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

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

University of Warwick
LICSB Workshop

16th June 2008

Outline

1. Introduction
2. Gaussian Process Inference for Linear Activation
3. Cascaded Differential Equations
4. Discussion and Future Work
5. Acknowledgements

Outline

Talk Outline

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

Talk Outline

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

Talk Outline

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

Talk Outline

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

Talk Outline

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

Talk Outline

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

ODE Model of Activation

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

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

- $x_j(t)$ – 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)

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

- $x_j(t)$ – 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?

Outline

Zero mean Gaussian distribution

- A multi-variate Gaussian distribution is defined by a mean and a covariance matrix.

$$N(\mathbf{f}|\mu, \mathbf{K}) = \frac{1}{(2\pi)^{\frac{N}{2}} |\mathbf{K}|^{\frac{1}{2}}} \exp\left(-\frac{(\mathbf{f}-\mu)^{\mathrm{T}} \mathbf{K}^{-1} (\mathbf{f}-\mu)}{2}\right).$$

- We will consider the special case where the mean is zero,

$$N(\mathbf{f}|\mathbf{0}, \mathbf{K}) = \frac{1}{(2\pi)^{\frac{N}{2}} |\mathbf{K}|^{\frac{1}{2}}} \exp\left(-\frac{\mathbf{f}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{f}}{2}\right).$$

Sampling a Function

Multi-variate Gaussians

- We will consider a Gaussian with a particular structure of covariance matrix.
- Generate a single sample from this 25 dimensional Gaussian distribution, $\mathbf{f} = [f_1, f_2 \ldots f_{25}]$.
- We will plot these points against their index.

demGPSample

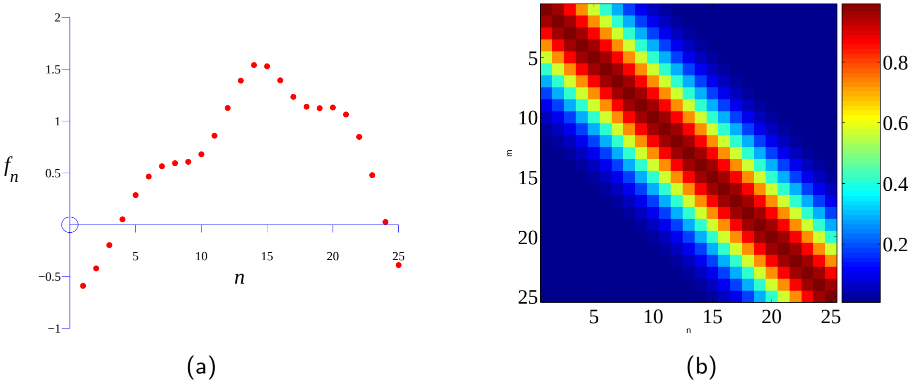


Figure: (a) 25 instantiations of a function, f_n, (b) colormap of covariance matrix.

Covariance Function

The covariance matrix

- Covariance matrix shows correlation between points f_m and f_n if n is near to m.
- Less correlation if n is distant from m.
- Our ordering of points means that the *function appears smooth*.
- Let's focus on the joint distribution of two points form the 25.

Covariance Function

The covariance matrix

- Covariance matrix shows correlation between points f_m and f_n if n is near to m.
- Less correlation if n is distant from m.
- Our ordering of points means that the *function appears smooth*.
- Let's focus on the joint distribution of two points form the 25.

Covariance Function

The covariance matrix

- Covariance matrix shows correlation between points f_m and f_n if n is near to m.
- Less correlation if n is distant from m.
- Our ordering of points means that the *function appears smooth*.
- Let's focus on the joint distribution of two points form the 25.

Covariance Function

The covariance matrix

- Covariance matrix shows correlation between points f_m and f_n if n is near to m.
- Less correlation if n is distant from m.
- Our ordering of points means that the *function appears smooth*.
- Let's focus on the joint distribution of two points form the 25.

Prediction of f_2 from f_1

`demGPCov2D([1 2])`

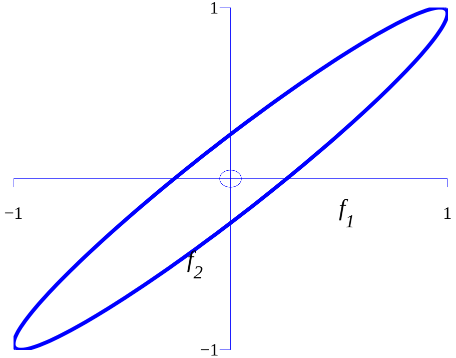


Figure: Covariance for $\begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$ is $\mathbf{K}_{12} = \begin{bmatrix} 1 & 0.966 \\ 0.966 & 1 \end{bmatrix}$.

```
demGPCov2D([1 2])
```

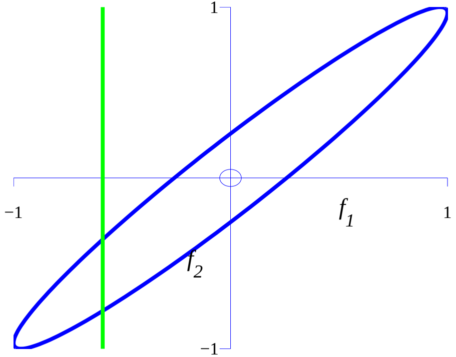


Figure: Covariance for $\begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$ is $\mathbf{K}_{12} = \begin{bmatrix} 1 & 0.966 \\ 0.966 & 1 \end{bmatrix}$.

Prediction of f_2 from f_1

```
demGPCov2D([1 2])
```

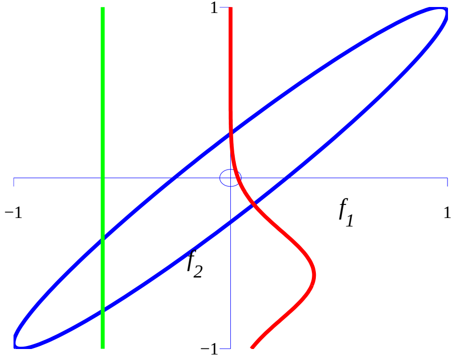


Figure: Covariance for $\begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$ is $\mathbf{K}_{12} = \begin{bmatrix} 1 & 0.966 \\ 0.966 & 1 \end{bmatrix}$.

Prediction of f_5 from f_1

```
demGPCov2D([1 5])
```

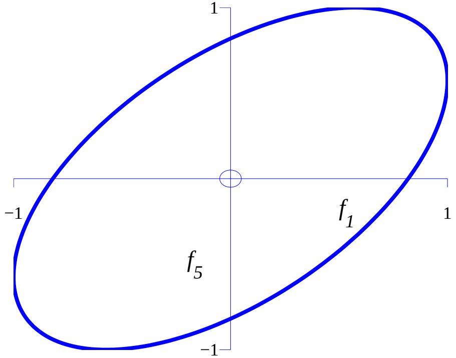


Figure: Covariance for $\begin{bmatrix} f_1 \\ f_5 \end{bmatrix}$ is $\mathbf{K}_{15} = \begin{bmatrix} 1 & 0.574 \\ 0.574 & 1 \end{bmatrix}$.

Prediction of f_5 from f_1

```
demGPCov2D([1 5])
```

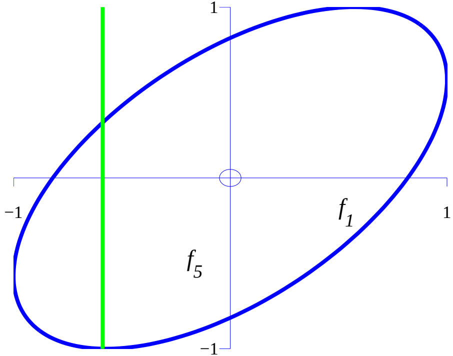


Figure: Covariance for $\begin{bmatrix} f_1 \\ f_5 \end{bmatrix}$ is $\mathbf{K}_{15} = \begin{bmatrix} 1 & 0.574 \\ 0.574 & 1 \end{bmatrix}$.

```
demGPCov2D([1 5])
```

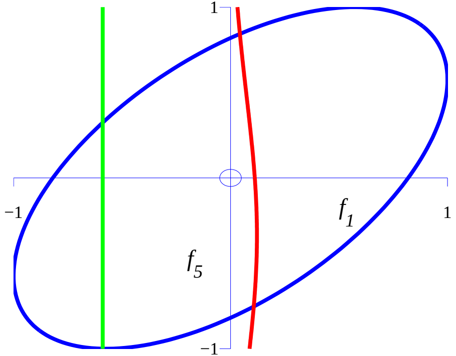
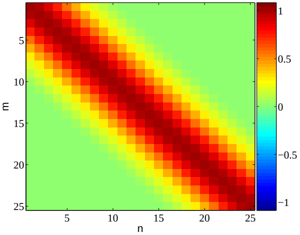


Figure: Covariance for $\begin{bmatrix} f_1 \\ f_5 \end{bmatrix}$ is $\mathbf{K}_{15} = \begin{bmatrix} 1 & 0.574 \\ 0.574 & 1 \end{bmatrix}$.

