

# Applications of Gaussian Processes in Computational Biology

Neil D. Lawrence

Institute Curie  
Paris, 3rd April 2014

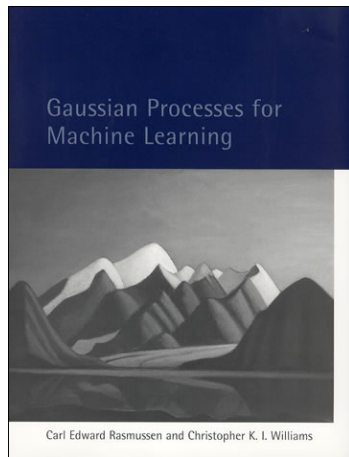
# Outline

Multivariate Gaussian Properties

Cascade Differential Equations

Multiple Transcription Factors

Conclusions



Rasmussen and Williams (2006)

# Outline

Multivariate Gaussian Properties

Cascade Differential Equations

Multiple Transcription Factors

Conclusions

# Recall Univariate Gaussian Properties

1. Sum of Gaussian variables is also Gaussian.

$$y_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$$

# Recall Univariate Gaussian Properties

1. Sum of Gaussian variables is also Gaussian.

$$y_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$$

$$\sum_{i=1}^n y_i \sim \mathcal{N}\left(\sum_{i=1}^n \mu_i, \sum_{i=1}^n \sigma_i^2\right)$$

# Recall Univariate Gaussian Properties

1. Sum of Gaussian variables is also Gaussian.

$$y_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$$

$$\sum_{i=1}^n y_i \sim \mathcal{N}\left(\sum_{i=1}^n \mu_i, \sum_{i=1}^n \sigma_i^2\right)$$

2. Scaling a Gaussian leads to a Gaussian.

# Recall Univariate Gaussian Properties

1. Sum of Gaussian variables is also Gaussian.

$$y_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$$

$$\sum_{i=1}^n y_i \sim \mathcal{N}\left(\sum_{i=1}^n \mu_i, \sum_{i=1}^n \sigma_i^2\right)$$

2. Scaling a Gaussian leads to a Gaussian.

$$y \sim \mathcal{N}(\mu, \sigma^2)$$



# Recall Univariate Gaussian Properties

1. Sum of Gaussian variables is also Gaussian.

$$y_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$$

$$\sum_{i=1}^n y_i \sim \mathcal{N}\left(\sum_{i=1}^n \mu_i, \sum_{i=1}^n \sigma_i^2\right)$$

2. Scaling a Gaussian leads to a Gaussian.

$$y \sim \mathcal{N}(\mu, \sigma^2)$$

$$wy \sim \mathcal{N}(w\mu, w^2\sigma^2)$$

# Multivariate Consequence

- If

$$\mathbf{t} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

# Multivariate Consequence

- If

$$\mathbf{t} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

- And

$$\mathbf{y} = \mathbf{W}\mathbf{t}$$

# Multivariate Consequence

- ▶ If

$$\mathbf{t} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

- ▶ And

$$\mathbf{y} = \mathbf{W}\mathbf{t}$$

- ▶ Then

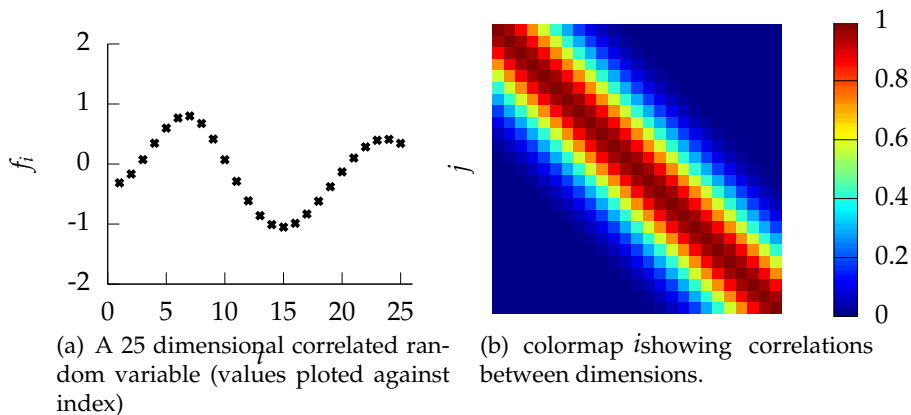
$$\mathbf{y} \sim \mathcal{N}(\mathbf{W}\boldsymbol{\mu}, \mathbf{W}\boldsymbol{\Sigma}\mathbf{W}^\top)$$

# 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,  $\mathbf{f} = [f_1, f_2 \dots f_{25}]$ .
- ▶ We will plot these points against their index.

# Gaussian Distribution Sample



**Figure :** A sample from a 25 dimensional Gaussian distribution.

# Gaussian Distribution Sample

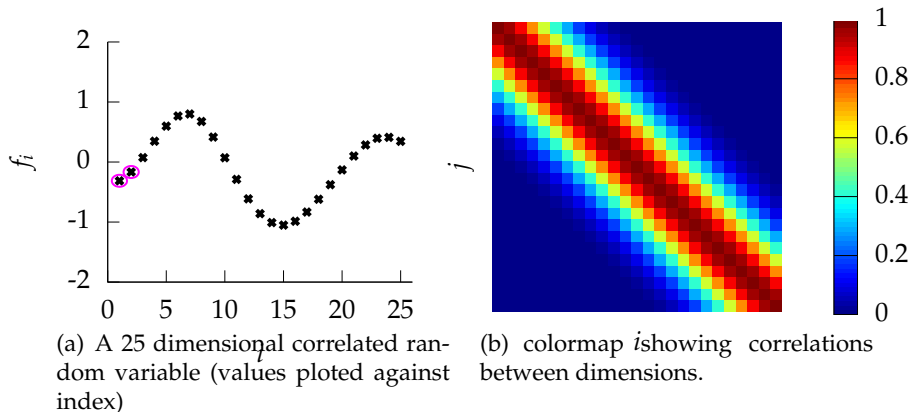
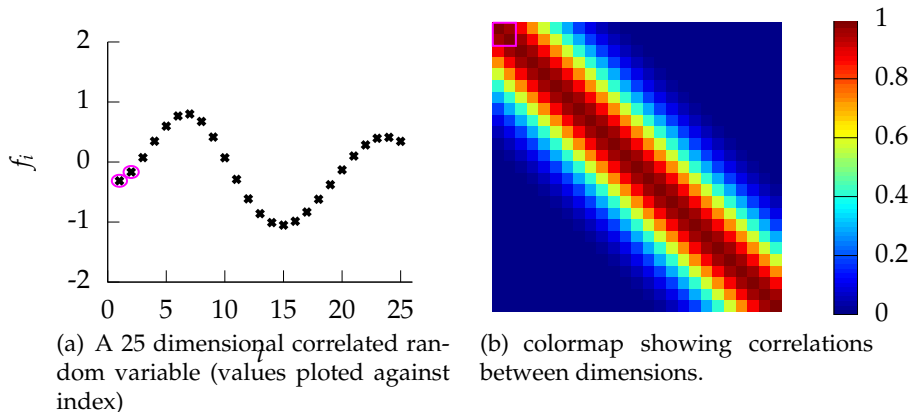


Figure : A sample from a 25 dimensional Gaussian distribution.

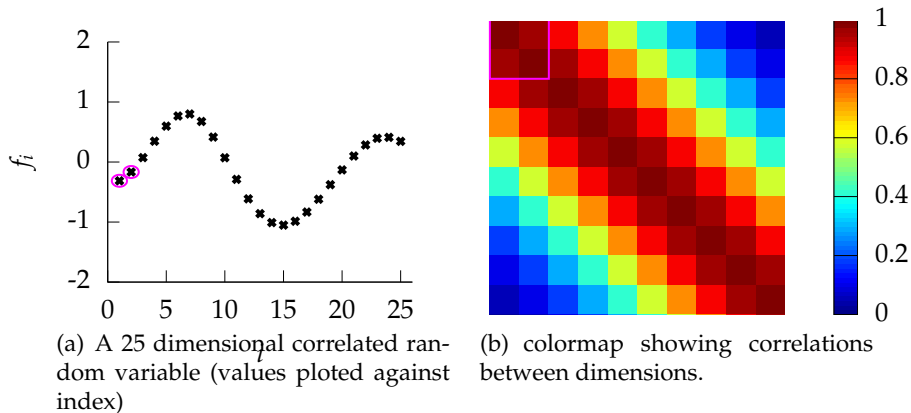
# Gaussian Distribution Sample



**Figure :** A sample from a 25 dimensional Gaussian distribution.



# Gaussian Distribution Sample



**Figure :** A sample from a 25 dimensional Gaussian distribution.

# Gaussian Distribution Sample

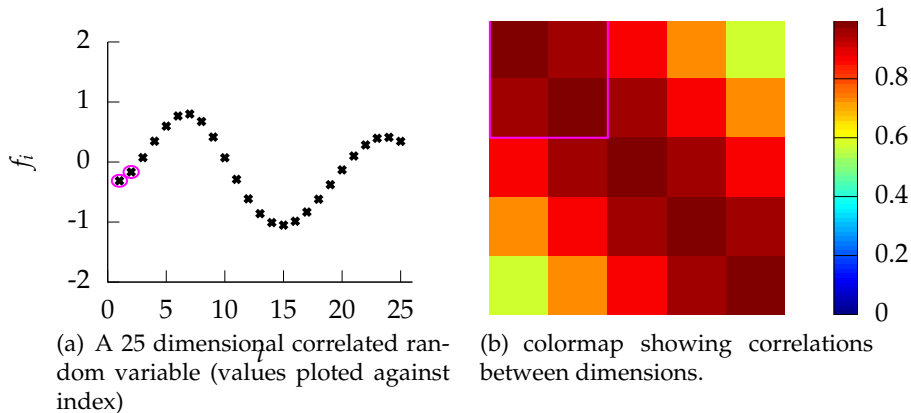
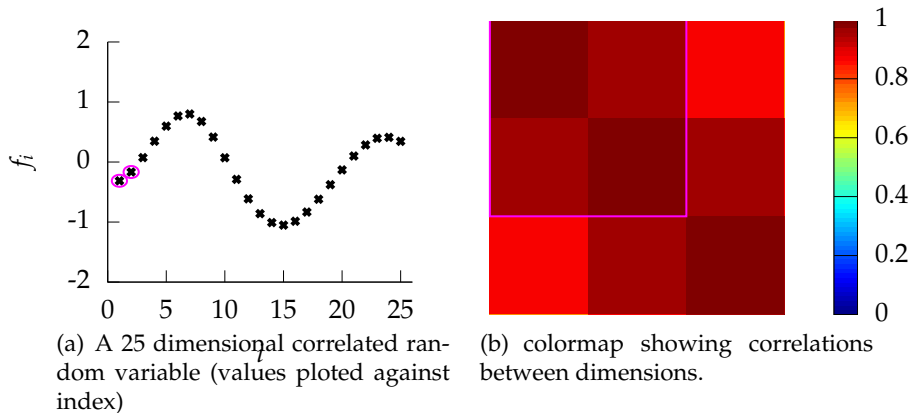


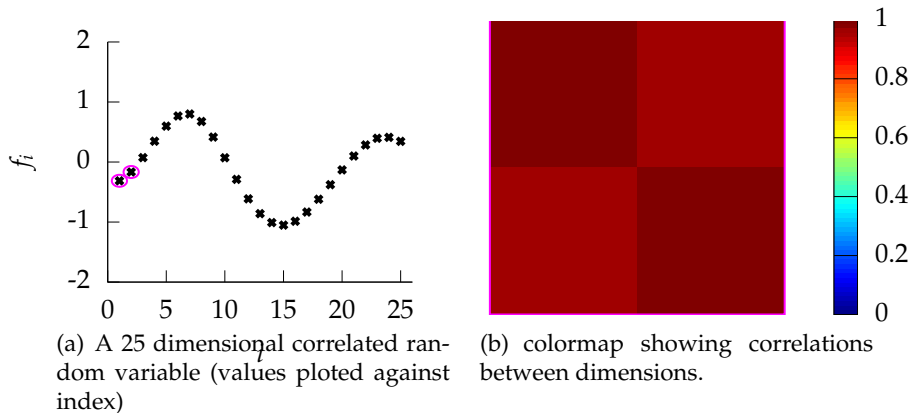
Figure : A sample from a 25 dimensional Gaussian distribution.

# Gaussian Distribution Sample



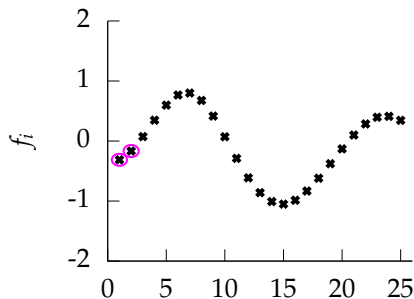
**Figure :** A sample from a 25 dimensional Gaussian distribution.

# Gaussian Distribution Sample



**Figure :** A sample from a 25 dimensional Gaussian distribution.

# Gaussian Distribution Sample



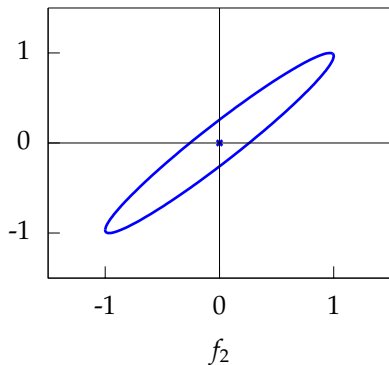
(a) A 25 dimensional correlated random variable (values plotted against index)

$$\begin{bmatrix} 1 & 0.96587 \\ 0.96587 & 1 \end{bmatrix}$$

(b) correlation between  $f_1$  and  $f_2$ .

**Figure :** A sample from a 25 dimensional Gaussian distribution.

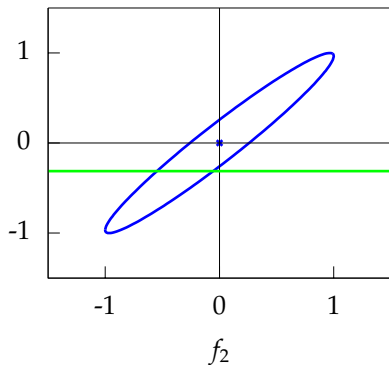
## Prediction of $f_2$ from $f_1$



$$\begin{bmatrix} 1 & 0.96587 \\ 0.96587 & 1 \end{bmatrix}$$

- The single contour of the Gaussian density represents the joint distribution,  $p(f_1, f_2)$ .

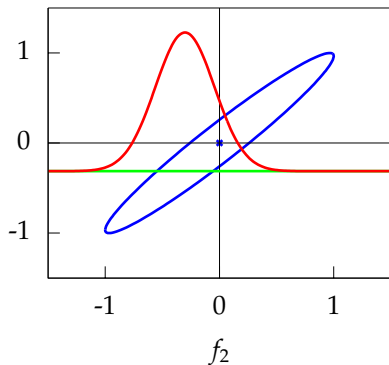
## Prediction of $f_2$ from $f_1$



$$\begin{bmatrix} 1 & 0.96587 \\ 0.96587 & 1 \end{bmatrix}$$

- ▶ The single contour of the Gaussian density represents the **joint distribution**,  $p(f_1, f_2)$ .
- ▶ We observe that  $f_1 = -0.313$ .

