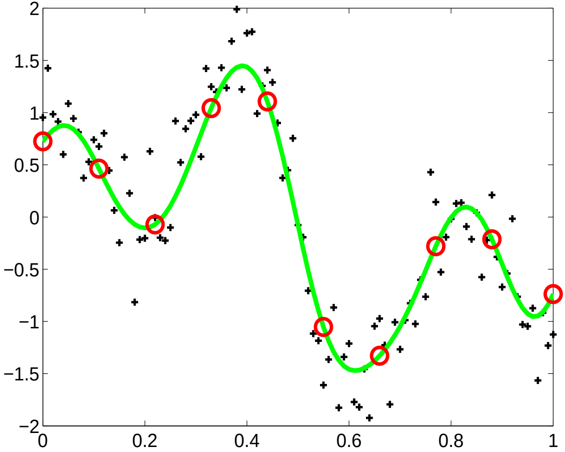
Sampling using control points: Regression-Examples

Sample 121 points using 10 control points



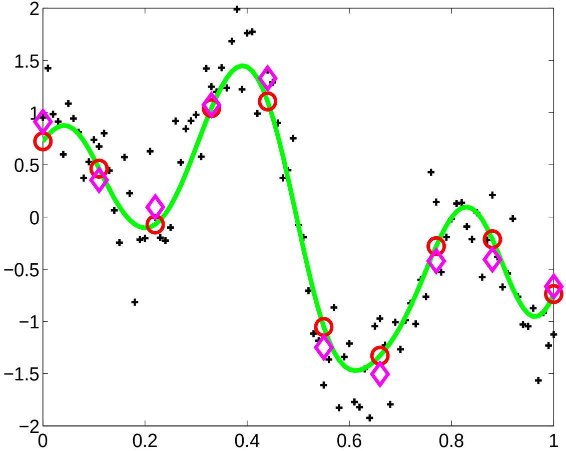
Sampling using control points: Regression-Examples

Sample 121 points using 10 control points



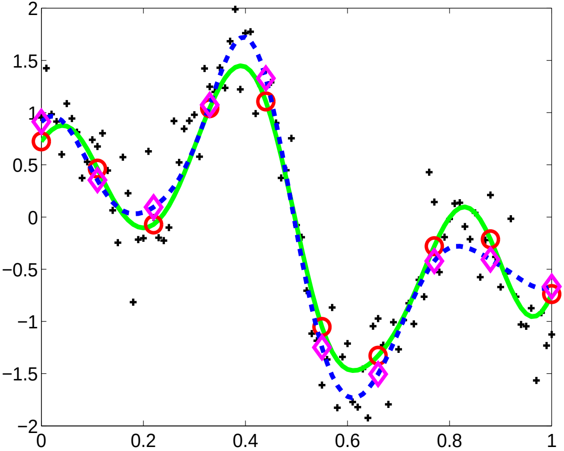
Sampling using control points: Regression-Examples

Sample 121 points using 10 control points



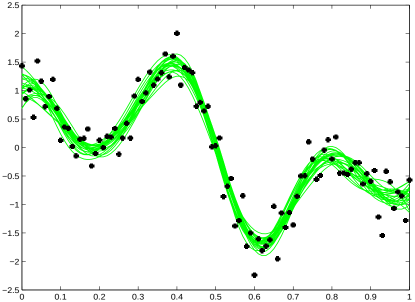
Sampling using control points: Regression-Examples