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.



Covariance Samples

`demCovFuncSample`

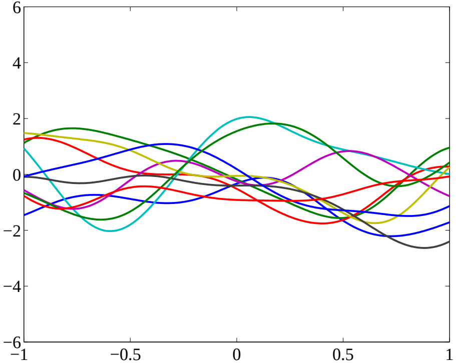


Figure: RBF kernel with $l = 10^{-\frac{1}{2}}$, $\alpha = 1$

Covariance Samples

`demCovFuncSample`

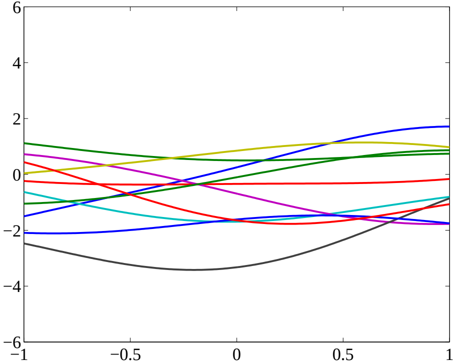


Figure: RBF kernel with $l = 1$, $\alpha = 1$

`demCovFuncSample`

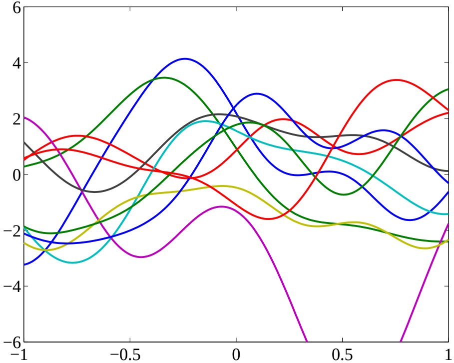


Figure: RBF kernel with $l = 0.3$, $\alpha = 4$

Gaussian Process Regression

demRegression

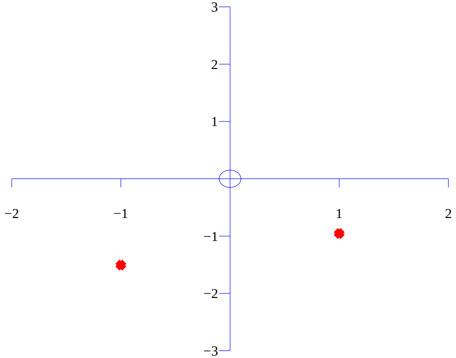


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

demRegression

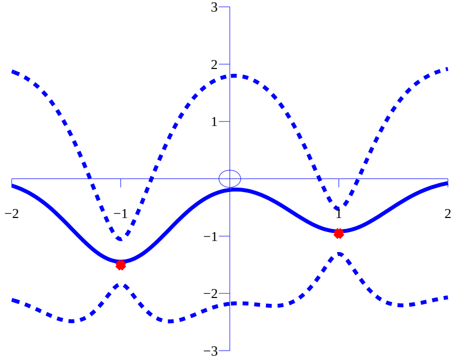


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

`demRegression`

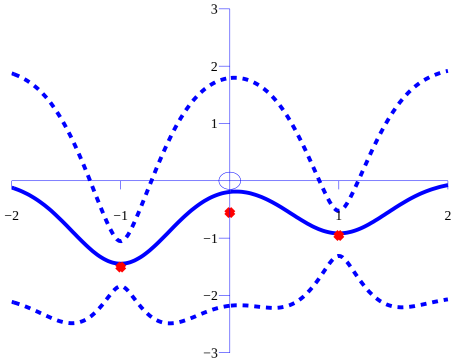


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

demRegression

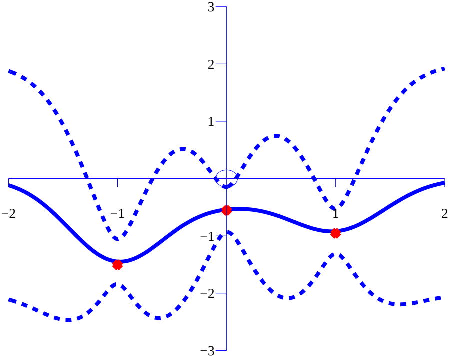


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

`demRegression`

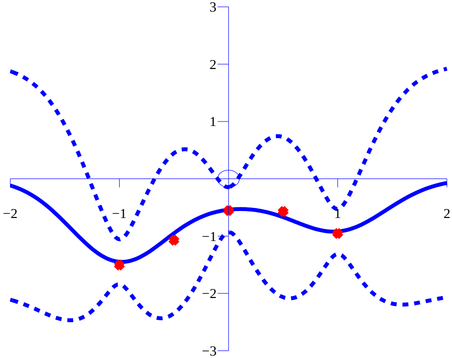


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

`demRegression`

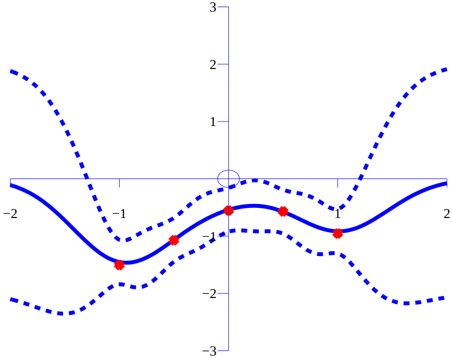


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

demRegression

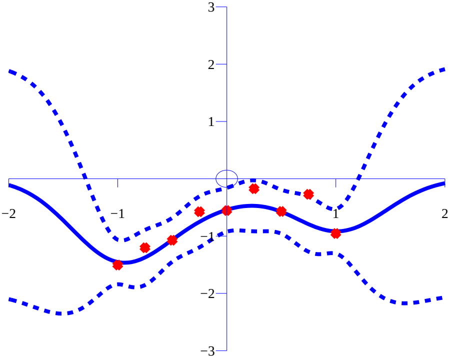


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

demRegression

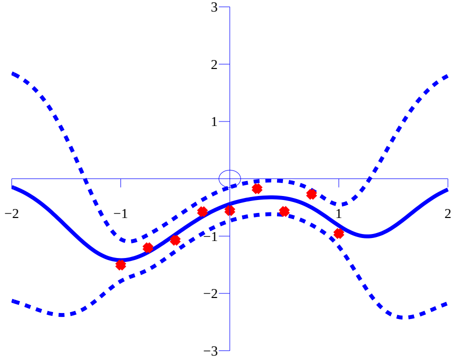
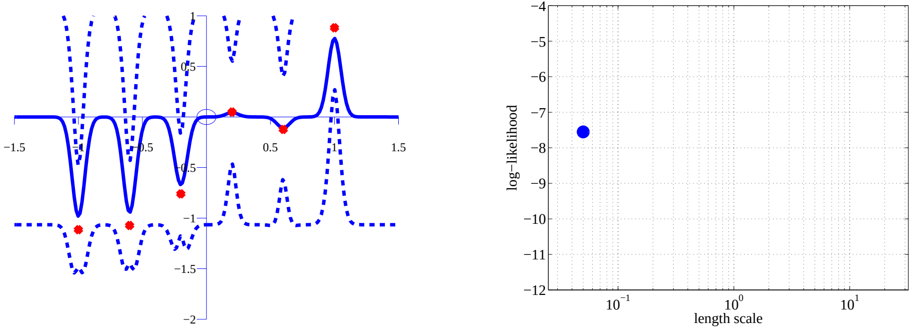


Figure: Examples include WiFi localization, C14 callibration curve.

Learning Kernel Parameters

Can we determine length scales and noise levels from the data?