## Prediction of $f_2$ from $f_1$

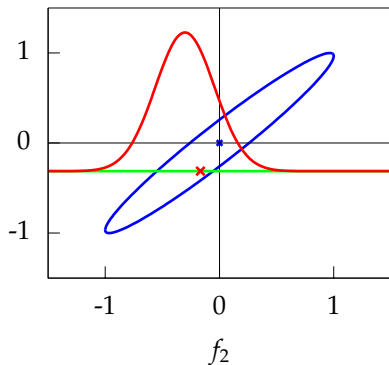


$$\begin{bmatrix} 1 & 0.96587 \\ 0.96587 & 1 \end{bmatrix}$$

- ▶ The single contour of the Gaussian density represents the **joint distribution**,  $p(f_1, f_2)$ .
- ▶ We observe that  $f_1 = -0.313$ .
- ▶ Conditional density:  $p(f_2 | f_1 = -0.313)$ .



## Prediction of $f_2$ from $f_1$



$$\begin{bmatrix} 1 & 0.96587 \\ 0.96587 & 1 \end{bmatrix}$$

- ▶ The single contour of the Gaussian density represents the **joint distribution**,  $p(f_1, f_2)$ .
- ▶ We observe that  $f_1 = -0.313$ .
- ▶ Conditional density:  $p(f_2|f_1 = -0.313)$ .

# Prediction with Correlated Gaussians

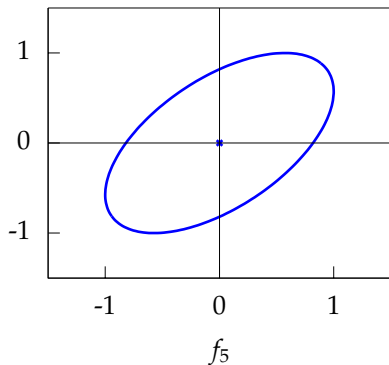
- ▶ Prediction of  $f_2$  from  $f_1$  requires *conditional density*.
- ▶ Conditional density is *also* Gaussian.

$$p(f_2|f_1) = \mathcal{N}\left(f_2 \middle| \frac{k_{1,2}}{k_{1,1}} f_1, k_{2,2} - \frac{k_{1,2}^2}{k_{1,1}}\right)$$

where covariance of joint density is given by

$$\mathbf{K} = \begin{bmatrix} k_{1,1} & k_{1,2} \\ k_{2,1} & k_{2,2} \end{bmatrix}$$

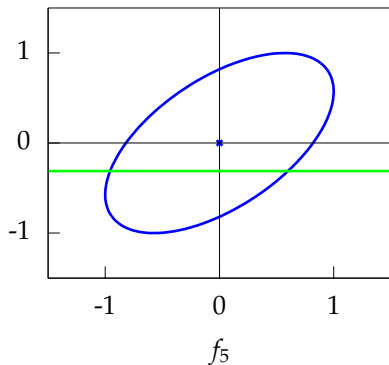
## Prediction of $f_5$ from $f_1$



$$\begin{bmatrix} 1 & 0.57375 \\ 0.57375 & 1 \end{bmatrix}$$

- The single contour of the Gaussian density represents the joint distribution,  $p(f_1, f_5)$ .

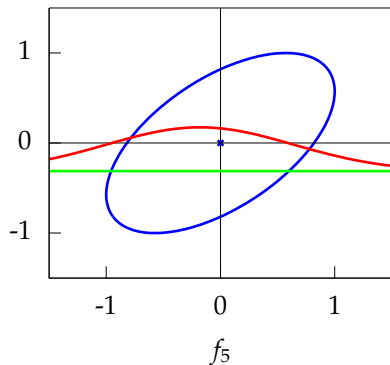
## Prediction of $f_5$ from $f_1$



$$\begin{bmatrix} 1 & 0.57375 \\ 0.57375 & 1 \end{bmatrix}$$

- ▶ The single contour of the Gaussian density represents the **joint distribution**,  $p(f_1, f_5)$ .
- ▶ We observe that  $f_1 = -0.313$ .

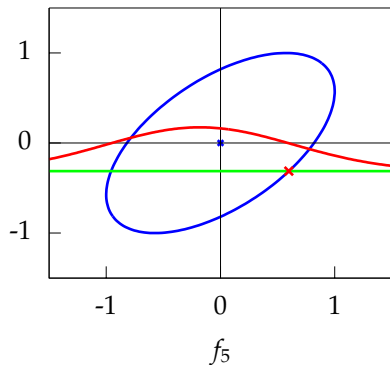
## Prediction of $f_5$ from $f_1$



$$\begin{bmatrix} 1 & 0.57375 \\ 0.57375 & 1 \end{bmatrix}$$

- ▶ The single contour of the Gaussian density represents the **joint distribution**,  $p(f_1, f_5)$ .
- ▶ We observe that  $f_1 = -0.313$ .
- ▶ Conditional density:  $p(f_5 | f_1 = -0.313)$ .

## Prediction of $f_5$ from $f_1$



$$\begin{bmatrix} 1 & 0.57375 \\ 0.57375 & 1 \end{bmatrix}$$

- ▶ The single contour of the Gaussian density represents the **joint distribution**,  $p(f_1, f_5)$ .
- ▶ We observe that  $f_1 = -0.313$ .
- ▶ Conditional density:  $p(f_5|f_1 = -0.313)$ .

# Prediction with Correlated Gaussians

- ▶ Prediction of  $\mathbf{f}_*$  from  $\mathbf{f}$  requires multivariate *conditional density*.
- ▶ Multivariate conditional density is *also* Gaussian.

$$p(\mathbf{f}_*|\mathbf{f}) = \mathcal{N}\left(\mathbf{f}_*|\mathbf{K}_{*,\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{f}, \mathbf{K}_{*,*} - \mathbf{K}_{*,\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{K}_{\mathbf{f},*}\right)$$

- ▶ Here covariance of joint density is given by

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{\mathbf{f},\mathbf{f}} & \mathbf{K}_{*,\mathbf{f}} \\ \mathbf{K}_{\mathbf{f},*} & \mathbf{K}_{*,*} \end{bmatrix}$$

# Prediction with Correlated Gaussians

- ▶ Prediction of  $\mathbf{f}_*$  from  $\mathbf{f}$  requires multivariate *conditional density*.
- ▶ Multivariate conditional density is *also* Gaussian.

$$p(\mathbf{f}_*|\mathbf{f}) = \mathcal{N}(\mathbf{f}_*|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$\boldsymbol{\mu} = \mathbf{K}_{*,\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{f}$$

$$\boldsymbol{\Sigma} = \mathbf{K}_{*,*} - \mathbf{K}_{*,\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{K}_{\mathbf{f},*}$$

- ▶ Here covariance of joint density is given by

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{\mathbf{f},\mathbf{f}} & \mathbf{K}_{*,\mathbf{f}} \\ \mathbf{K}_{\mathbf{f},*} & \mathbf{K}_{*,*} \end{bmatrix}$$



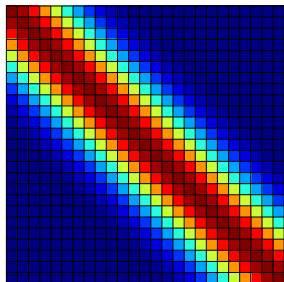
# Covariance Functions

Where did this covariance matrix come from?

## Exponentiated Quadratic Kernel Function (RBF, Squared Exponential, Gaussian)

$$k(t, t') = \alpha \exp\left(-\frac{\|t - t'\|_2^2}{2\ell^2}\right)$$

- ▶ 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 known as a kernel.



# Covariance Functions

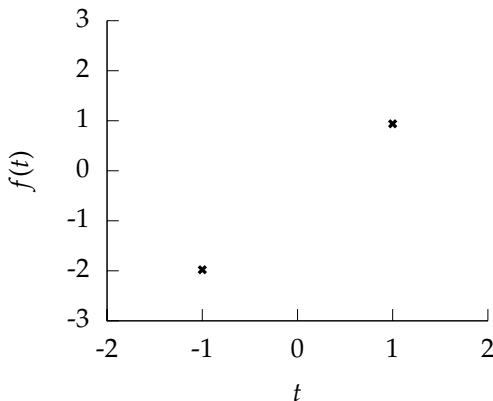
Where did this covariance matrix come from?

## Exponentiated Quadratic Kernel Function (RBF, Squared Exponential, Gaussian)

$$k(t, t') = \alpha \exp\left(-\frac{\|t - t'\|_2^2}{2\ell^2}\right)$$

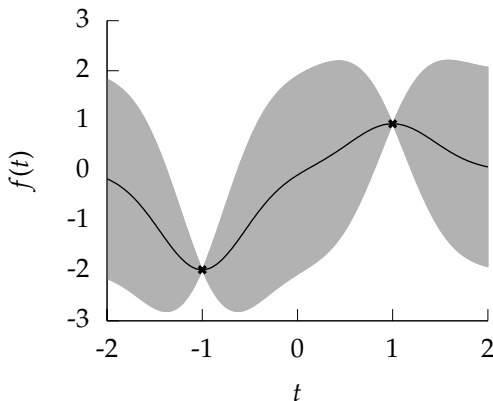
- ▶ Covariance matrix is built using the *inputs* to the function  $t$ .
- ▶ For the example above it was based on Euclidean distance.
- ▶ The covariance function is also known as a kernel.

# Gaussian Process Interpolation



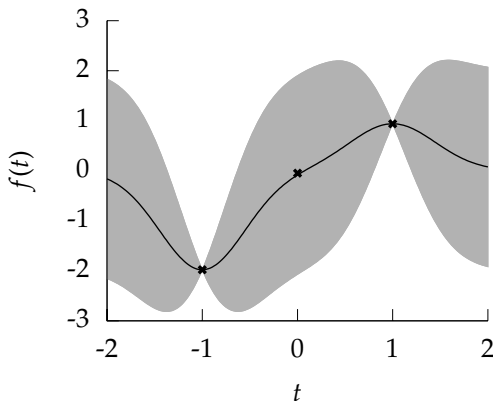
**Figure :** Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)).  
Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

# Gaussian Process Interpolation



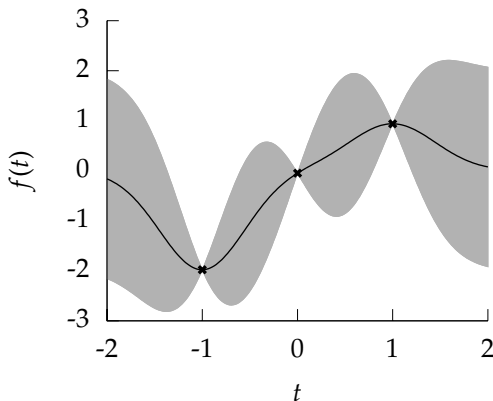
**Figure :** Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)).  
Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

# Gaussian Process Interpolation



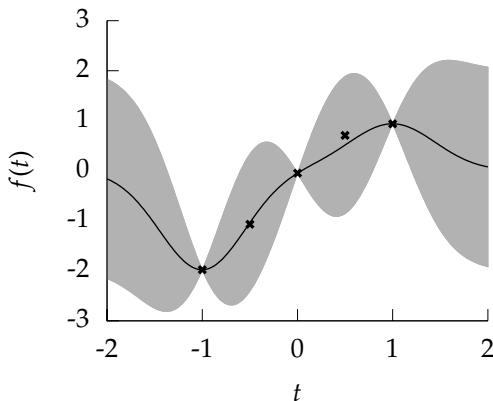
**Figure :** Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)).  
Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

# Gaussian Process Interpolation



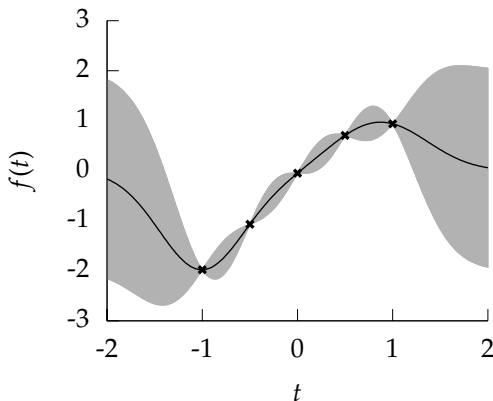
**Figure :** Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)).  
Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

# Gaussian Process Interpolation



**Figure :** Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)).  
Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

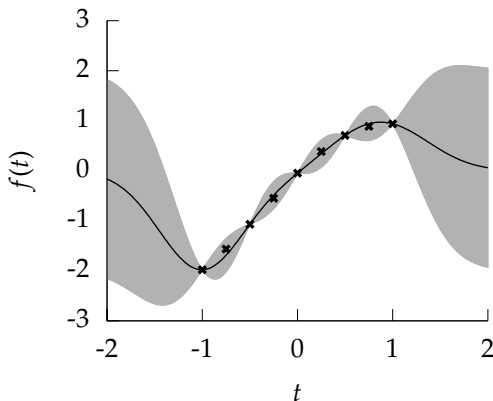
# Gaussian Process Interpolation



**Figure :** Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)).  
Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

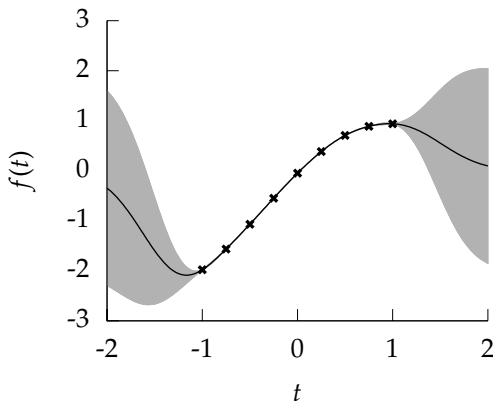


# Gaussian Process Interpolation



**Figure :** Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)).  
Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

# Gaussian Process Interpolation



**Figure :** Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

# Gaussian Noise

- ▶ Gaussian noise model,

$$p(y_i|f_i) = \mathcal{N}(y_i|f_i, \sigma^2)$$

where  $\sigma^2$  is the variance of the noise.

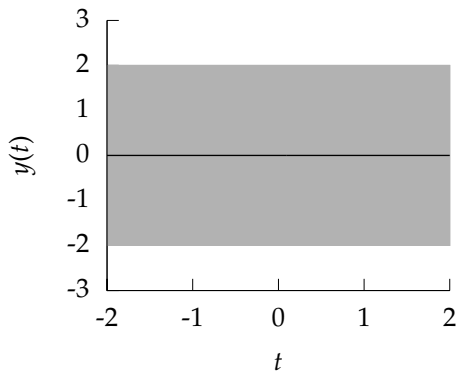
- ▶ Equivalent to a covariance function of the form

$$k(t_i, t_j) = \delta_{i,j} \sigma^2$$

where  $\delta_{i,j}$  is the Kronecker delta function.

- ▶ Additive nature of Gaussians means we can simply add this term to existing covariance matrices.