Sample 121 points using 10 control points



Sampling using control points

Few samples drawn during MCMC



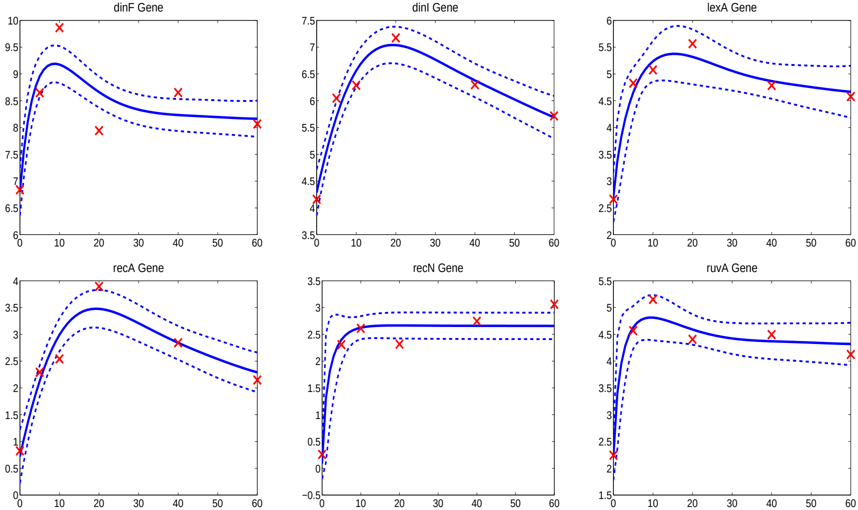
Results on SOS System

- Again consider the Michaelis-Menten kinetic equation

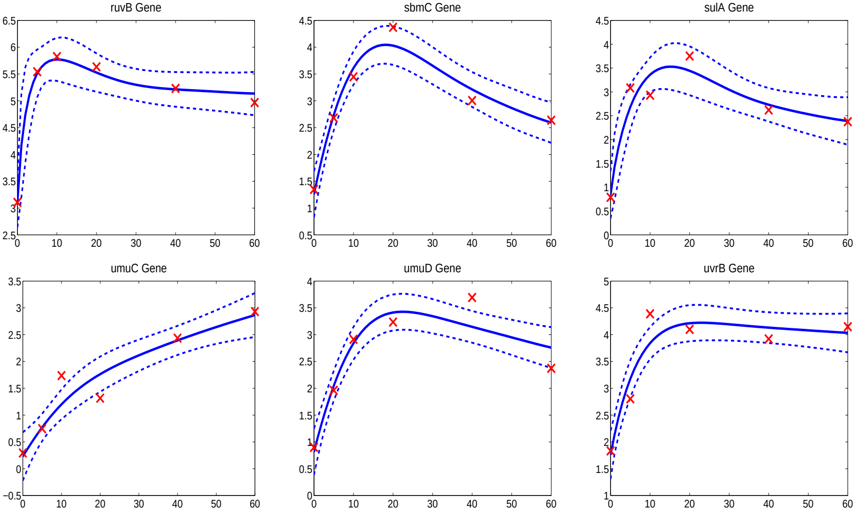
$$\frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = B_j + S_j \frac{1}{\exp(f(t)) + \gamma_j} - D_j x_j(t)$$

- We have 14 genes (5 kinetic parameters each)
- Gene expressions are available for $T = 6$ time slots
- TF ($\mathbf{f}$) is discretized using 121 points
- MCMC details:
  - 6 control points are used (placed in a equally spaced grid)
  - Running time was 5 hours for 2 million sampling iterations plus burn in
  - Acceptance rate for $\mathbf{f}$ after burn in was between $15\% - 25\%$

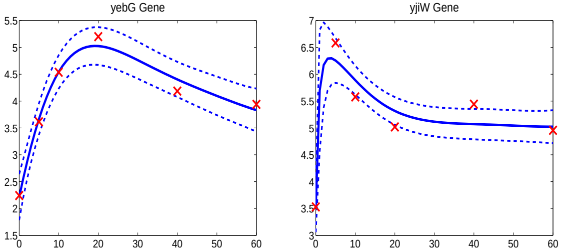
Results in E.coli data: Predicted gene expressions



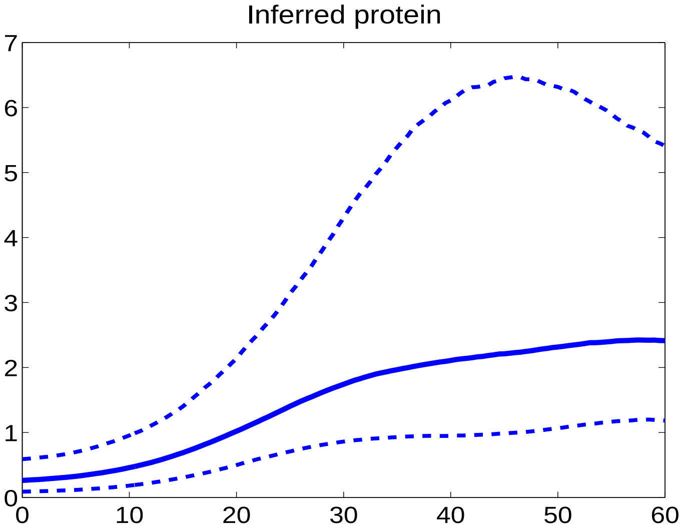
Results in E.coli data: Predicted gene expressions