`demOptimiseKern`

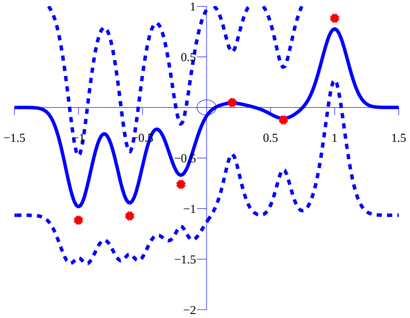


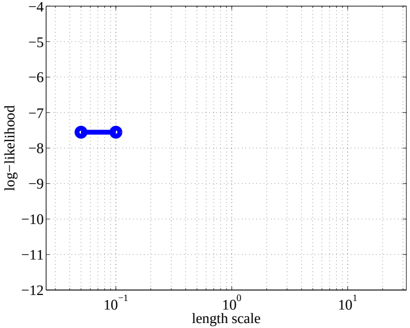
$$\log N\left(\mathbf{f}|\mathbf{0},\mathbf{K}\right) = -\frac{N}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}| - \frac{\mathbf{f}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{f}}{2}$$

Learning Kernel Parameters

Can we determine length scales and noise levels from the data?

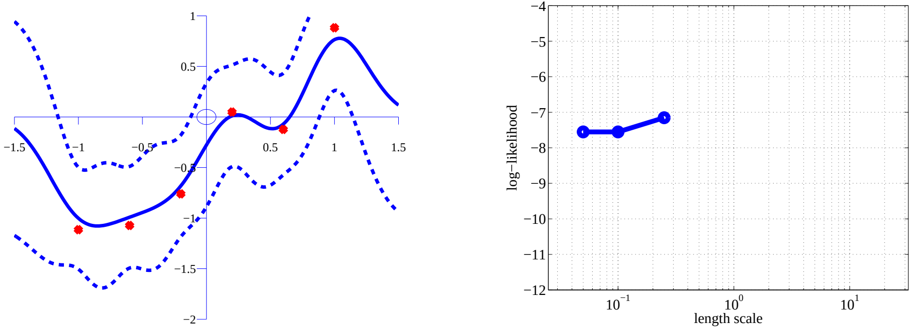
`demOptimiseKern`





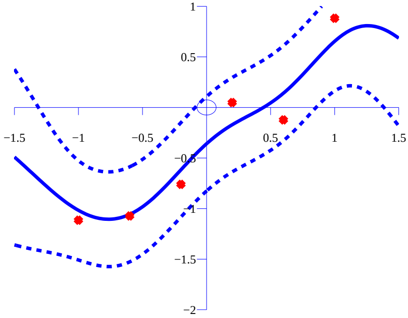
$$\log N(\mathbf{f}|\mathbf{0},\mathbf{K}) = -\frac{N}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}| - \frac{\mathbf{f}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{f}}{2}$$

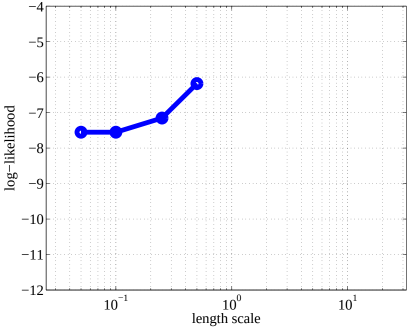
demOptimiseKern



$$\log N(\mathbf{f}|\mathbf{0},\mathbf{K}) = -\frac{N}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}| - \frac{\mathbf{f}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{f}}{2}$$

```
demOptimiseKern
```



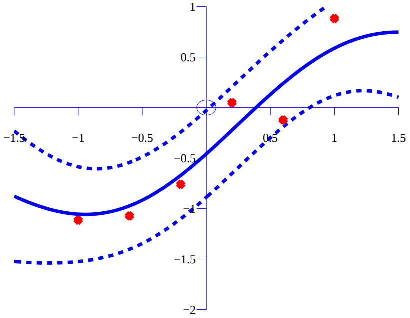


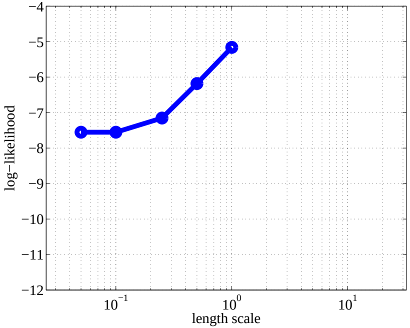
$$\log N(\mathbf{f}|\mathbf{0}, \mathbf{K}) = -\frac{N}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}| - \frac{\mathbf{f}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{f}}{2}$$

Learning Kernel Parameters

Can we determine length scales and noise levels from the data?

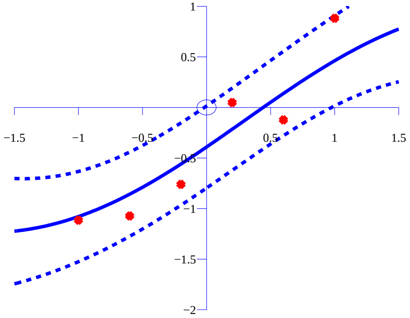
`demOptimiseKern`

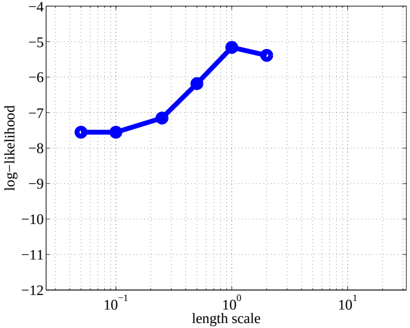




$$\log N(\mathbf{f}|\mathbf{0}, \mathbf{K}) = -\frac{N}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}| - \frac{\mathbf{f}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{f}}{2}$$

`demOptimiseKern`



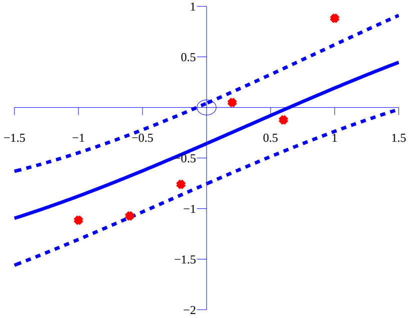


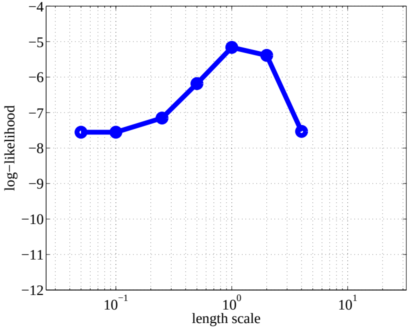
$$\log N(\mathbf{f}|\mathbf{0}, \mathbf{K}) = -\frac{N}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}| - \frac{\mathbf{f}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{f}}{2}$$

Learning Kernel Parameters

Can we determine length scales and noise levels from the data?

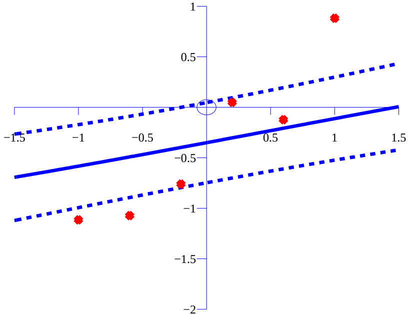
`demOptimiseKern`

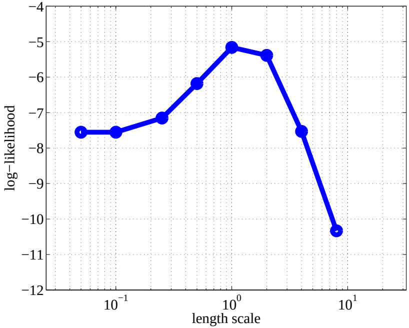




$$\log N(\mathbf{f}|\mathbf{0},\mathbf{K}) = -\frac{N}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}| - \frac{\mathbf{f}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{f}}{2}$$

`demOptimiseKern`



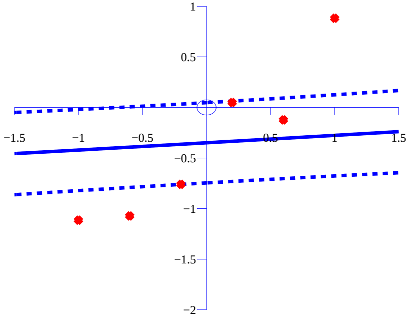


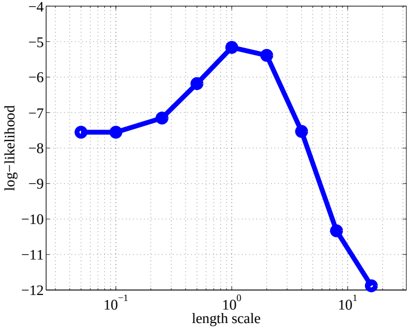
$$\log N(\mathbf{f}|\mathbf{0},\mathbf{K}) = -\frac{N}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}| - \frac{\mathbf{f}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{f}}{2}$$

Learning Kernel Parameters

Can we determine length scales and noise levels from the data?