# Gaussian Process Regression



**Figure :** Examples include WiFi localization, C14 calibration curve.

# Gaussian Process Regression

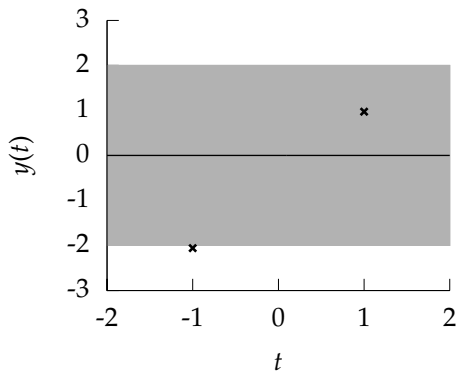
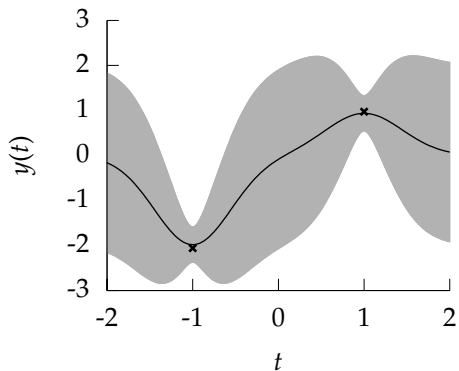


Figure : Examples include WiFi localization, C14 calibration curve.

# Gaussian Process Regression



**Figure :** Examples include WiFi localization, C14 calibration curve.

# Gaussian Process Regression

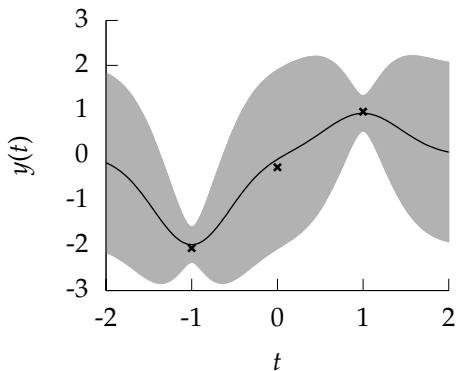
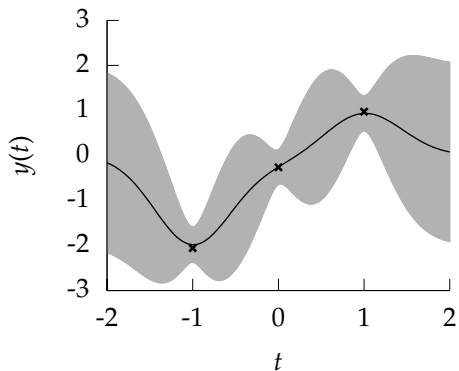


Figure : Examples include WiFi localization, C14 calibration curve.

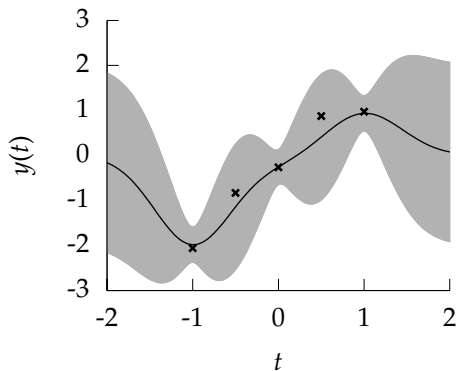
# Gaussian Process Regression



**Figure :** Examples include WiFi localization, C14 calibration curve.



# Gaussian Process Regression



**Figure :** Examples include WiFi localization, C14 calibration curve.

# Gaussian Process Regression

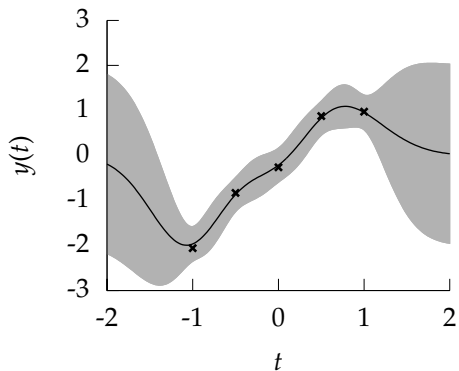


Figure : Examples include WiFi localization, C14 calibration curve.

# Gaussian Process Regression

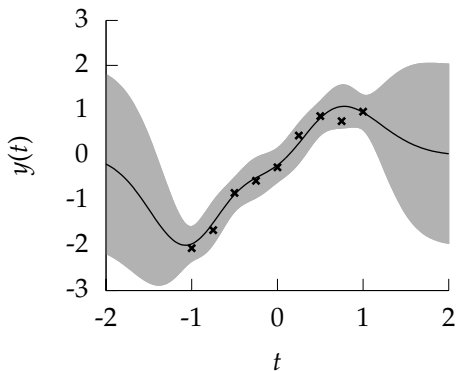


Figure : Examples include WiFi localization, C14 calibration curve.

# Gaussian Process Regression

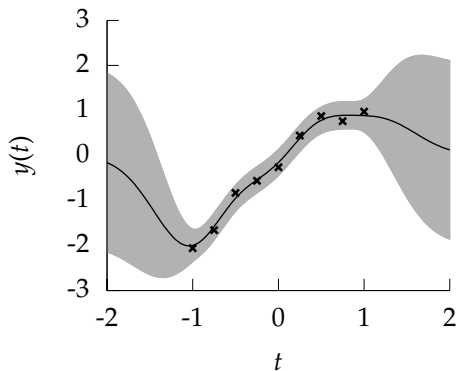


Figure : Examples include WiFi localization, C14 calibration curve.

# Learning Covariance Parameters

Can we determine covariance parameters from the data?

$$\mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\mathbf{K}|} \exp\left(-\frac{\mathbf{y}^\top \mathbf{K}^{-1} \mathbf{y}}{2}\right)$$

The parameters are *inside* the covariance function (matrix).

$$k_{i,j} = k(t_i, t_j; \boldsymbol{\theta})$$

# Learning Covariance Parameters

Can we determine covariance parameters from the data?

$$\mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\mathbf{K}|} \exp\left(-\frac{\mathbf{y}^\top \mathbf{K}^{-1} \mathbf{y}}{2}\right)$$

The parameters are *inside* the covariance function (matrix).

$$k_{i,j} = k(t_i, t_j; \boldsymbol{\theta})$$

# Learning Covariance Parameters

Can we determine covariance parameters from the data?

$$\log \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}) = -\frac{1}{2} \log |\mathbf{K}| - \frac{\mathbf{y}^\top \mathbf{K}^{-1} \mathbf{y}}{2} - \frac{n}{2} \log 2\pi$$

The parameters are *inside* the covariance function (matrix).

$$k_{i,j} = k(t_i, t_j; \boldsymbol{\theta})$$

# Learning Covariance Parameters

Can we determine covariance parameters from the data?

$$E(\theta) = \frac{1}{2} \log |\mathbf{K}| + \frac{\mathbf{y}^\top \mathbf{K}^{-1} \mathbf{y}}{2}$$

The parameters are *inside* the covariance function (matrix).

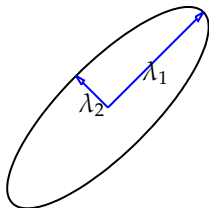
$$k_{i,j} = k(t_i, t_j; \theta)$$



# Eigendecomposition of Covariance

A useful decomposition for understanding the objective function.

$$\mathbf{K} = \mathbf{R}\mathbf{\Lambda}^2\mathbf{R}^\top$$



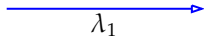
Diagonal of  $\mathbf{\Lambda}$  represents distance along axes.

$\mathbf{R}$  gives a rotation of these axes.

where  $\mathbf{\Lambda}$  is a *diagonal* matrix and  $\mathbf{R}^\top\mathbf{R} = \mathbf{I}$ .

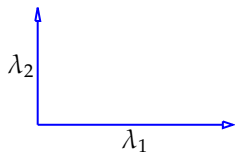
## Capacity control: $\log |\mathbf{K}|$

$$\mathbf{\Lambda} = \begin{bmatrix} \boxed{\lambda_1 & 0} \\ 0 & \lambda_2 \end{bmatrix}$$



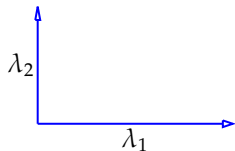
## Capacity control: $\log |\mathbf{K}|$

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$



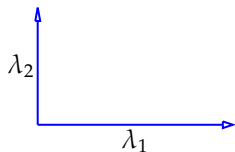
## Capacity control: $\log |\mathbf{K}|$

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$



## Capacity control: $\log |\mathbf{K}|$

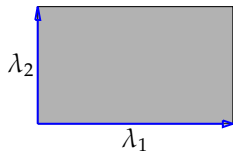
$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$



$$|\mathbf{\Lambda}| = \lambda_1 \lambda_2$$

## Capacity control: $\log |\mathbf{K}|$

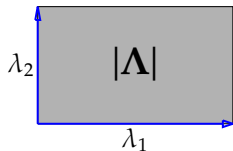
$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$



$$|\mathbf{\Lambda}| = \lambda_1 \lambda_2$$

## Capacity control: $\log |\mathbf{K}|$

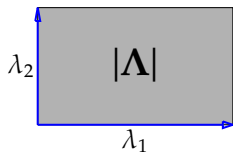
$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$



$$|\mathbf{\Lambda}| = \lambda_1 \lambda_2$$

## Capacity control: $\log |\mathbf{K}|$

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

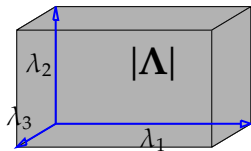


$$|\mathbf{\Lambda}| = \lambda_1 \lambda_2$$



## Capacity control: $\log |\mathbf{K}|$

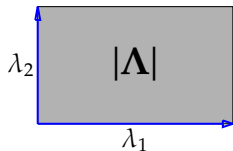
$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$



$$|\mathbf{\Lambda}| = \lambda_1 \lambda_2 \lambda_3$$

## Capacity control: $\log |\mathbf{K}|$

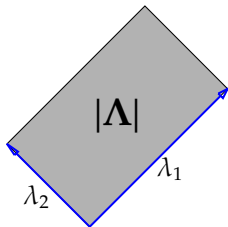
$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$



$$|\mathbf{\Lambda}| = \lambda_1 \lambda_2$$

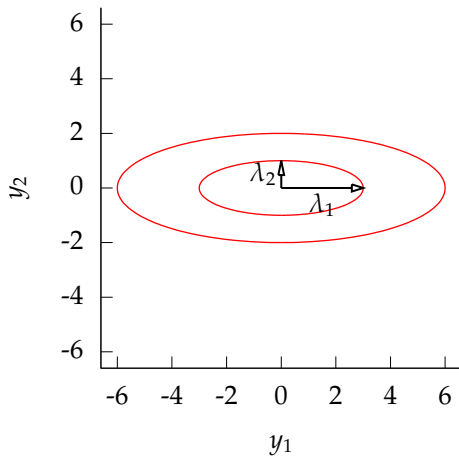
## Capacity control: $\log |\mathbf{K}|$

$$\mathbf{R}\mathbf{\Lambda} = \begin{bmatrix} w_{1,1} & w_{1,2} \\ w_{2,1} & w_{2,2} \end{bmatrix}$$

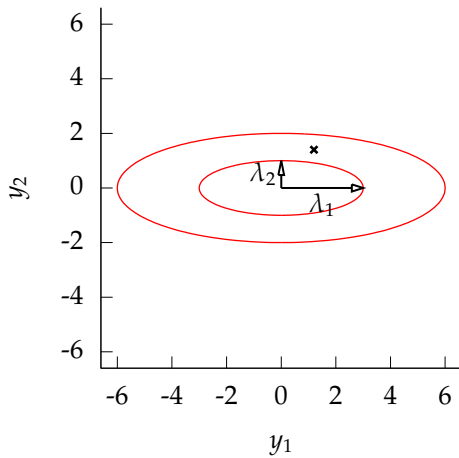


$$|\mathbf{R}\mathbf{\Lambda}| = \lambda_1 \lambda_2$$

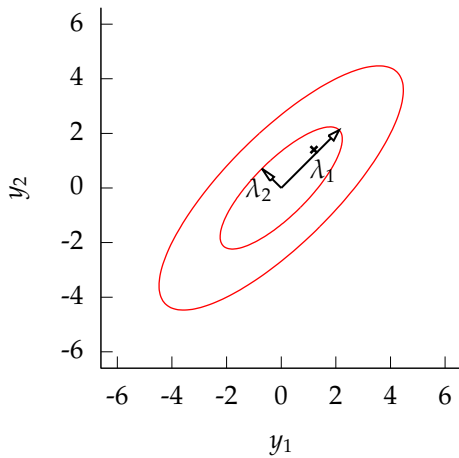
Data Fit:  $\frac{\mathbf{y}^\top \mathbf{K}^{-1} \mathbf{y}}{2}$



Data Fit:  $\frac{\mathbf{y}^\top \mathbf{K}^{-1} \mathbf{y}}{2}$

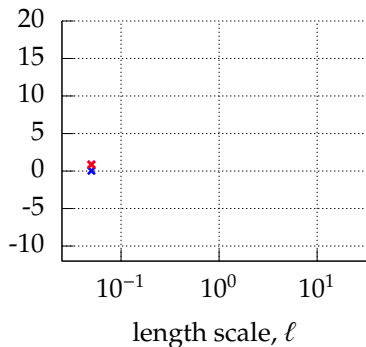
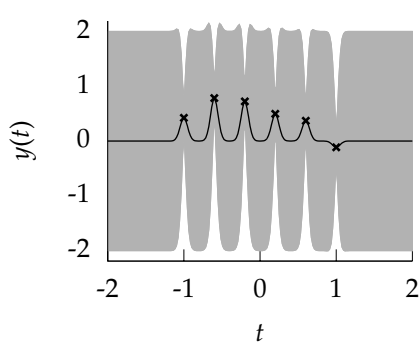


Data Fit:  $\frac{\mathbf{y}^\top \mathbf{K}^{-1} \mathbf{y}}{2}$



# Learning Covariance Parameters

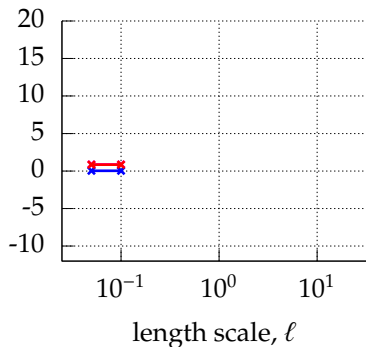
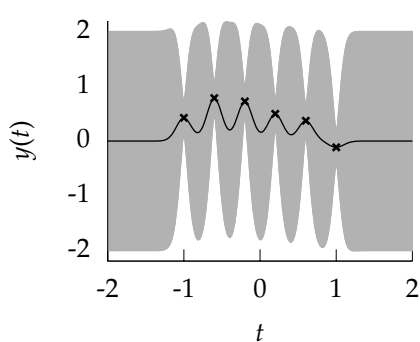
Can we determine length scales and noise levels from the data?



$$E(\theta) = \frac{1}{2} \log |\mathbf{K}| + \frac{\mathbf{y}^\top \mathbf{K}^{-1} \mathbf{y}}{2}$$