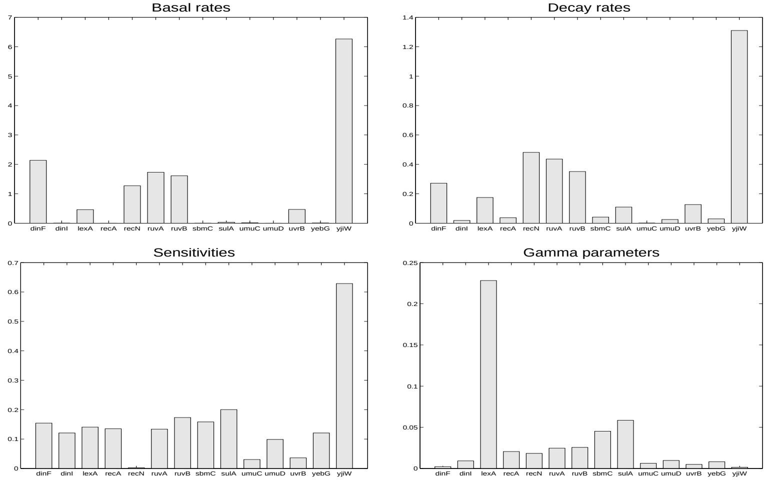
Results in E.coli data: Predicted gene expressions