`demOptimiseKern`





$$\log N(\mathbf{f}|\mathbf{0}, \mathbf{K}) = -\frac{N}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}| - \frac{\mathbf{f}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{f}}{2}$$

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 $f(t)$.

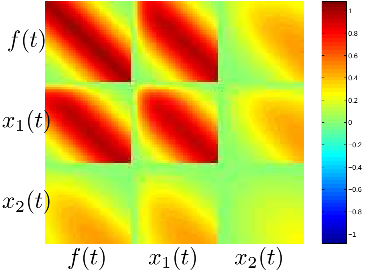
If $f(t)$ is a zero mean Gaussian process then $x_i(t)$ is also a Gaussian process with mean $\frac{B_i}{D_i}$.

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`

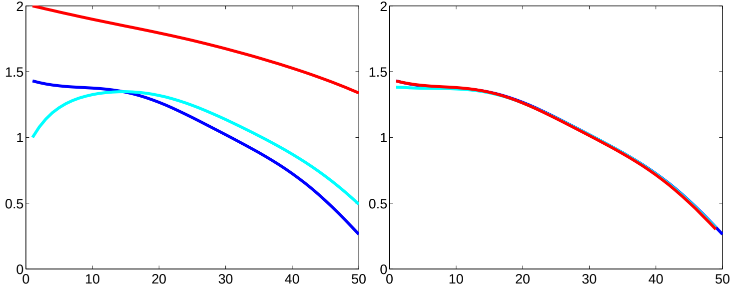


Figure: *Left*: joint samples from the transcription covariance, *blue*: $f(t)$, *cyan*: $x_1(t)$ and *red*: $x_2(t)$. *Right*: numerical solution for $f(t)$ of the differential equation from $x_1(t)$ and $x_2(t)$ (blue and cyan). True $f(t)$ included for comparison.

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

`gpsimTest`

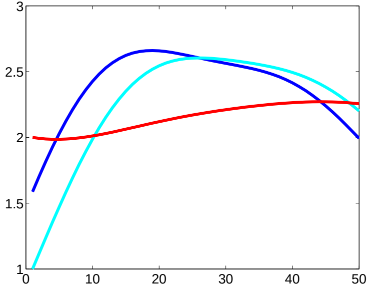


Figure: *Left*: joint samples from the transcription covariance, *blue*: $f(t)$, *cyan*: $x_1(t)$ and *red*: $x_2(t)$. *Right*: numerical solution for $f(t)$ of the differential equation from $x_1(t)$ and $x_2(t)$ (blue and cyan). True $f(t)$ included for comparison.

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

`gpsimTest`

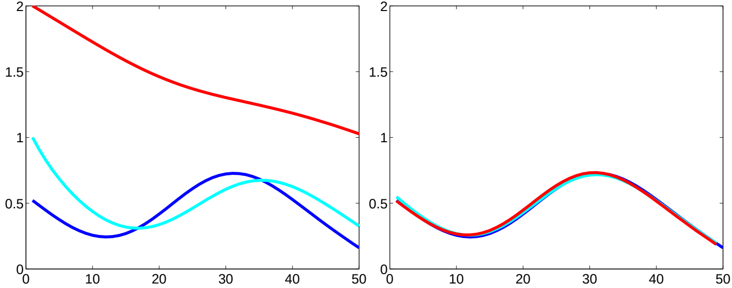
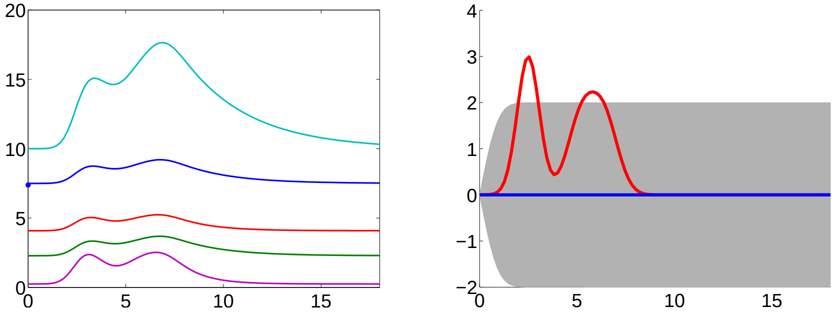
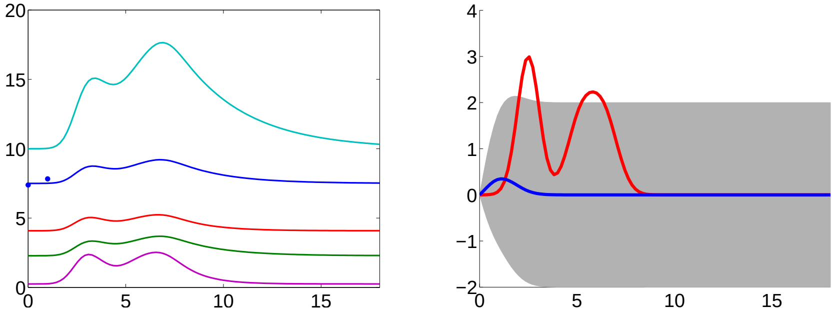


Figure: *Left*: joint samples from the transcription covariance, *blue*: $f(t)$, *cyan*: $x_1(t)$ and *red*: $x_2(t)$. *Right*: numerical solution for $f(t)$ of the differential equation from $x_1(t)$ and $x_2(t)$ (blue and cyan). True $f(t)$ included for comparison.