# Learning Covariance Parameters

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

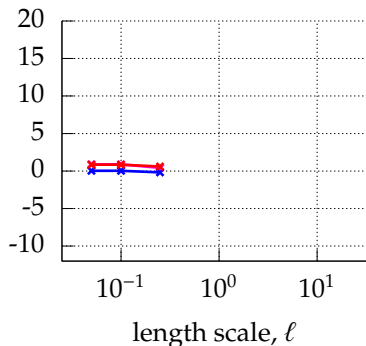
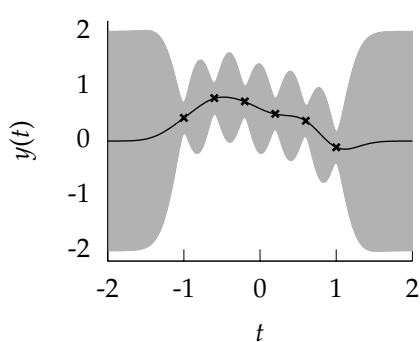


$$E(\theta) = \frac{1}{2} \log |\mathbf{K}| + \frac{\mathbf{y}^\top \mathbf{K}^{-1} \mathbf{y}}{2}$$



# Learning Covariance Parameters

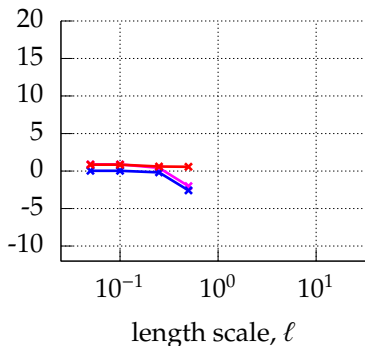
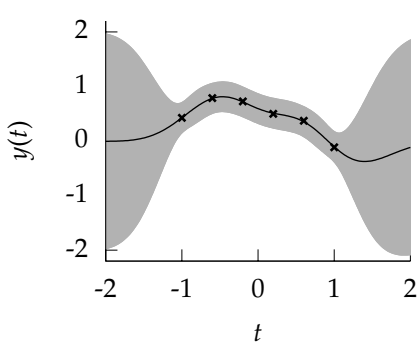
Can we determine length scales and noise levels from the data?



$$E(\theta) = \frac{1}{2} \log |\mathbf{K}| + \frac{\mathbf{y}^\top \mathbf{K}^{-1} \mathbf{y}}{2}$$

# Learning Covariance Parameters

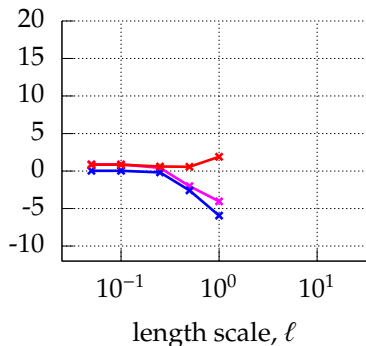
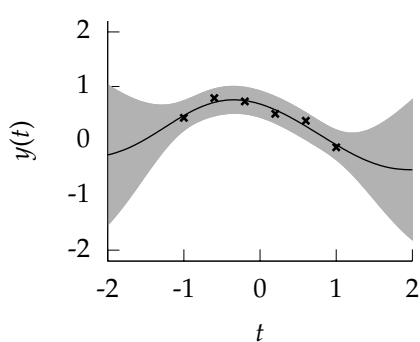
Can we determine length scales and noise levels from the data?



$$E(\theta) = \frac{1}{2} \log |\mathbf{K}| + \frac{\mathbf{y}^\top \mathbf{K}^{-1} \mathbf{y}}{2}$$

# Learning Covariance Parameters

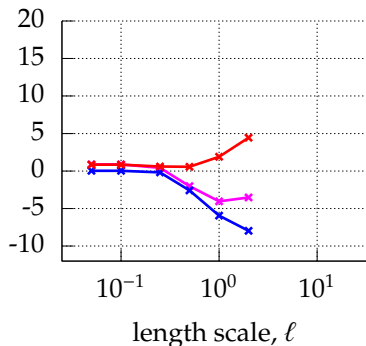
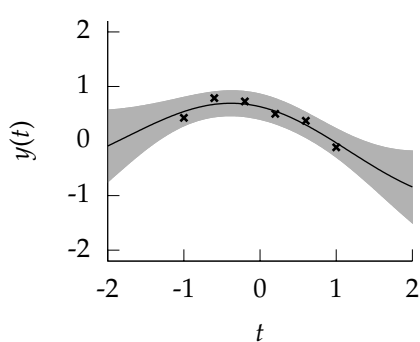
Can we determine length scales and noise levels from the data?



$$E(\theta) = \frac{1}{2} \log |\mathbf{K}| + \frac{\mathbf{y}^\top \mathbf{K}^{-1} \mathbf{y}}{2}$$

# Learning Covariance Parameters

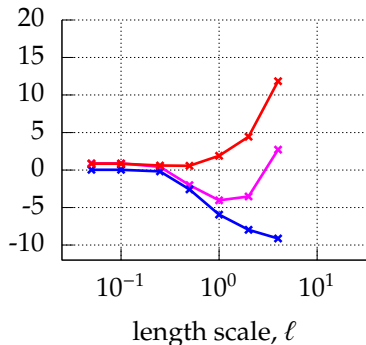
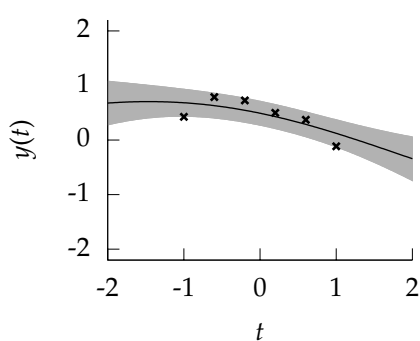
Can we determine length scales and noise levels from the data?



$$E(\theta) = \frac{1}{2} \log |\mathbf{K}| + \frac{\mathbf{y}^\top \mathbf{K}^{-1} \mathbf{y}}{2}$$

# Learning Covariance Parameters

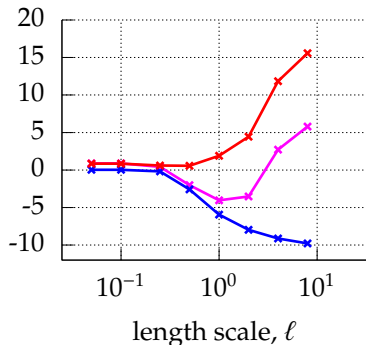
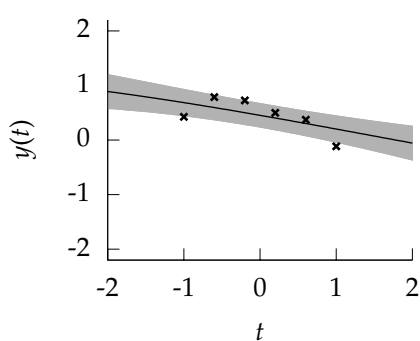
Can we determine length scales and noise levels from the data?



$$E(\theta) = \frac{1}{2} \log |\mathbf{K}| + \frac{\mathbf{y}^\top \mathbf{K}^{-1} \mathbf{y}}{2}$$

# Learning Covariance Parameters

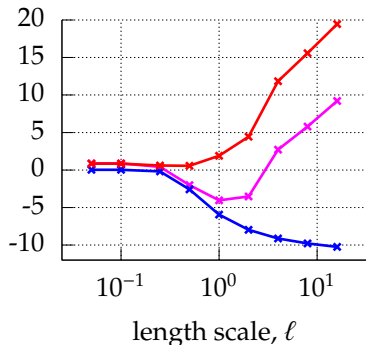
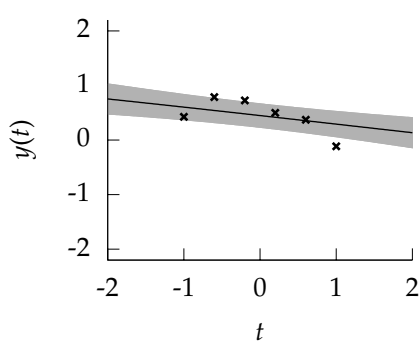
Can we determine length scales and noise levels from the data?



$$E(\theta) = \frac{1}{2} \log |\mathbf{K}| + \frac{\mathbf{y}^\top \mathbf{K}^{-1} \mathbf{y}}{2}$$

# Learning Covariance Parameters

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



$$E(\theta) = \frac{1}{2} \log |\mathbf{K}| + \frac{\mathbf{y}^\top \mathbf{K}^{-1} \mathbf{y}}{2}$$

# Gene Expression Example

- ▶ Given given expression levels in the form of a time series from Della Gatta et al. (2008).
- ▶ Want to detect if a gene is expressed or not, fit a GP to each gene (Kalaitzis and Lawrence, 2011).



RESEARCH ARTICLE

Open Access

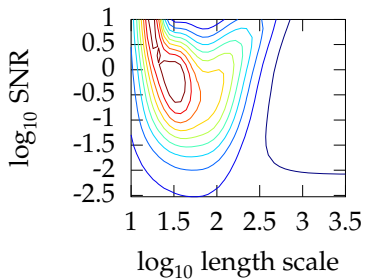
# A Simple Approach to Ranking Differentially Expressed Gene Expression Time Courses through Gaussian Process Regression

Alfredo A Kalaitzis\* and Neil D Lawrence\*

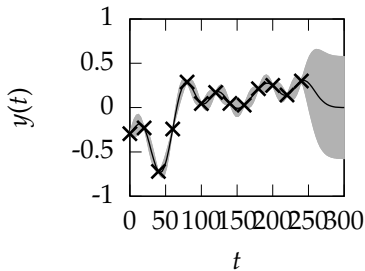
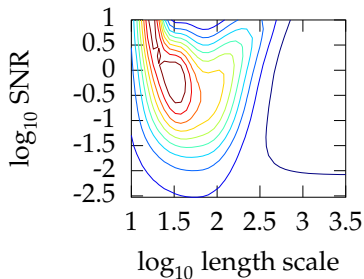
## Abstract

**Background:** The analysis of gene expression from time series underpins many biological studies. Two basic forms of analysis recur for data of this type: removing inactive (quiet) genes from the study and determining which genes are differentially expressed. Often these analysis stages are applied disregarding the fact that the data is drawn from a time series. In this paper we propose a simple model for accounting for the underlying temporal nature of the data based on a Gaussian process.

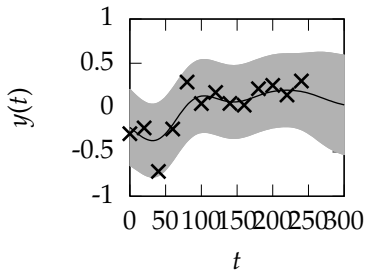
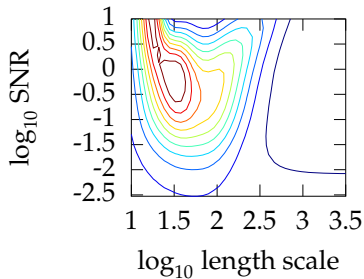
**Results:** We review Gaussian process (GP) regression for estimating the continuous trajectories underlying in gene expression time-series. We present a simple approach which can be used to filter quiet genes, or for the case of time series in the form of expression ratios, quantify differential expression. We assess via ROC curves the rankings produced by our regression framework and compare them to a recently proposed hierarchical Bayesian model for the analysis of gene expression time-series (BATS). We compare on both simulated and experimental data showing that the proposed approach considerably outperforms the current state of the art.



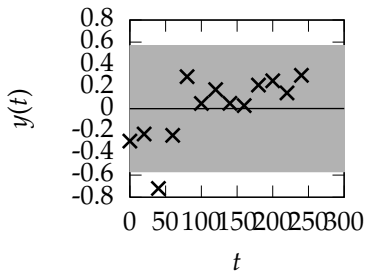
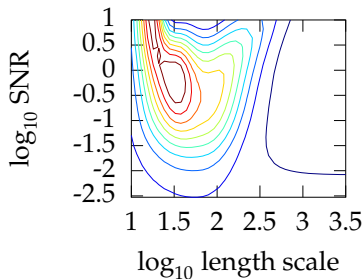
Contour plot of Gaussian process likelihood.



Optima: length scale of 1.2221 and  $\log_{10}$  SNR of 1.9654  
 log likelihood is -0.22317.



Optima: length scale of 1.5162 and  $\log_{10}$  SNR of 0.21306  
 log likelihood is -0.23604.



Optima: length scale of 2.9886 and  $\log_{10}$  SNR of -4.506  
 log likelihood is -2.1056.

# Can a Biologist Fix a Radio? Lazebnik (2002)

## The Case for Systems Biology

*“It is difficult to find a black cat in a dark room, especially if there is no cat.”*

# Can a Biologist Fix a Radio? Lazebnik (2002)

## The Case for Systems Biology

*“It is difficult to find a black cat in a dark room, especially if there is no cat.”*

- ▶ Biological systems are immensely complicated.

# Can a Biologist Fix a Radio? Lazebnik (2002)

## The Case for Systems Biology

*"It is difficult to find a black cat in a dark room, especially if there is no cat."*

- ▶ Biological systems are immensely complicated.
- ▶ Lazebnik argues the need for models that are quantitative.



# Can a Biologist Fix a Radio? Lazebnik (2002)

## The Case for Systems Biology

*"It is difficult to find a black cat in a dark room, especially if there is no cat."*

- ▶ Biological systems are immensely complicated.
- ▶ Lazebnik argues the need for models that are quantitative.
  - ▶ Such models should be predictive of biological behaviour.

# Can a Biologist Fix a Radio? Lazebnik (2002)

## The Case for Systems Biology

*"It is difficult to find a black cat in a dark room, especially if there is no cat."*

- ▶ Biological systems are immensely complicated.
- ▶ Lazebnik argues the need for models that are quantitative.
  - ▶ Such models should be predictive of biological behaviour.
  - ▶ Such models need to be combined with biological data.

# Can a Biologist Fix a Radio? Lazebnik (2002)

## The Case for Systems Biology

*"It is difficult to find a black cat in a dark room, especially if there is no cat."*

- ▶ Biological systems are immensely complicated.
- ▶ Lazebnik argues the need for models that are quantitative.
  - ▶ Such models should be predictive of biological behaviour.
  - ▶ Such models need to be combined with biological data.
- ▶ Systems biology:

# Can a Biologist Fix a Radio? Lazebnik (2002)

## The Case for Systems Biology

*"It is difficult to find a black cat in a dark room, especially if there is no cat."*

- ▶ Biological systems are immensely complicated.
- ▶ Lazebnik argues the need for models that are quantitative.
  - ▶ Such models should be predictive of biological behaviour.
  - ▶ Such models need to be combined with biological data.
- ▶ Systems biology:
  - ▶ Build mechanistic models (based on biochemical knowledge) of the system.

# Can a Biologist Fix a Radio? Lazebnik (2002)

## The Case for Systems Biology

*"It is difficult to find a black cat in a dark room, especially if there is no cat."*

- ▶ Biological systems are immensely complicated.
- ▶ Lazebnik argues the need for models that are quantitative.
  - ▶ Such models should be predictive of biological behaviour.
  - ▶ Such models need to be combined with biological data.
- ▶ Systems biology:
  - ▶ Build mechanistic models (based on biochemical knowledge) of the system.
  - ▶ Identify modules, submodules, and parameterize the models.