Results in E.coli data: Protein concentration



Results in E.coli data: Kinetic parameters



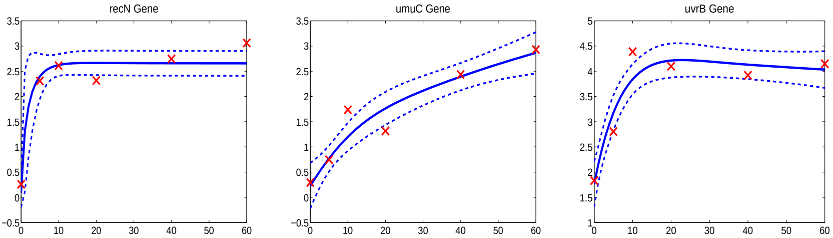
Basal rates

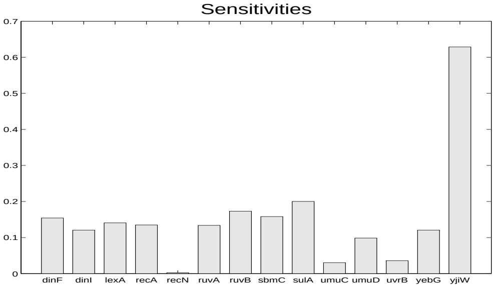
Decay rates

Sensitivities

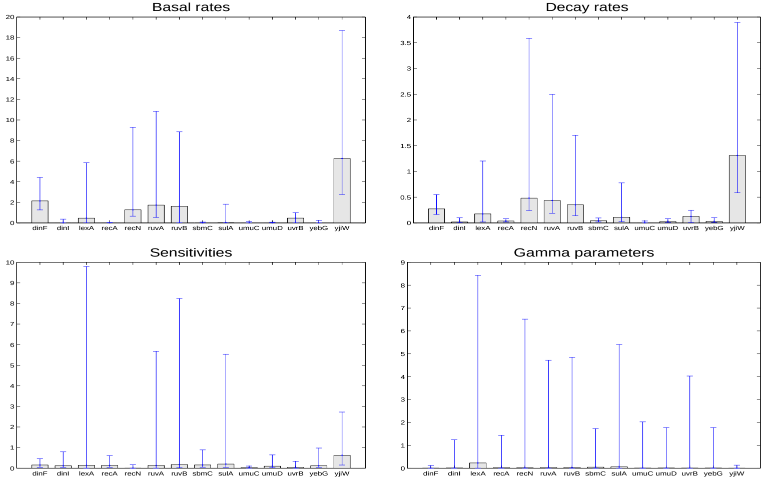
Gamma parameters

Results in E.coli data: Genes with low sensitivity value





Results in E.coli data: Confidence intervals for the kinetic parameters



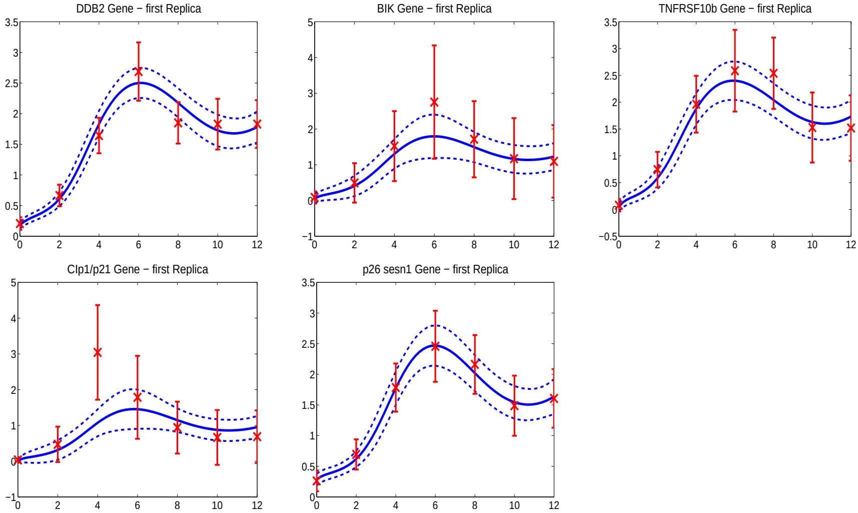
p53 System Again

- One transcription factor (p53) that acts as an activator. We consider the Michaelis-Menten kinetic equation

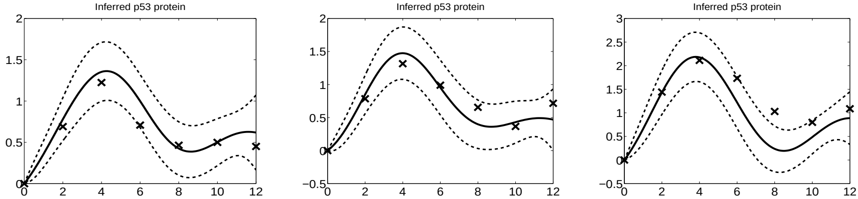
$$\frac{\mathrm{d}x_j(t)}{dt} = B_j + S_j \frac{\exp(f(t))}{\exp(f(t)) + \gamma_j} - D_j x_j(t)$$

- We have 5 genes
- Gene expressions are available for $T = 7$ times and there are 3 replicas of the time series data
- TF ($\mathbf{f}$) is discretized using 121 points
- MCMC details:
  - 7 control points are used (placed in a equally spaced grid)
  - Running time 4/5 hours for 2 million sampling iterations plus burn in
  - Acceptance rate for **f** after burn in was between 15% – 25%

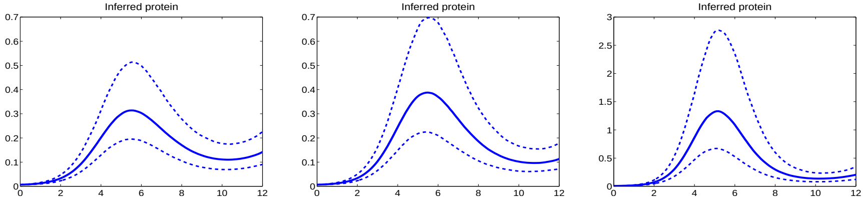
Data used by Barenco et al. (2006): Predicted gene expressions for the 1st replica