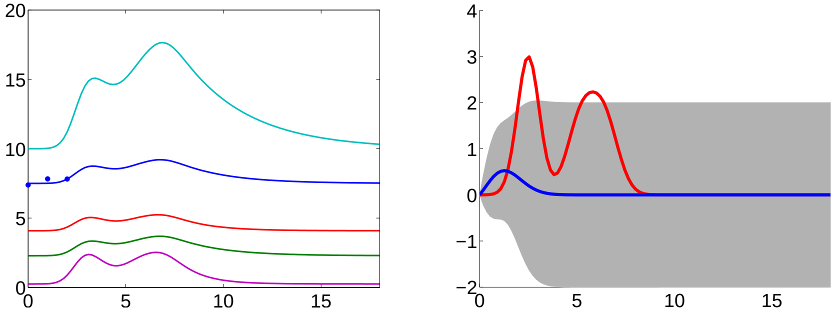
Artificial Example: Inferring $f(t)$



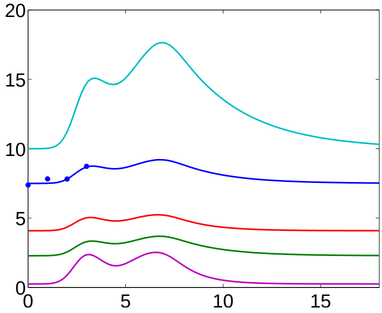
Artificial Example: Inferring $f(t)$

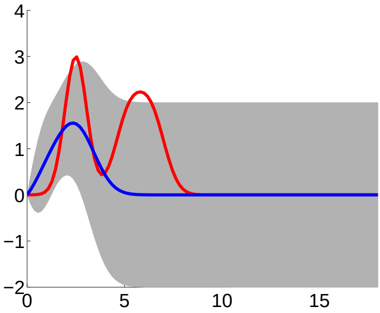


Artificial Example: Inferring $f(t)$

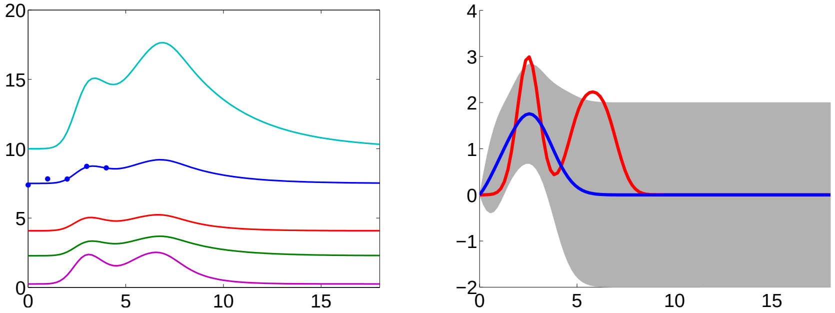


Artificial Example: Inferring $f(t)$

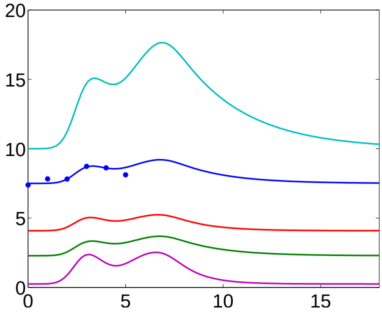


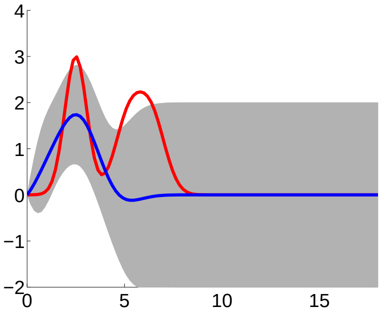


Artificial Example: Inferring $f(t)$

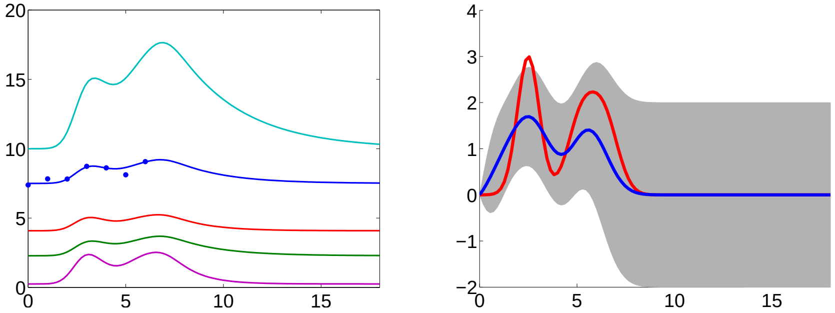


Artificial Example: Inferring $f(t)$

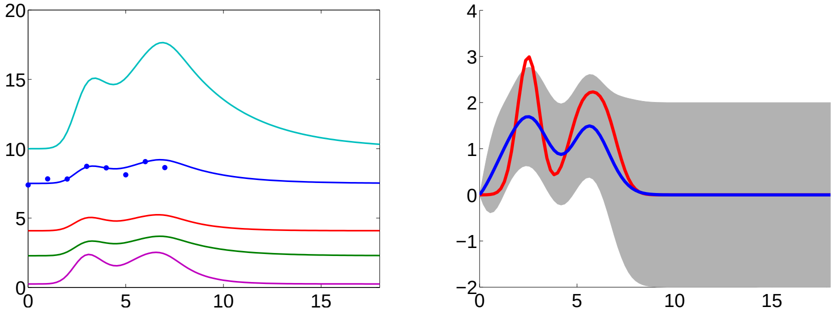




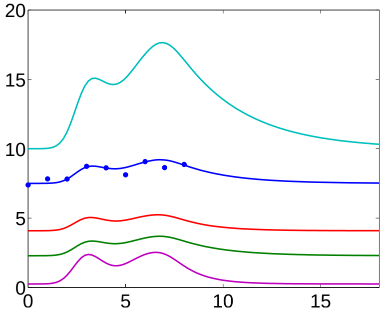
Artificial Example: Inferring $f(t)$

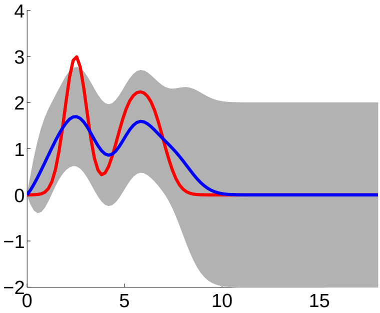


Artificial Example: Inferring $f(t)$

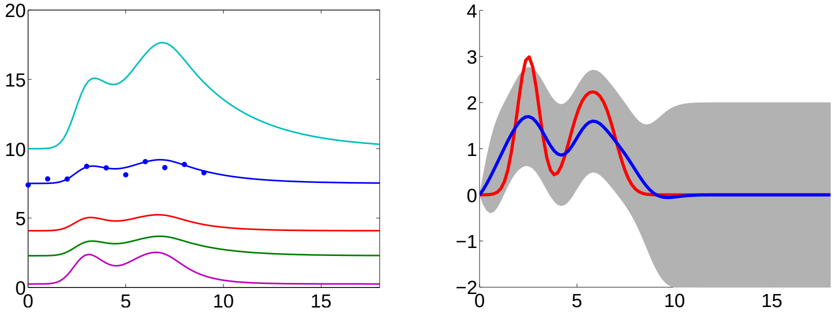


Artificial Example: Inferring $f(t)$

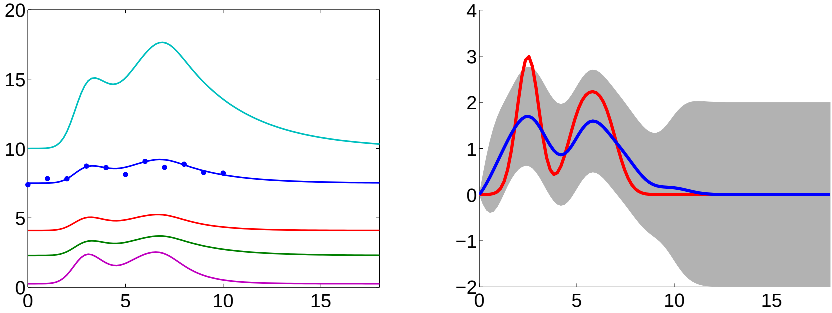




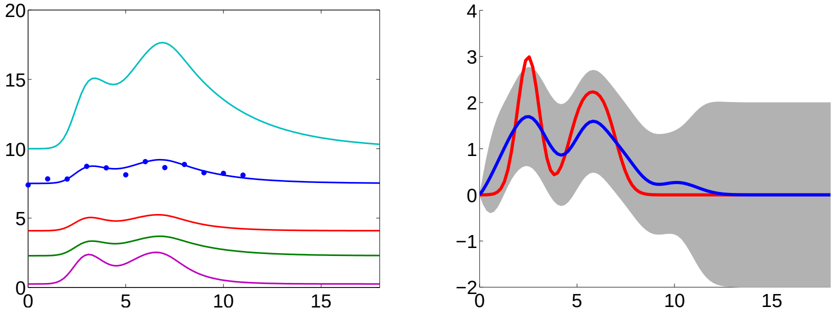
Artificial Example: Inferring $f(t)$



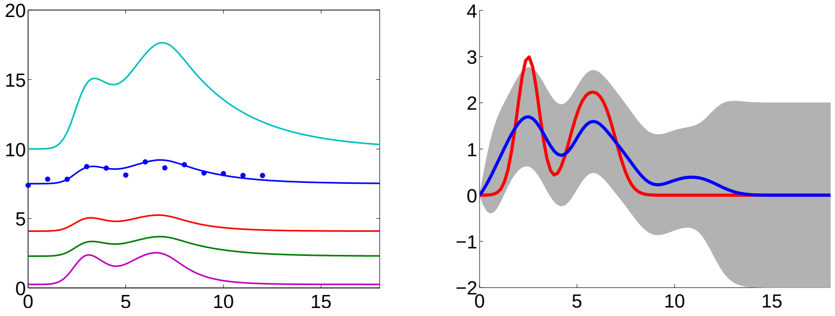
Artificial Example: Inferring $f(t)$



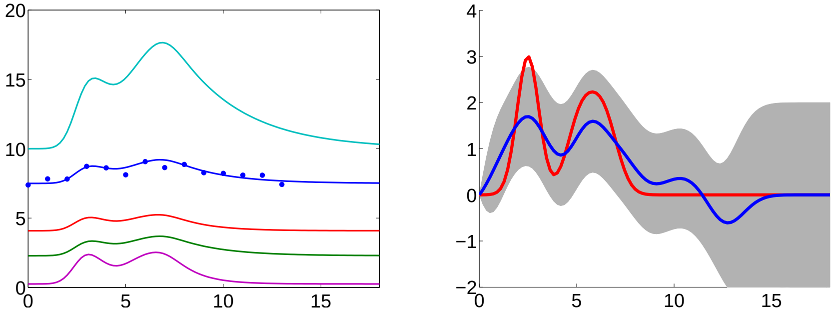
Artificial Example: Inferring $f(t)$



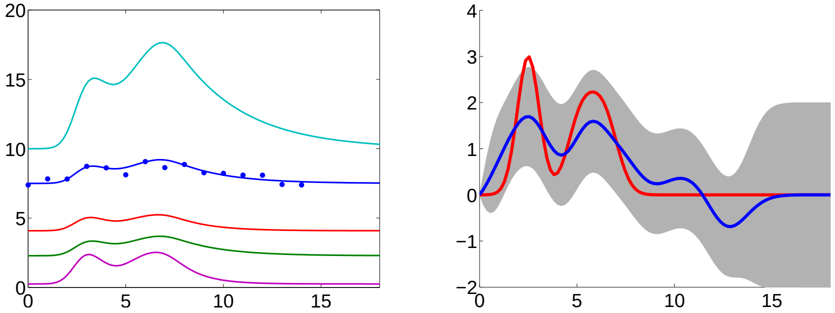
Artificial Example: Inferring $f(t)$



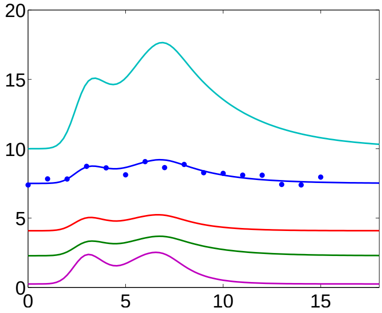