# Coregulation of Gene Expression

## The Case for Computational Biology

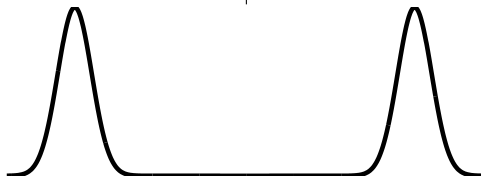
- ▶ Gene Expression to Transcriptional Regulation.
- ▶ A “data exploration” problem (computational biology/bioinformatics):
  - ▶ Use gene expression data to speculate on coregulated genes.
  - ▶ Traditionally use clustering of gene expression profiles.
- ▶ Contrast with (computational) systems biology approach:
  - ▶ Detailed mechanistic model of the system is created.
  - ▶ Fit parameters of the model to data.
  - ▶ Problematic for large data (genome wide).
  - ▶ Need to deal with unobserved biochemical species (TFs).

# Computational Biology vs Computational Systems Biology

Broadly Speaking: Two approaches to modeling

*data-modeling*

*mechanistic-modeling*



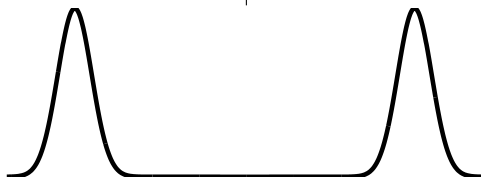
# Computational Biology vs Computational Systems Biology

Broadly Speaking: Two approaches to modeling

*data-modeling*

let the data “speak”

*mechanistic-modeling*





# Computational Biology vs Computational Systems Biology

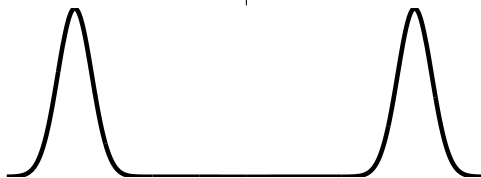
Broadly Speaking: Two approaches to modeling

*data-modeling*

let the data “speak”

*mechanistic-modeling*

impose physical laws



# Computational Biology vs Computational Systems Biology

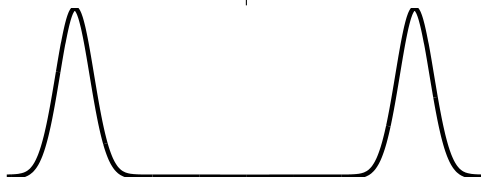
Broadly Speaking: Two approaches to modeling

*data-modeling*

let the data “speak”  
computational models

*mechanistic-modeling*

impose physical laws



# Computational Biology vs Computational Systems Biology

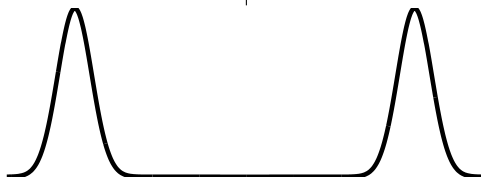
Broadly Speaking: Two approaches to modeling

*data-modeling*

let the data “speak”  
computational models

*mechanistic-modeling*

impose physical laws  
systems models



# Computational Biology vs Computational Systems Biology

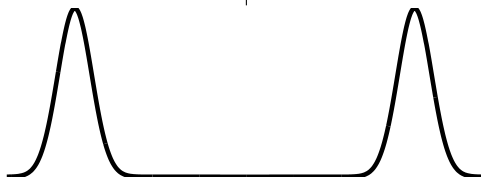
Broadly Speaking: Two approaches to modeling

*data-modeling*

let the data “speak”  
computational models  
adaptive models

*mechanistic-modeling*

impose physical laws  
systems models



# Computational Biology vs Computational Systems Biology

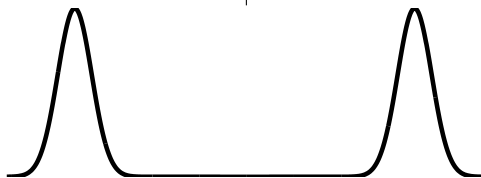
Broadly Speaking: Two approaches to modeling

*data-modeling*

let the data “speak”  
computational models  
adaptive models

*mechanistic-modeling*

impose physical laws  
systems models  
differential equations



# Computational Biology vs Computational Systems Biology

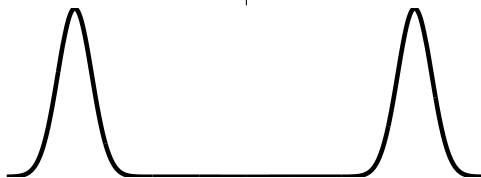
Broadly Speaking: Two approaches to modeling

*data-modeling*

let the data “speak”  
computational models  
adaptive models  
PCA, clustering

*mechanistic-modeling*

impose physical laws  
systems models  
differential equations



# Computational Biology vs Computational Systems Biology

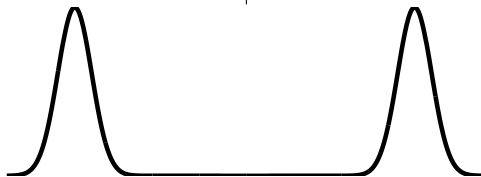
Broadly Speaking: Two approaches to modeling

*data-modeling*

let the data “speak”  
computational models  
adaptive models  
PCA, clustering

*mechanistic-modeling*

impose physical laws  
systems models  
differential equations  
SDE, ODE models



# Computational Biology vs Computational Systems Biology

Broadly Speaking: Two approaches to modeling

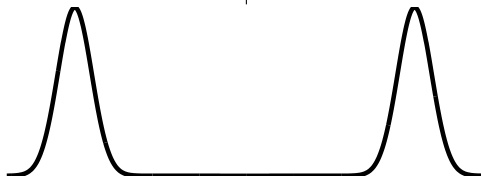
*data-modeling*

let the data "speak"  
computational models  
adaptive models  
PCA, clustering

Weakly Mechanistic

*mechanistic-modeling*

impose physical laws  
systems models  
differential equations  
SDE, ODE models





# Computational Biology vs Computational Systems Biology

Broadly Speaking: Two approaches to modeling

*data-modeling*

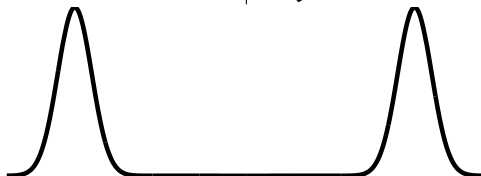
let the data "speak"  
computational models  
adaptive models  
PCA, clustering

Weakly Mechanistic

*mechanistic-modeling*

impose physical laws  
systems models  
differential equations  
SDE, ODE models

Strongly Mechanistic



# A Hybrid Approach

Introduce aspects of systems biology to computational models

- ▶ We advocate an approach *between* systems and computational biology.
- ▶ Introduce aspects of systems biology to the computational approach.
  - ▶ There is a computational penalty, but it may be worth paying.
  - ▶ Ideally there should be a smooth transition from pure computational (PCA, clustering, SVM classification) to systems (non-linear (stochastic) differential equations).
  - ▶ This work is one part of that transition.

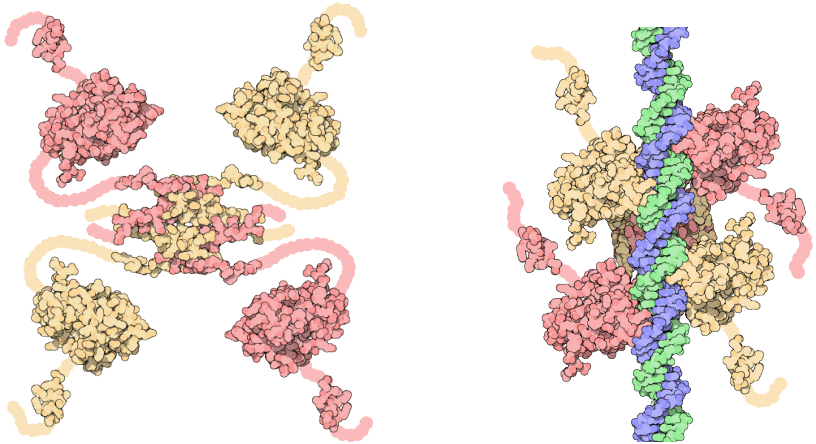
# Radiation Damage in the Cell

- ▶ Radiation can damage molecules including DNA.
- ▶ Most DNA damage is quickly repaired—single strand breaks, backbone break.
- ▶ Double strand breaks are more serious—a complete disconnect along the chromosome.
- ▶ Cell cycle stages:
  - ▶  $G_1$ : Cell is not dividing.
  - ▶  $G_2$ : Cell is preparing for meiosis, chromosomes have divided.
  - ▶ S: Cell is undergoing meiosis (DNA synthesis).
- ▶ Main problem is in  $G_1$ . In  $G_2$  there are two copies of the chromosome. In  $G_1$  only one copy.

## p53 “Guardian of the Cell”

- ▶ Responsible for Repairing DNA damage
- ▶ Activates DNA Repair proteins
- ▶ Pauses the Cell Cycle (prevents replication of damage DNA)
- ▶ Initiates *apoptosis* (cell death) in the case where damage can't be repaired.
- ▶ Large scale feedback loop with NF- $\kappa$ B.

# p53 DNA Damage Repair



**Figure :** p53. *Left unbound, Right bound to DNA.* Images by David S. Goodsell from <http://www.rcsb.org/> (see the "Molecule of the Month" feature).

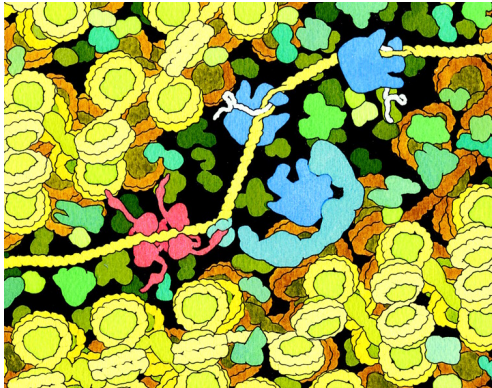


Figure : Repair of DNA damage by p53. Image from Goodsell (1999).

## Some p53 Targets

*DDB2* DNA Damage Specific DNA Binding Protein 2.  
(also governed by C/EBP-beta, E2F1, E2F3,...).

*p21* Cycline-dependent kinase inhibitor 1A  
(CDKN1A). A regulator of cell cycle progression.  
(also governed by SREBP-1a, Sp1, Sp3,... ).

*hPA26/SESN1* sestrin 1 Cell Cycle arrest.

*BIK* BCL2-interacting killer. Induces cell death  
(apoptosis)

*TNFRSF10b* tumor necrosis factor receptor superfamily,  
member 10b. A transducer of apoptosis signals.

# Modelling Assumption

- Assume p53 affects targets as a single input module network motif (SIM).

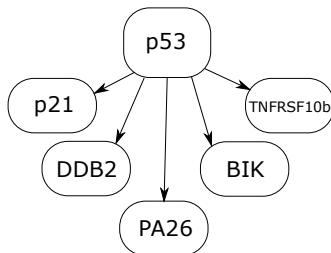


Figure : p53 SIM network motif as modelled by Barenco et al. 2006.



# Standard Approach

## Clustering of Gene Expression Profiles

- ▶ Assume that coregulated genes will cluster in the same groups.
- ▶ Perform clustering, and look for clusters containing target genes.
- ▶ These are candidates, look for confirmation in the literature etc.

Method

Open Access

## Ranked prediction of p53 targets using hidden variable dynamic modeling

Martino Barenco<sup>\*†</sup>, Daniela Tomescu<sup>\*</sup>, Daniel Brewer<sup>\*†</sup>, Robin Callard<sup>\*†</sup>, Jaroslav Stark<sup>†‡</sup> and Michael Hubank<sup>\*†</sup>

Addresses: <sup>\*</sup>Institute of Child Health, University College London, Guilford Street, London WC1N 1EH, UK. <sup>†</sup>CoMPLEX (Centre for Mathematics and Physics in the Life Sciences and Experimental Biology), University College London, Stephenson Way, London, NW1 2HE, UK. <sup>‡</sup>Department of Mathematics, Imperial College London, London SW7 2AZ, UK.

Correspondence: Michael Hubank. Email: m.hubank@ich.ucl.ac.uk

Published: 31 March 2006

*Genome Biology* 2006, **7**:R25 (doi:10.1186/gb-2006-7-3-r25)

Received: 24 November 2005

Revised: 30 January 2006

Accepted: 21 February 2006

# Transcription Model

mRNA  
production  
rate

# Transcription Model

$$\text{mRNA production rate} = \text{base rate}$$

# Transcription Model

$$\begin{array}{c} \text{mRNA} \\ \text{production} \\ \text{rate} \end{array} = \begin{array}{c} \text{base} \\ \text{rate} \end{array} + \begin{array}{c} \text{TF ac-} \\ \text{tivity} \end{array}$$

# Transcription Model

$$\begin{array}{c} \text{mRNA} \\ \text{production} \\ \text{rate} \end{array} = \begin{array}{c} \text{base} \\ \text{rate} \end{array} + \begin{array}{c} \text{TF ac-} \\ \text{tivity} \end{array} - \begin{array}{c} \text{mRNA} \\ \text{decay} \end{array}$$

# Mathematical Model

- Differential equation model of system.

rate of mRNA transcription, baseline transcription rate,  
transcription factor activity, mRNA decay

- We have observations of  $m_j(t)$  from gene expression.

# Mathematical Model

- Differential equation model of system.

$$\frac{dm_j(t)}{dt} = b_j + s_j p(t) - d_j m_j(t)$$

rate of mRNA transcription, baseline transcription rate,  
transcription factor activity, mRNA decay

- We have observations of  $m_j(t)$  from gene expression.
- Reorder differential equation.



# Mathematical Model

- Differential equation model of system.

$$\frac{dm_j(t)}{dt} = b_j + s_j p(t) - d_j m_j(t)$$
$$d_j m_j(t) + \frac{dm_j(t)}{dt} = b_j + s_j p(t)$$

rate of mRNA transcription, baseline transcription rate,  
transcription factor activity, mRNA decay

- We have observations of  $m_j(t)$  from gene expression.
- Reorder differential equation.
- An estimate of  $\frac{dm_j(t)}{dt}$  is obtained through fitting polynomials.

# Mathematical Model

- Differential equation model of system.

$$\frac{dm_j(t)}{dt} = b_j + s_j p(t) - d_j m_j(t)$$
$$d_j m_j(t) + \frac{dm_j(t)}{dt} = b_j + s_j p(t)$$

rate of mRNA transcription, baseline transcription rate,  
transcription factor activity, mRNA decay

- We have observations of  $m_j(t)$  from gene expression.
- Reorder differential equation.
- An estimate of  $\frac{dm_j(t)}{dt}$  is obtained through fitting polynomials.
- Jointly estimate  $p(t)$  at observations of time points along with  $\{b_j, d_j, s_j\}_{j=1}^g$ .

# Mathematical Model

- Differential equation model of system.

$$\frac{dm_j(t)}{dt} = b_j + s_j p(t) - d_j m_j(t)$$
$$d_j m_j(t) + \frac{dm_j(t)}{dt} = b_j + s_j p(t)$$

rate of mRNA transcription, baseline transcription rate,  
transcription factor activity, mRNA decay

- We have observations of  $m_j(t)$  from gene expression.
- Reorder differential equation.
- An estimate of  $\frac{dm_j(t)}{dt}$  is obtained through fitting polynomials.
- Jointly estimate  $p(t)$  at observations of time points along with  $\{b_j, d_j, s_j\}_{j=1}^g$ .
- Fit parameters by maximum likelihood or MCMC sampling.