Data used by Barenco et al. (2006): Protein concentrations

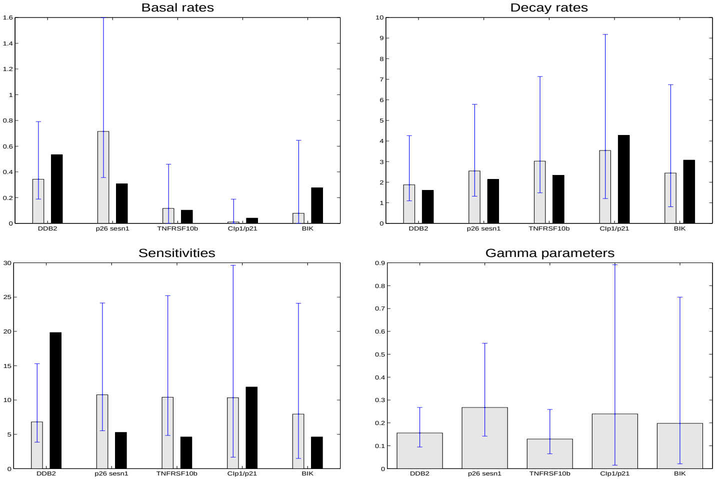


Linear model (Barenco et al. predictions are shown as crosses)



Nonlinear (Michaelis-Menten kinetic equation)

p53 Data Kinetic parameters



Our results (grey) compared with Barenco et al. (2006) (black). Note that Barenco et al. use a linear model

Outline

1. Introduction
2. Gaussian Process Inference for Linear Activation
3. Non-linear Response Models
4. Cascaded Differential Equations
5. Discussion and Future Work
6. Acknowledgements

Cascaded Differential Equations

Antti Honkela

- Transcription factor protein also has governing mRNA.
- This mRNA can be measured.
- In signalling systems this measurement can be misleading because it is activated (phosphorylated) transcription factor that counts.
- In development phosphorylation plays less of a role.

Drosophila *Mesoderm* Development

Data from Furlong Lab in EMBL Heidelberg.

- Describe mesoderm development.

Cascaded Differential Equations

Antti Honkela

We take the production rate of active transcription factor to be given by

$$\frac{\mathrm{d}f\left(t\right)}{\mathrm{d}t} = \sigma y\left(t\right) - \delta f\left(t\right)$$
$$\frac{\mathrm{d}x_j\left(t\right)}{\mathrm{d}t} = B_j + S_j f\left(t\right) - D_j x_j\left(t\right)$$

The solution for $f(t)$, setting transient terms to zero, is

$$f(t) = \sigma \exp\left(-\delta t\right) \int_0^t y(u) \exp\left(\delta u\right) \mathrm{d}u \ .$$

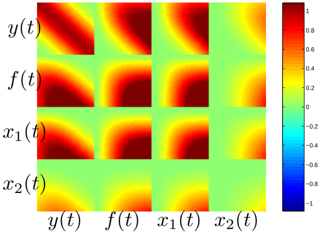
Covariance for Translation/Transcription Model

RBF covariance function for $y(t)$

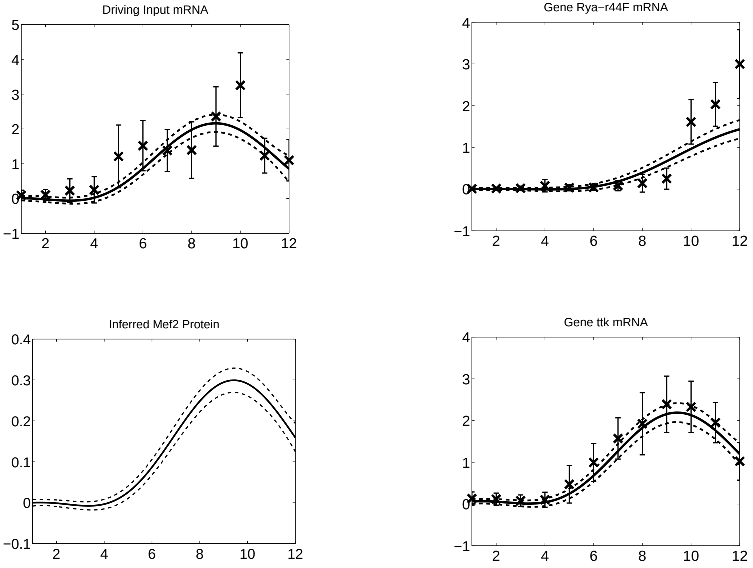
$$
\begin{aligned}
f(t) &= \sigma \exp(-\delta t) \int_0^t y(u) \exp(\delta u)\,\mathrm{d}u \\
x_i(t) &= \frac{B_i}{D_i} + S_i \exp(-D_i t) \int_0^t f(u) \exp(D_i u)\,\mathrm{d}u.
\end{aligned}
$$