Artificial Example: Inferring $f(t)$

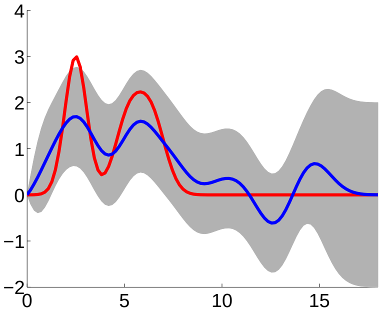


Artificial Example: Inferring $f(t)$

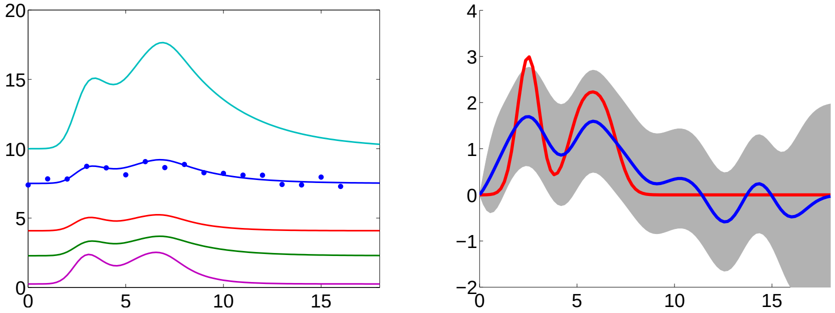


Artificial Example: Inferring $f(t)$

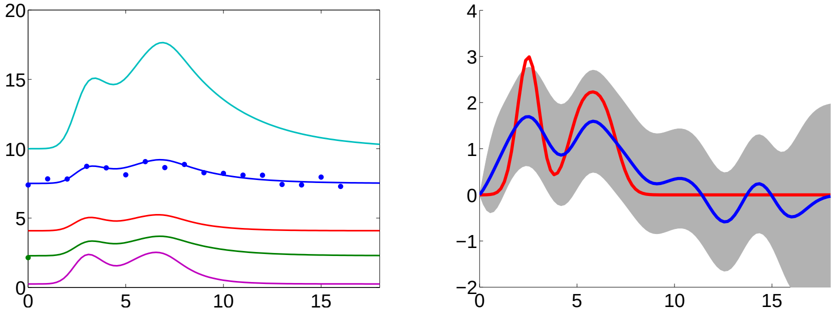




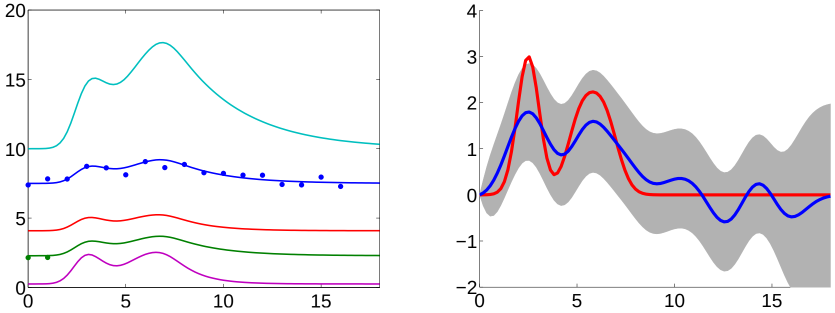
Artificial Example: Inferring $f(t)$



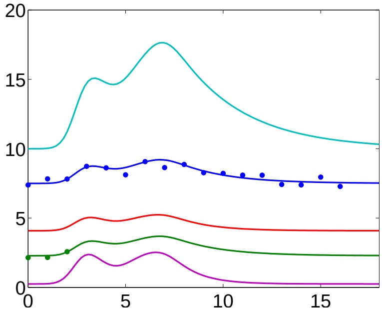
Artificial Example: Inferring $f(t)$

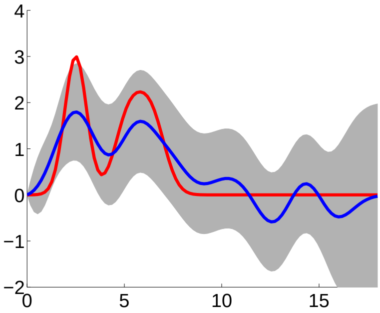


Artificial Example: Inferring $f(t)$

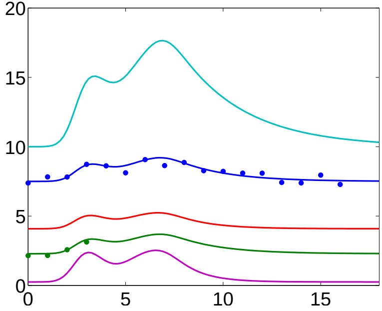


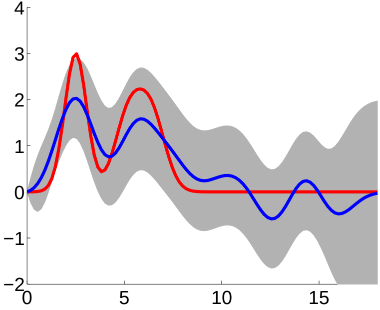
Artificial Example: Inferring $f(t)$



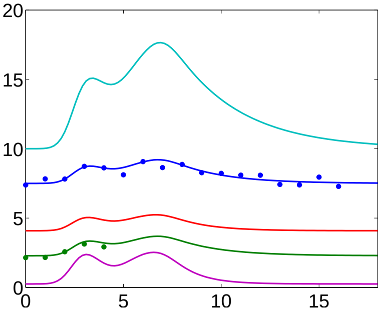


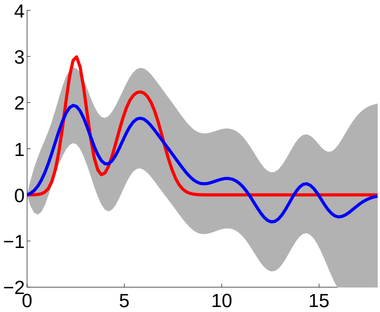
Artificial Example: Inferring $f(t)$



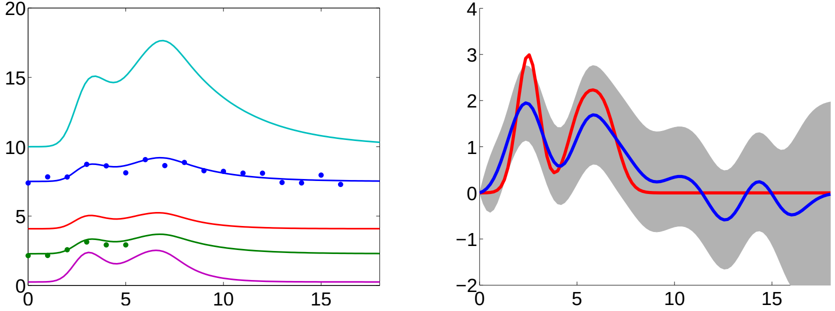


Artificial Example: Inferring $f(t)$

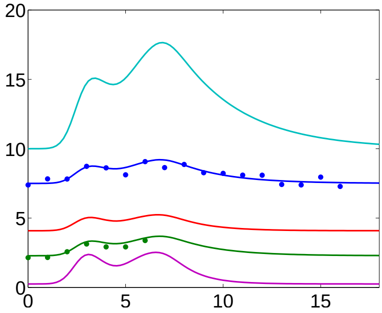


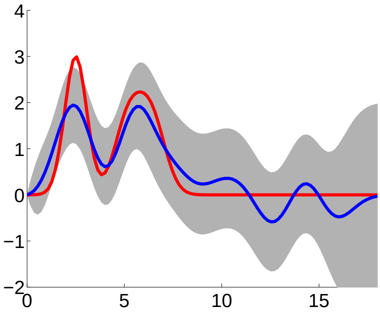


Artificial Example: Inferring $f(t)$

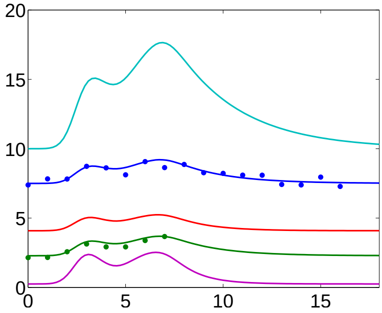


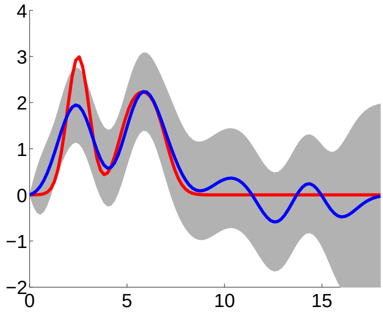
Artificial Example: Inferring $f(t)$



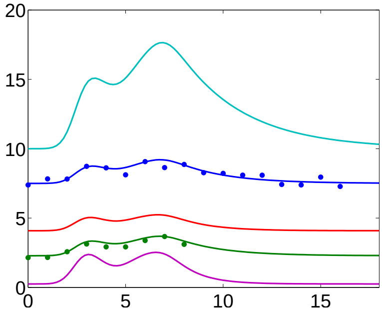


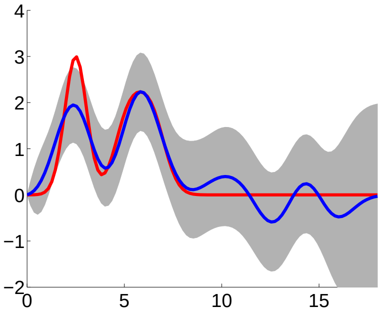
Artificial Example: Inferring $f(t)$



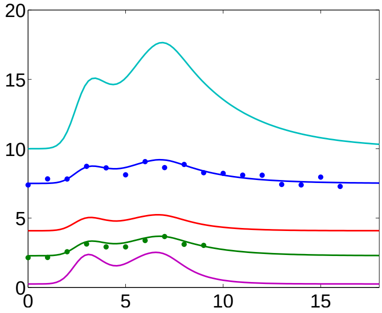


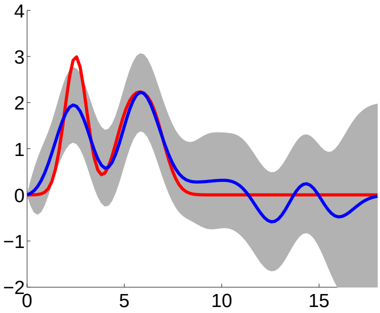