# Mathematical Model

- ▶ Clustering model is equivalent to assuming  $d_j$ ,  $b_j$ , and  $s_j$  are v. large.

$$\frac{dm_j(t)}{dt} = b_j + s_j p(t) - d_j m_j(t)$$

rate of mRNA transcription, baseline transcription rate,  
transcription factor activity, mRNA decay

- ▶ We have observations of  $m_j(t)$  from gene expression.

# Mathematical Model

- ▶ Clustering model is equivalent to assuming  $d_j$ ,  $b_j$ , and  $s_j$  are v. large.

$$\frac{dm_j(t)}{dt} = b_j + s_j p(t) - d_j m_j(t)$$
$$d_j m_j(t) \approx b_j + s_j p(t)$$

rate of mRNA transcription, baseline transcription rate,  
transcription factor activity, mRNA decay

- ▶ We have observations of  $m_j(t)$  from gene expression.
- ▶ Reorder differential equation and ignore gradient term.

# Mathematical Model

- ▶ Clustering model is equivalent to assuming  $d_j$ ,  $b_j$ , and  $s_j$  are v. large.

$$\frac{dm_j(t)}{dt} = b_j + s_j p(t) - d_j m_j(t)$$
$$d_j m_j(t) \approx b_j + s_j p(t)$$

rate of mRNA transcription, baseline transcription rate,  
transcription factor activity, mRNA decay

- ▶ We have observations of  $m_j(t)$  from gene expression.
- ▶ Reorder differential equation and ignore gradient term.
- ▶ This suggests genes are scaled and offset versions of the TF.

# Mathematical Model

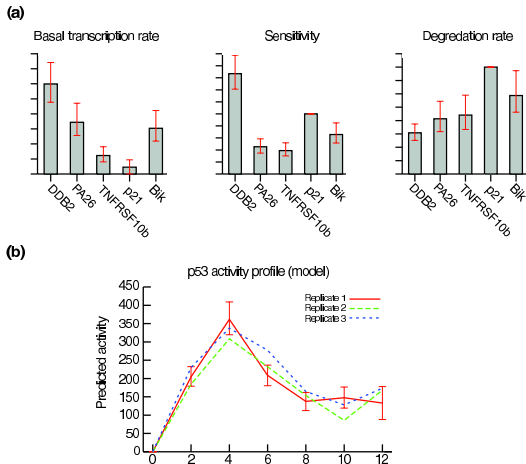
- ▶ Clustering model is equivalent to assuming  $d_j$ ,  $b_j$ , and  $s_j$  are v. large.

$$\frac{dm_j(t)}{dt} = b_j + s_j p(t) - d_j m_j(t)$$
$$d_j m_j(t) \approx b_j + s_j p(t)$$

rate of mRNA transcription, baseline transcription rate,  
transcription factor activity, mRNA decay

- ▶ We have observations of  $m_j(t)$  from gene expression.
- ▶ Reorder differential equation and ignore gradient term.
- ▶ This suggests genes are scaled and offset versions of the TF.
- ▶ By normalizing data and clustering we hope to find those TFs.

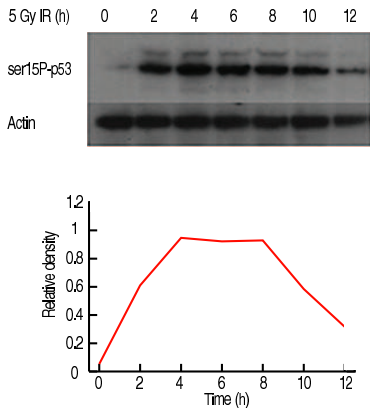
# Response of p53



**Figure :** Results from Barenco et al. (2006). Top is parameter estimates. Bottom is inferred profile.



# Response to p53 ...



**Figure :** Results from Barenco et al. (2006). Activity profile of p53 was measured by Western blot to determine the levels of ser-15 phosphorylated p53 (ser15P-p53).

# Example: Transcriptional Regulation

- ▶ First Order Differential Equation

$$\frac{dm_j(t)}{dt} = b_j + s_j p(t) - d_j m_j(t)$$

- ▶ It turns out that our Gaussian process assumption for  $p(t)$ , implies  $m(t)$  is also a Gaussian process.
- ▶ The new Gaussian process is over  $p(t)$  and all its targets:  $m_1(t), m_2(t), \dots$  etc.
- ▶ Our new covariance matrix gives correlations between all these functions.
- ▶ This gives us a *probabilistic* model for transcriptional regulation.

## Example: Transcriptional Regulation

- ▶ First Order Differential Equation

$$\frac{dm_j(t)}{dt} = b_j + s_j p(t) - d_j m_j(t)$$

- ▶ It turns out that our Gaussian process assumption for  $p(t)$ , implies  $m(t)$  is also a Gaussian process.
- ▶ The new Gaussian process is over  $p(t)$  and all its targets:  $m_1(t), m_2(t), \dots$  etc.
- ▶ Our new covariance matrix gives correlations between all these functions.
- ▶ This gives us a *probabilistic* model for transcriptional regulation.

## Example: Transcriptional Regulation

- ▶ First Order Differential Equation

$$\frac{dm_j(t)}{dt} = b_j + s_j p(t) - d_j m_j(t)$$

- ▶ It turns out that our Gaussian process assumption for  $p(t)$ , implies  $m(t)$  is also a Gaussian process.
- ▶ The new Gaussian process is over  $p(t)$  and all its targets:  $m_1(t), m_2(t), \dots$  etc.
- ▶ Our new covariance matrix gives correlations between all these functions.
- ▶ This gives us a *probabilistic* model for transcriptional regulation.

## Example: Transcriptional Regulation

- ▶ First Order Differential Equation

$$\frac{dm_j(t)}{dt} = b_j + s_j p(t) - d_j m_j(t)$$

- ▶ It turns out that our Gaussian process assumption for  $p(t)$ , implies  $m(t)$  is also a Gaussian process.
- ▶ The new Gaussian process is over  $p(t)$  and all its targets:  $m_1(t), m_2(t), \dots$  etc.
- ▶ Our new covariance matrix gives correlations between all these functions.
- ▶ This gives us a *probabilistic* model for transcriptional regulation.

## Example: Transcriptional Regulation

- ▶ First Order Differential Equation

$$\frac{dm_j(t)}{dt} = b_j + s_j p(t) - d_j m_j(t)$$

- ▶ It turns out that our Gaussian process assumption for  $p(t)$ , implies  $m(t)$  is also a Gaussian process.
- ▶ The new Gaussian process is over  $p(t)$  and all its targets:  $m_1(t), m_2(t), \dots$  etc.
- ▶ Our new covariance matrix gives correlations between all these functions.
- ▶ This gives us a *probabilistic* model for transcriptional regulation.

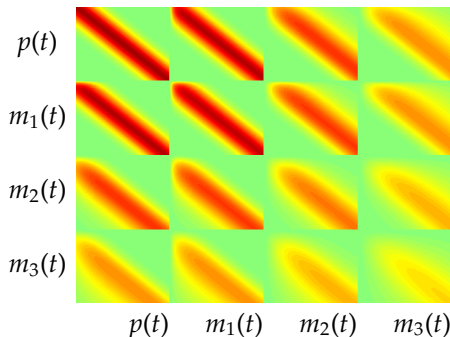
# Covariance for Transcription Model

## RBF covariance function for $p(t)$

$$m_i(t) = \frac{b_i}{d_i} + s_i \exp(-d_i t) \int_0^t p(u) \exp(d_i u) du.$$

- ▶ Joint distribution for  $m_1(t)$ ,  $m_2(t)$ ,  $m_3(t)$ , and  $p(t)$ .
- ▶ Here:

$d_1$	$s_1$	$d_2$	$s_2$	$d_3$	$s_3$
5	5	1	1	0.5	0.5



# Covariance for Transcription Model

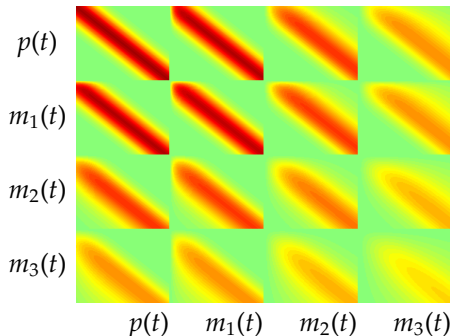
**RBF covariance function for  $p(t)$**

$$m = b/d + \sum_i \mathbf{e}_i^\top \mathbf{p} \quad \mathbf{p} \sim \mathcal{N}(\mathbf{0}, \Sigma_i) \rightarrow m \sim \mathcal{N}\left(b/d, \sum_i \mathbf{e}_i^\top \Sigma_i \mathbf{e}_i\right)$$

- ▶ Joint distribution for  $m_1(t)$ ,  $m_2(t)$ ,  $m_3(t)$ , and  $p(t)$ .

- ▶ Here:

$d_1$	$s_1$	$d_2$	$s_2$	$d_3$	$s_3$
5	5	1	1	0.5	0.5





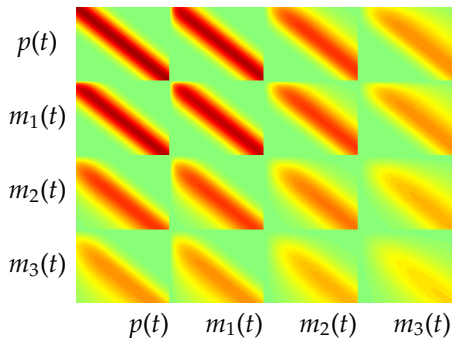
# Covariance for Transcription Model

## RBF covariance function for $p(t)$

$$m_i(t) = \frac{b_i}{d_i} + s_i \exp(-d_i t) \int_0^t p(u) \exp(d_i u) du.$$

- ▶ Joint distribution for  $m_1(t)$ ,  $m_2(t)$ ,  $m_3(t)$ , and  $p(t)$ .
- ▶ Here:

$d_1$	$s_1$	$d_2$	$s_2$	$d_3$	$s_3$
5	5	1	1	0.5	0.5



# Joint Sampling of $p(t)$ and $m(t)$

► `simSample`

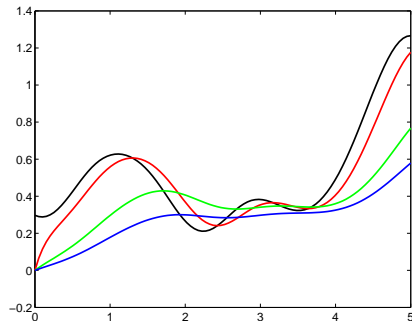


Figure : Joint samples from the ODE covariance, *black*:  $p(t)$ , *red*:  $m_1(t)$  (high decay/sensitivity), *green*:  $m_2(t)$  (medium decay/sensitivity) and *blue*:  $m_3(t)$  (low decay/sensitivity).

# Joint Sampling of $p(t)$ and $m(t)$

► `simSample`

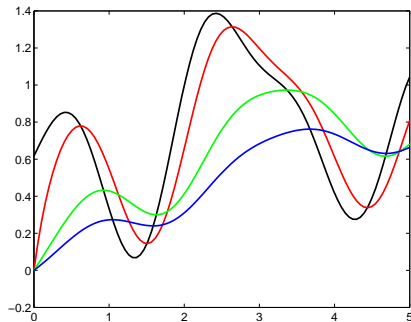


Figure : Joint samples from the ODE covariance, *black*:  $p(t)$ , *red*:  $m_1(t)$  (high decay/sensitivity), *green*:  $m_2(t)$  (medium decay/sensitivity) and *blue*:  $m_3(t)$  (low decay/sensitivity).

# Joint Sampling of $p(t)$ and $m(t)$

## ► `simSample`

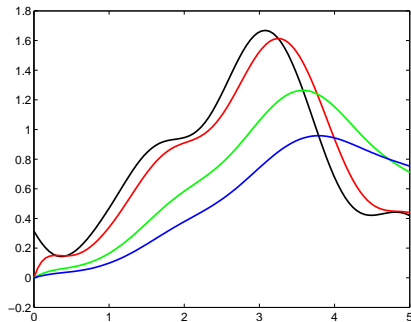


Figure : Joint samples from the ODE covariance, *black*:  $p(t)$ , *red*:  $m_1(t)$  (high decay/sensitivity), *green*:  $m_2(t)$  (medium decay/sensitivity) and *blue*:  $m_3(t)$  (low decay/sensitivity).

# Joint Sampling of $p(t)$ and $m(t)$

► `simSample`

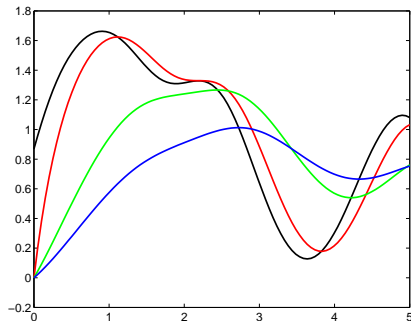
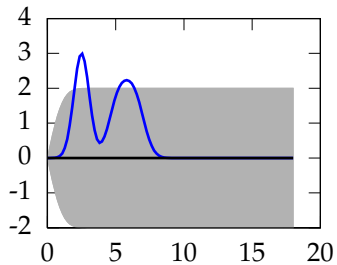
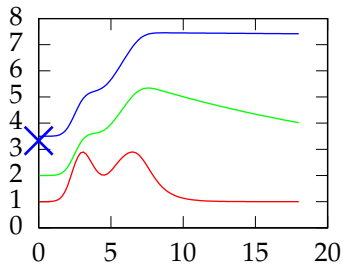
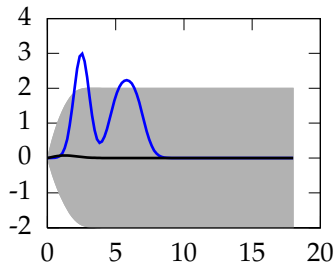
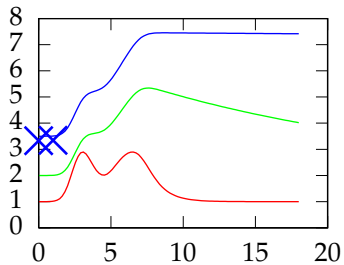
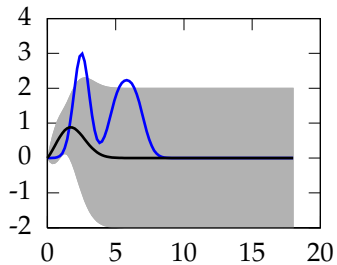
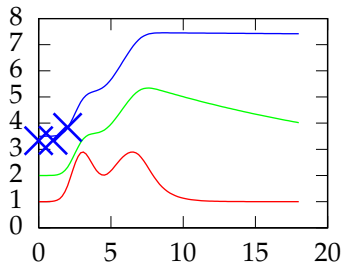


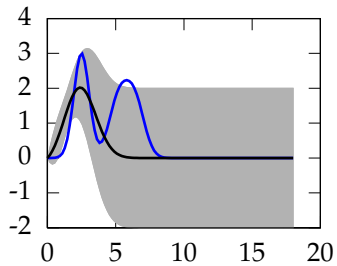
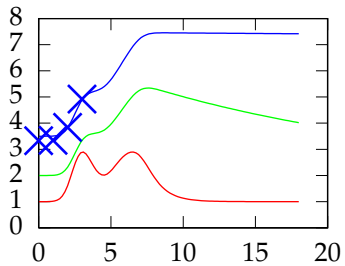
Figure : Joint samples from the ODE covariance, *black*:  $p(t)$ , *red*:  $m_1(t)$  (high decay/sensitivity), *green*:  $m_2(t)$  (medium decay/sensitivity) and *blue*:  $m_3(t)$  (low decay/sensitivity).