- Joint distribution for $x_1(t)$, $x_2(t)$, $f(t)$ and $y(t)$.
- Here:

δ	D_1	S_1	D_2	S_2
0.1	5	5	0.5	0.5



Results for Mef2 using the Cascade model



Outline

1. Introduction
2. Gaussian Process Inference for Linear Activation
3. Non-linear Response Models
4. Cascaded Differential Equations
5. Discussion and Future Work
6. Acknowledgements

Discussion and Future Work

- Integration of probabilistic inference with mechanistic models.
- These results are small simple systems.
- Ongoing work:
  - Scaling up to larger systems
  - Applications to other types of system, *e.g.* non-steady-state metabolomics, spatial systems etc.
  - Improved approximations.
  - Stochastic differential equations

Outline

1. Introduction

2. Gaussian Process Inference for Linear Activation

3. Non-linear Response Models

4. Cascaded Differential Equations

5. Discussion and Future Work

6. Acknowledgements

Acknowledgements

- Investigators: Neil Lawrence and Magnus Rattray
- Researchers: Peo Gao, Antti Honkela, Michalis Titsias and Jennifer Withers
- Charles Girardot and Eileen Furlong of EMBL in Heidelberg (mesoderm development in *D. Melanogaster*).
- Martino Barenco and Mike Hubank at the Institute of Child Health in UCL (p53 pathway).
- Raya Khanin and Ernst Wit of the University of Glasgow and the University of Lancaster (*E. coli* repressor system).

Funded by the BBSRC award "Improved Processing of microarray data using probabilistic models" and EPSRC award "Gaussian Processes for Systems Identification with applications in Systems Biology"

References I

M. Barenco, D. Tomescu, D. Brewer, R. Callard, J. Stark, and M. Hubank. Ranked prediction of p53 targets using hidden variable dynamic modeling. *Genome Biology*, 7(3):R25, 2006. [PDF].

R. T. Cirz, J. K. Chin, D. R. Andes, V. de CrÃ©cy-Lagard, W. A. Craig, and F. E. Romesberg. Inhibition of mutation and combating the evolution of antibiotic resistance. *PLoS Biology*, 3(6), 2005.

J. Courcelle, A. Khodursky, B. Peter, P. O. Brown, , and P. C. Hanawalt. Comparative gene expression profiles following UV exposure in wild-type and SOS-deficient *Escherichia coli*. *Genetics*, 158:41–64, 2001.

P. Gao, A. Honkela, M. Rattray, and N. D. Lawrence. Gaussian process modelling of latent chemical species: Applications to inferring transcription factor activities. *Bioinformatics*, 2008. To appear at ECCB '08.

R. Khanin, V. Viciotti, and E. Wit. Reconstructing repressor protein levels from expression of gene targets in *E. Coli*. *Proc. Natl. Acad. Sci. USA*, 103(49):18592–18596, 2006. [PDF]. [DOI].

A. M. Lee, C. T. Ross, B.-B. Zeng, , and S. F. Singleton. A molecular target for suppression of the evolution of antibiotic resistance: Inhibition of the *Escherichia coli* RecA protein by N6-(1-Naphthyl)-ADP. *J. Med. Chem.*, 48(17), 2005.

S. Rogers and M. Girolami. Model based identification of transcription factor regulatory activity via Markov chain Monte Carlo. Presentation at MASAMB '06, 2006.

C. K. I. Williams. Computing with infinite networks. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, Cambridge, MA, 1997. MIT Press.