Artificial Example: Inferring $f(t)$



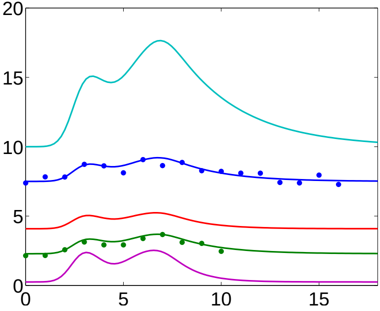


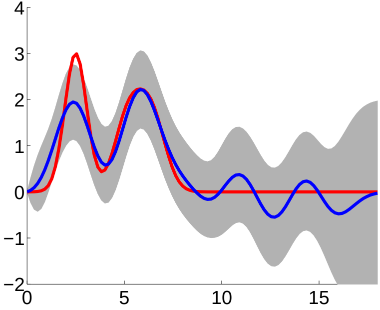
Artificial Example: Inferring $f(t)$



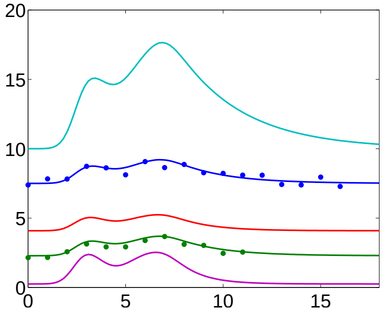


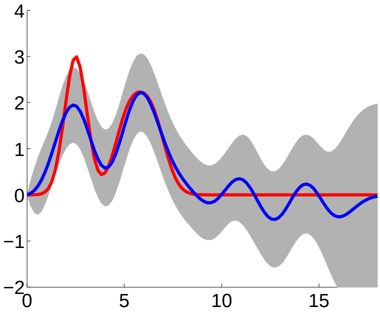
Artificial Example: Inferring $f(t)$



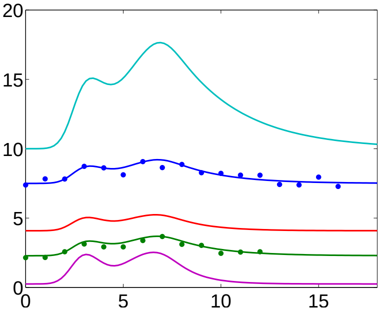


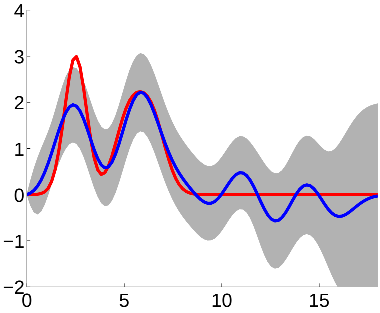
Artificial Example: Inferring $f(t)$



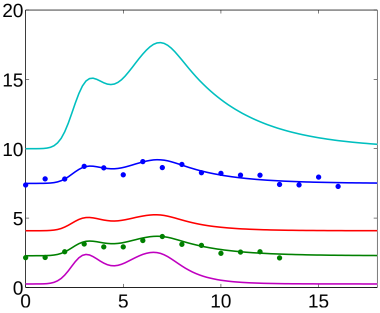


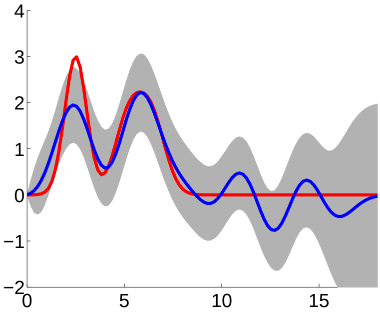
Artificial Example: Inferring $f(t)$



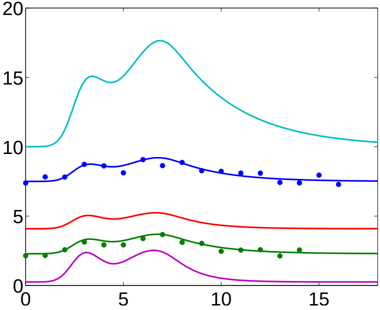


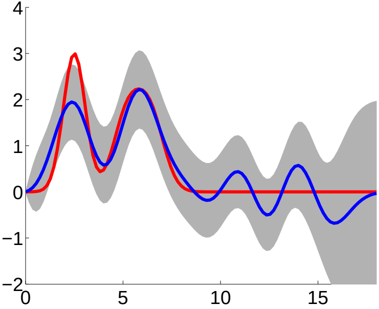
Artificial Example: Inferring $f(t)$



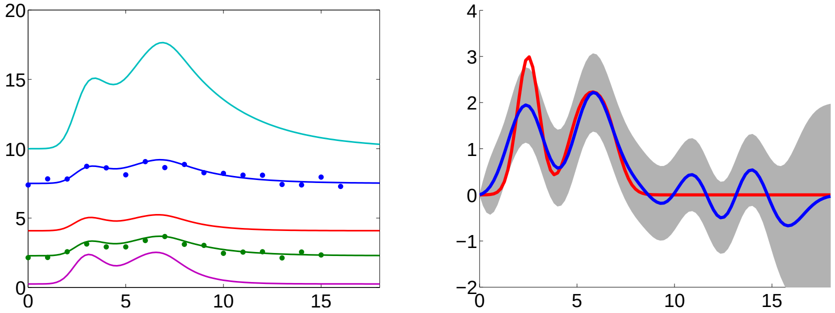


Artificial Example: Inferring $f(t)$

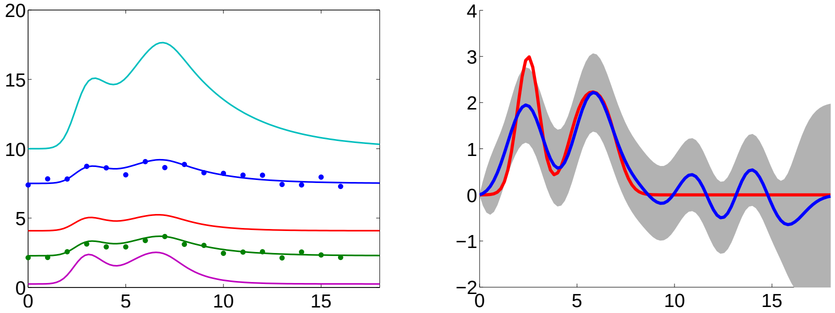




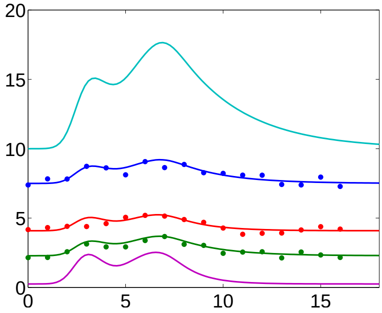
Artificial Example: Inferring $f(t)$

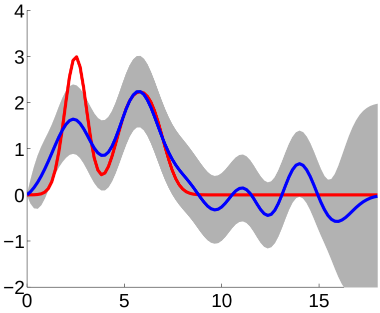


Artificial Example: Inferring $f(t)$

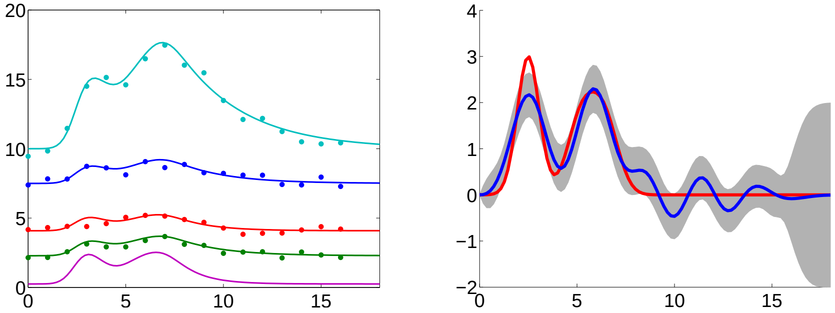


Artificial Example: Inferring $f(t)$

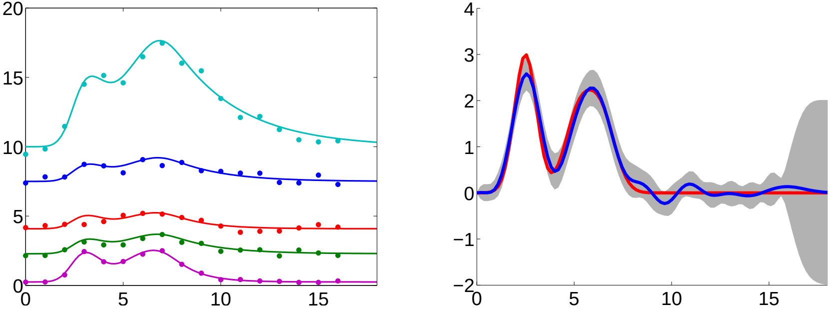




Artificial Example: Inferring $f(t)$

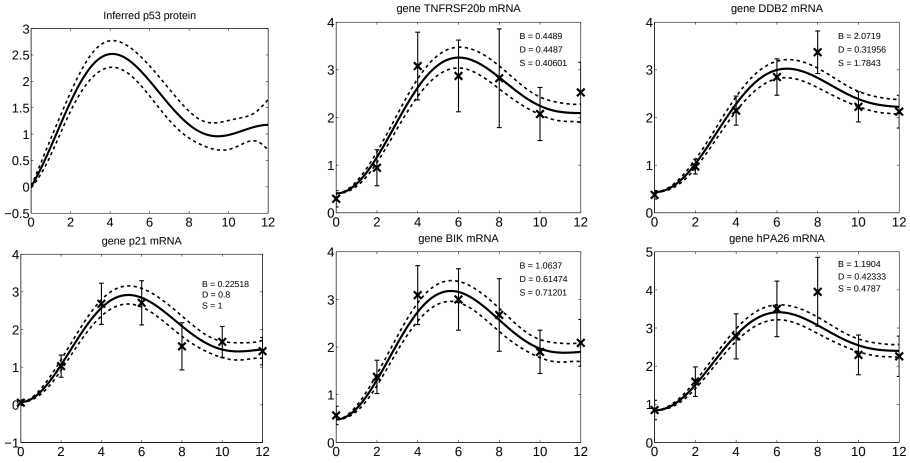


Artificial Example: Inferring $f(t)$



p53 (RBF covariance)

Pei Gao