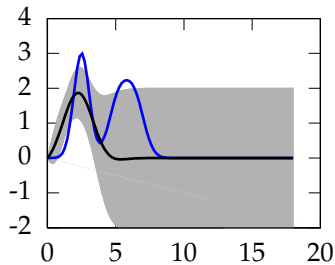
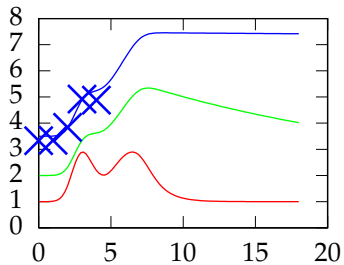


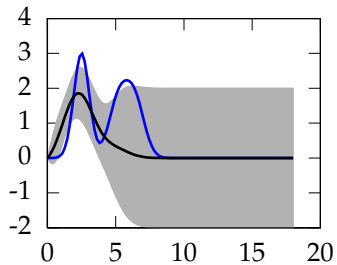
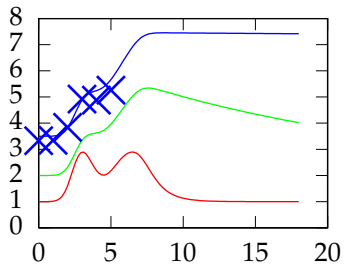


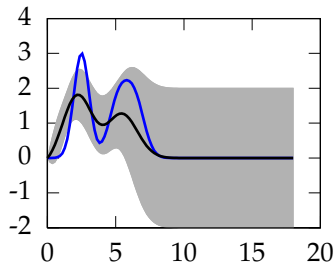
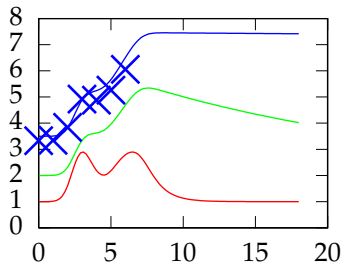


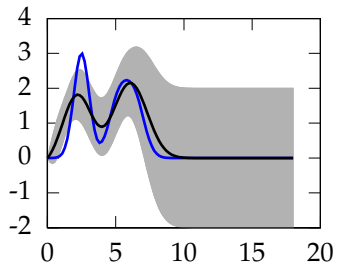
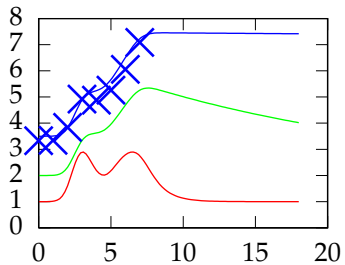


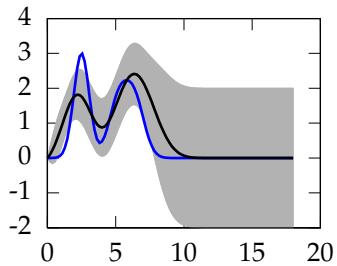
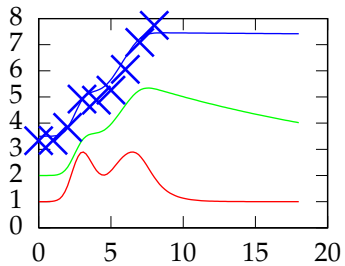


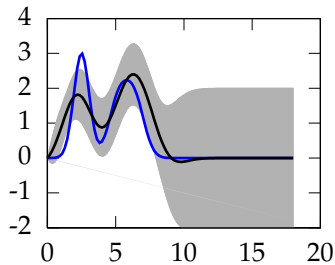
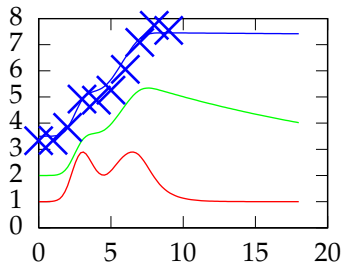


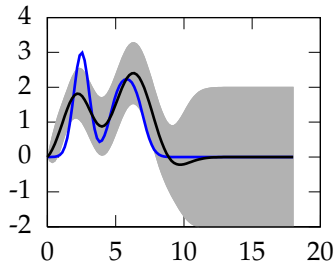
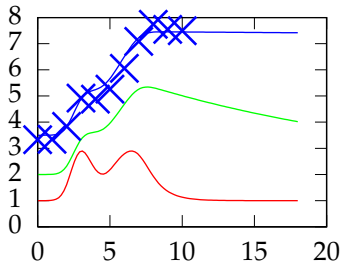




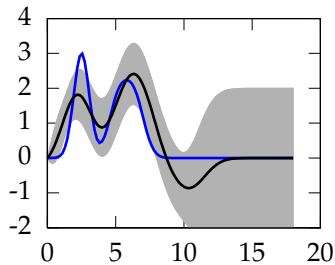
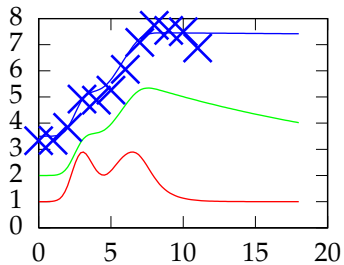


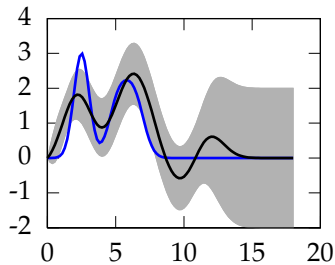
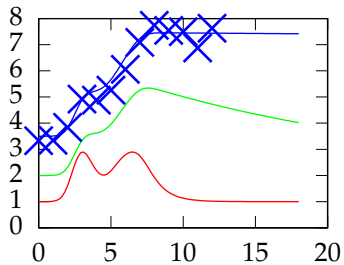


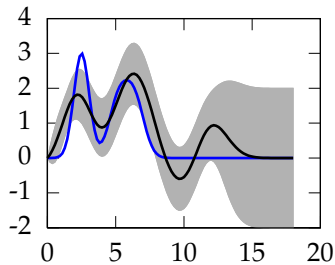
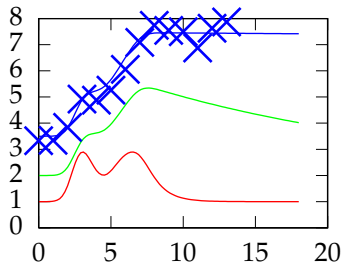


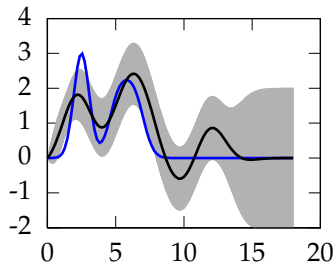
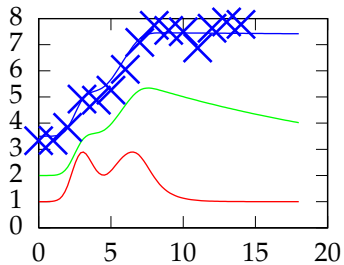


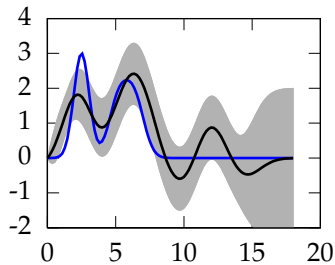
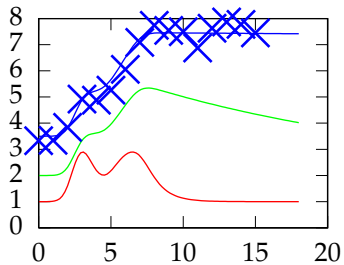


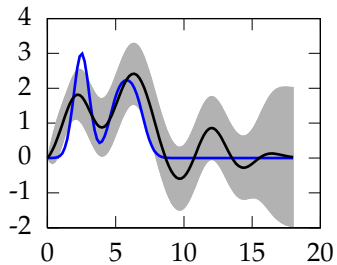
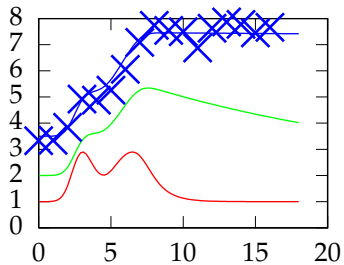


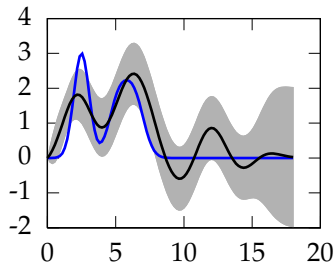
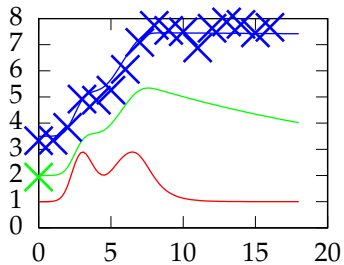


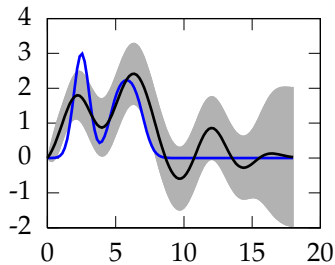
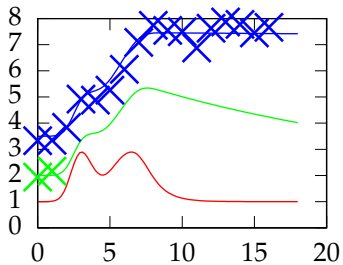




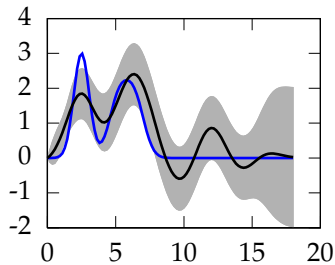
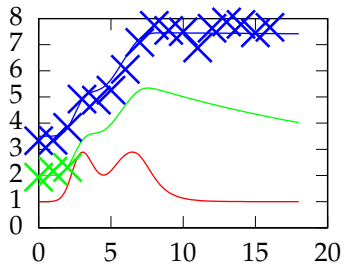


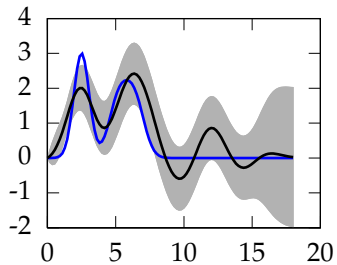
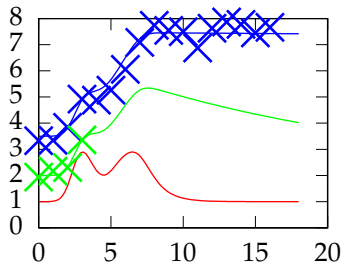


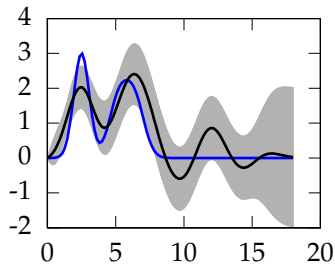
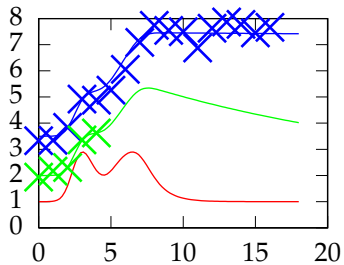


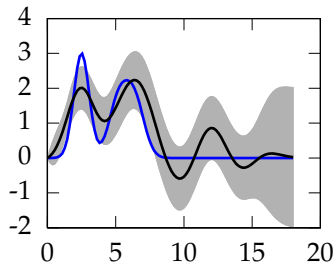
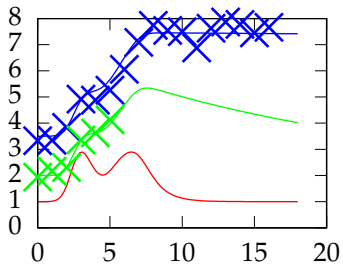


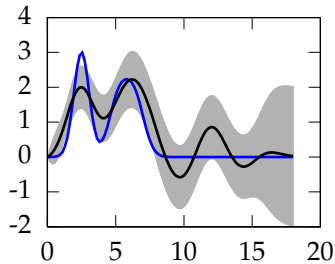
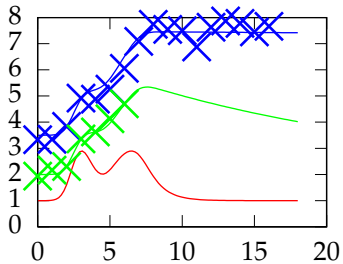


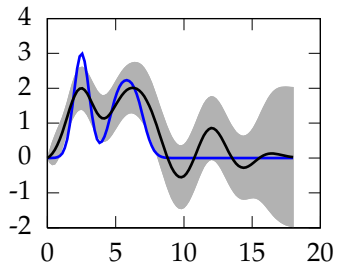
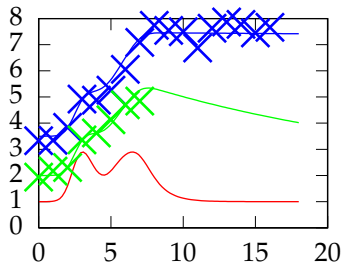


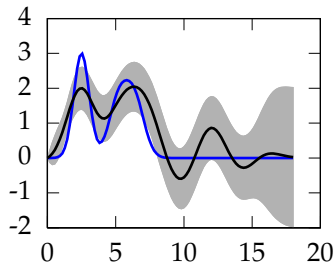
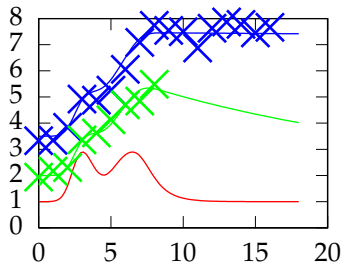


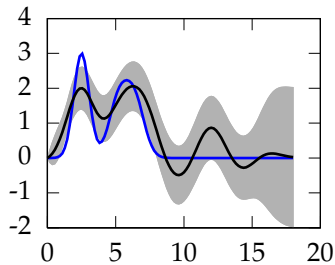
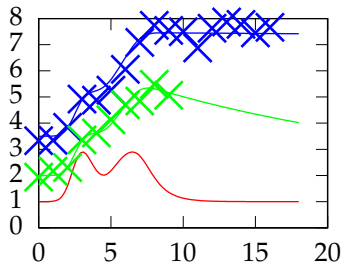