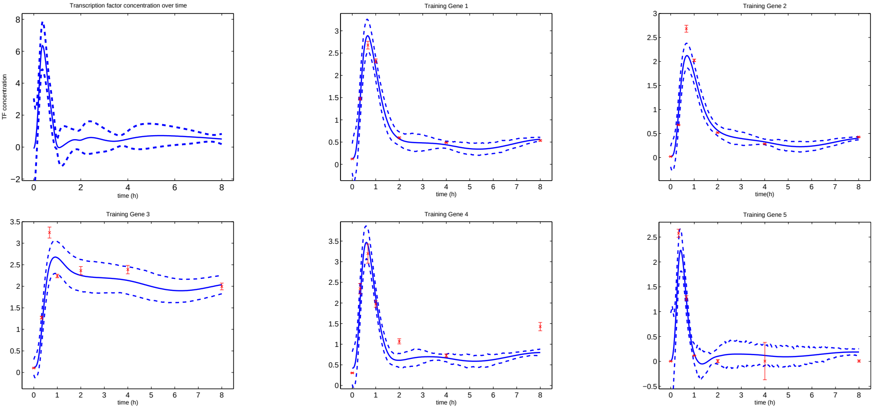


Ranking with ERK Signalling

- 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.

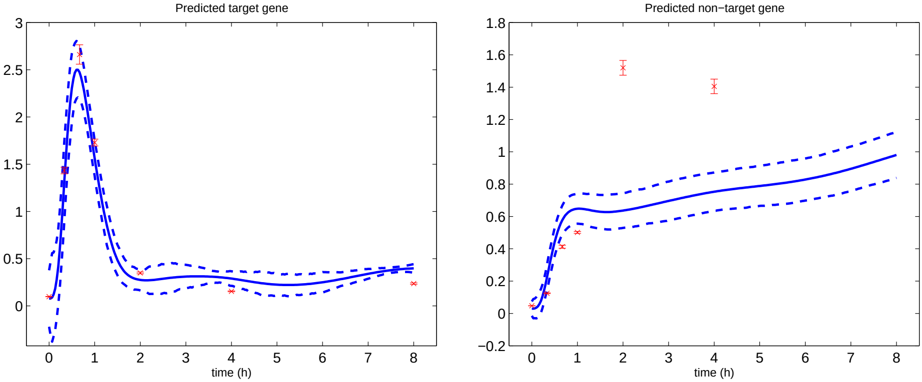
Elk-1 (MLP covariance)

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. Cascaded Differential Equations
4. Discussion and Future Work
5. 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.

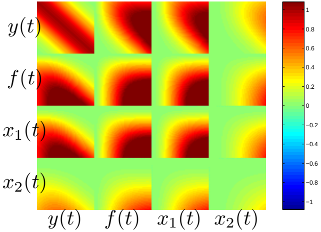
Covariance for Translation/Transcription Model

RBF covariance function for $y(t)$

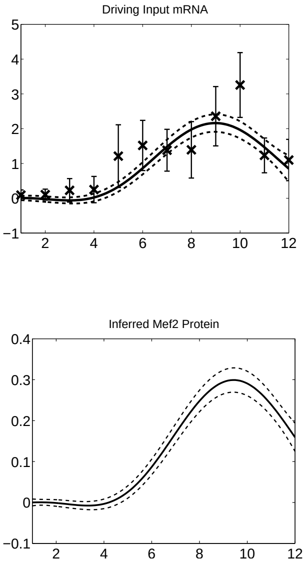
$$
\begin{aligned}
f(t) &= \sigma \exp(-\delta t) \int_0^t y(u) \exp(\delta u)\, \mathrm{d}u \\
x_i(t) &= \frac{B_i}{D_i} + S_i \exp(-D_i t) \int_0^t f(u) \exp(D_i u)\, \mathrm{d}u.
\end{aligned}
$$

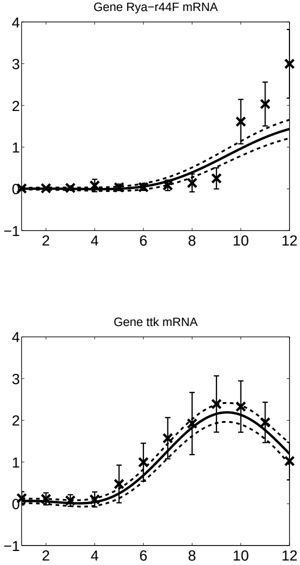
- Joint distribution for $x_1(t)$, $x_2(t)$, $f(t)$ and $y(t)$.
- Here:

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

Discussion and Future Work

- Integration of probabilistic inference with mechanistic models.
- These results are small simple systems (we skipped non-linear).
- 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

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).

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].