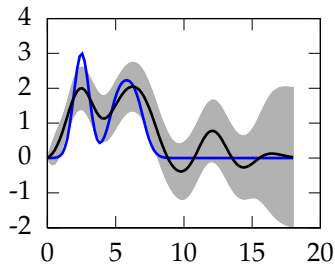
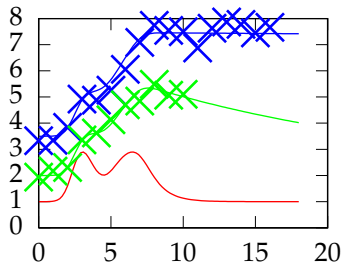


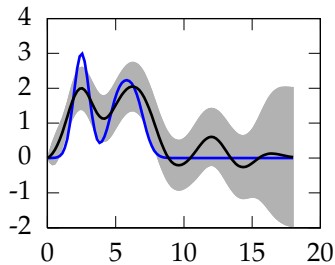
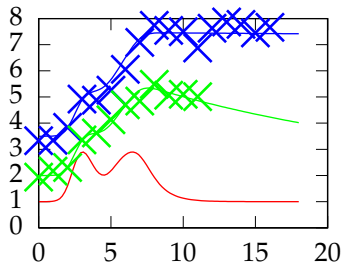


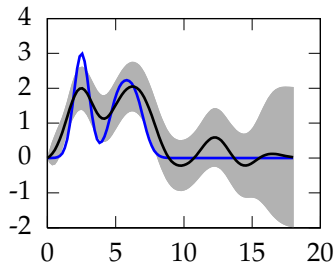
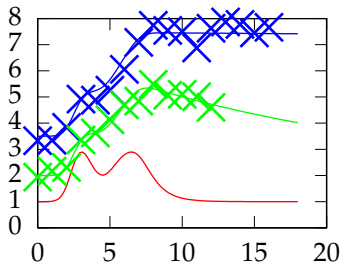


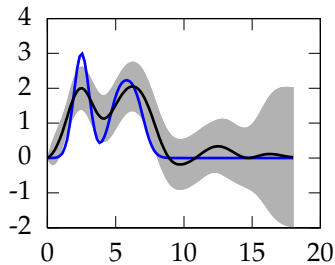
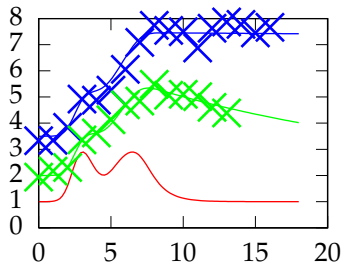


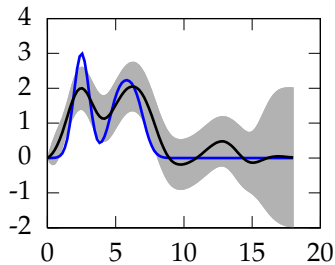
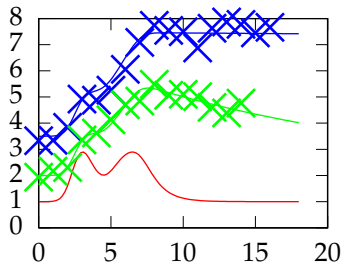


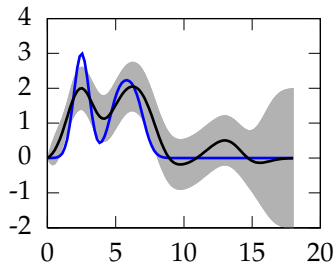
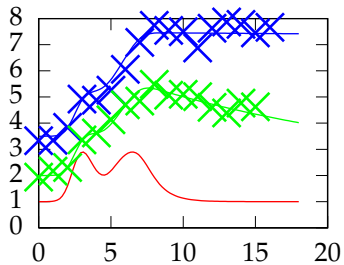


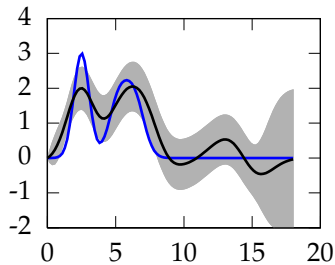
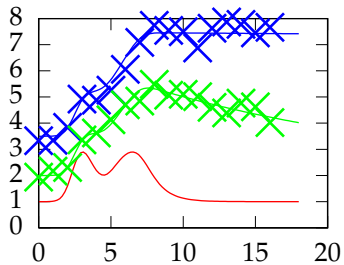


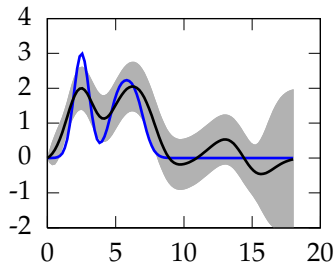
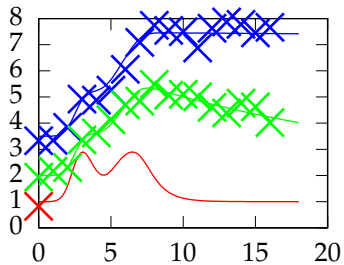




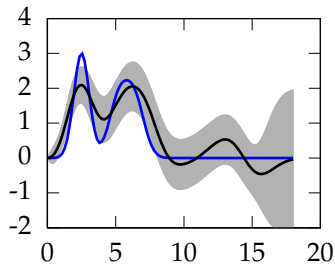
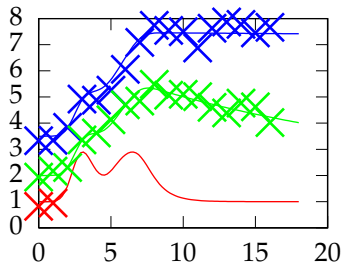


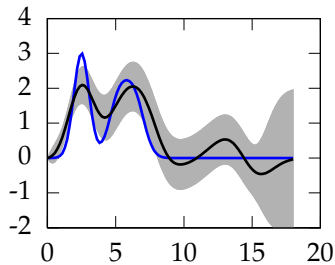
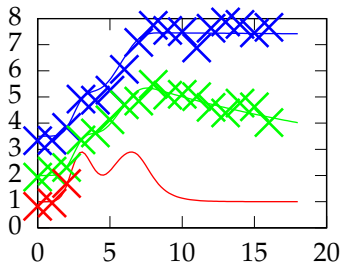


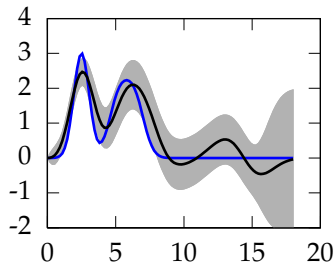
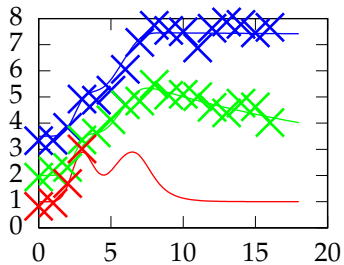


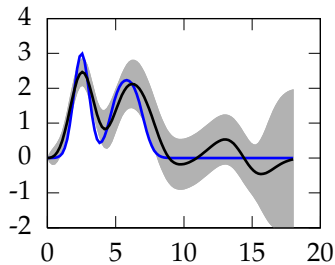
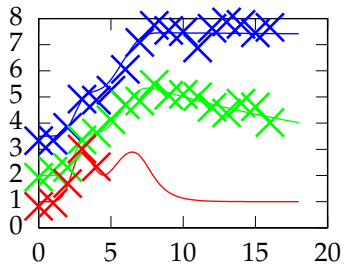


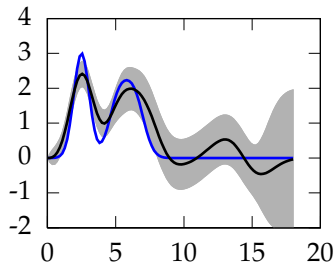
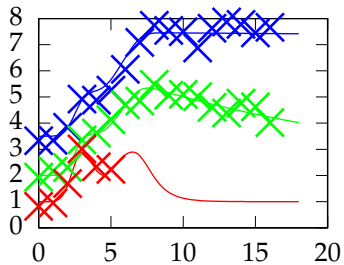


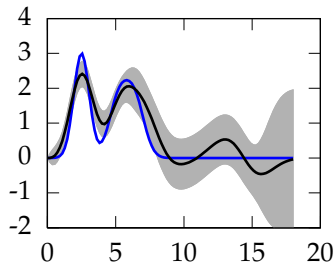
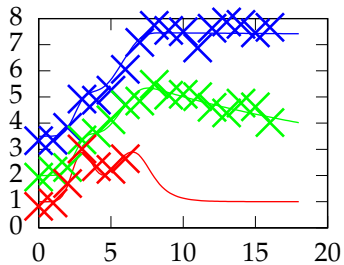


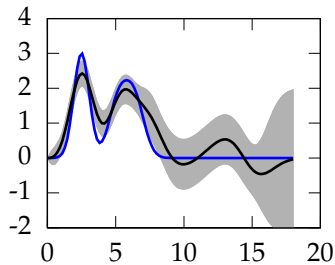
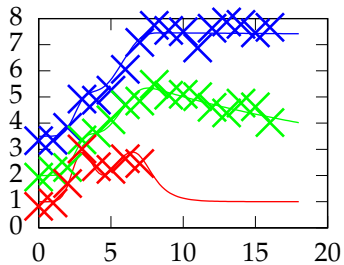


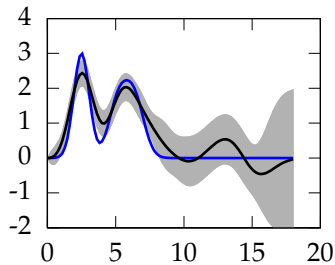
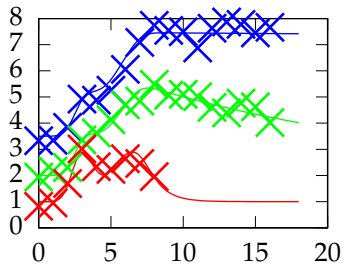




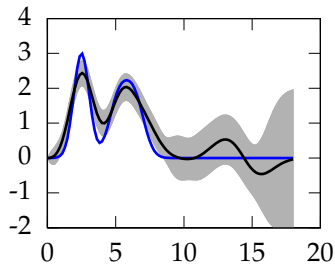
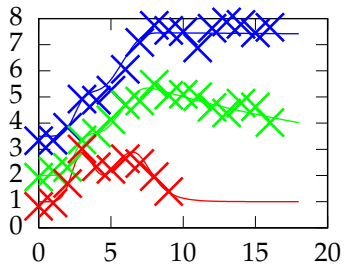


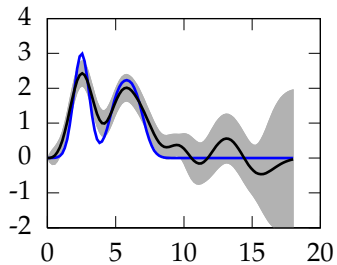
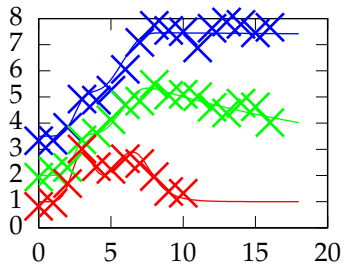


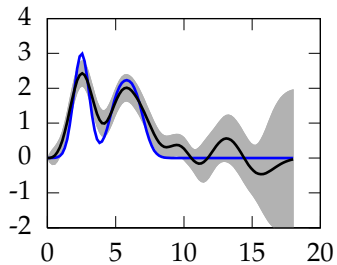
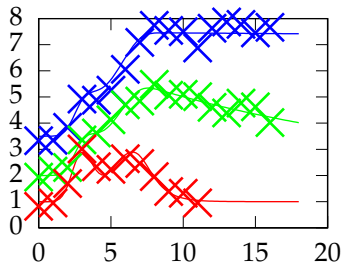


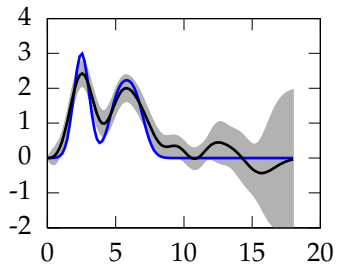
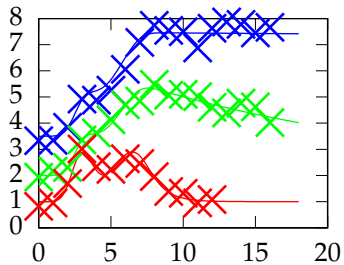


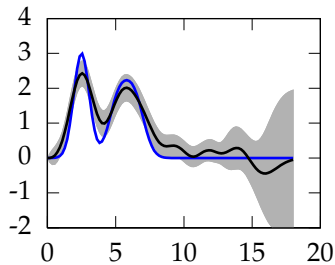
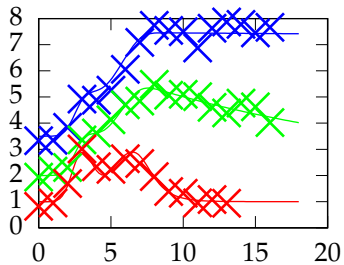


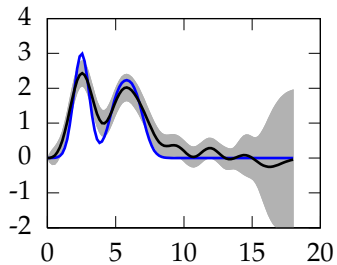
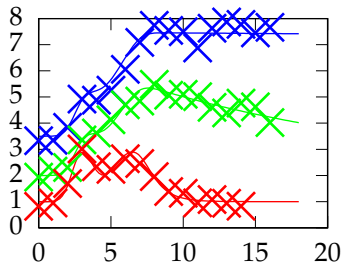


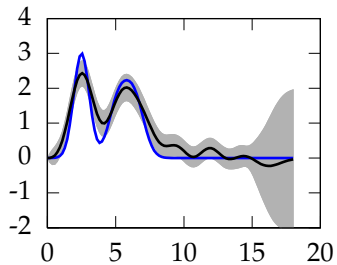
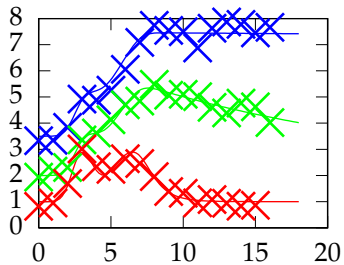


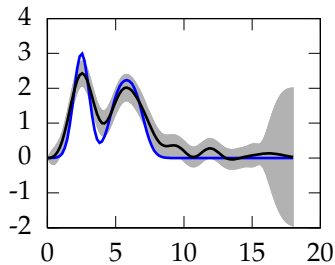
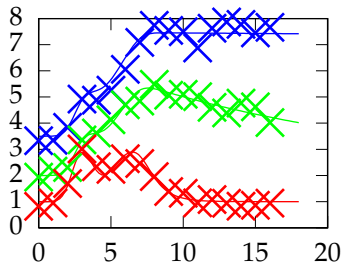














# Gene Expression Example

- ▶ TIGRE Bioconductor package.
- ▶ <http://www.bioconductor.org/packages/2.6/bioc/html/tigre.html> (Antti Honkela is the maintainer).

---

## Gaussian process modelling of latent chemical species: applications to inferring transcription factor activities

Pei Gao<sup>1</sup>, Antti Honkela<sup>2</sup>, Magnus Rattray<sup>1</sup> and Neil D. Lawrence<sup>1,\*</sup>

<sup>1</sup>School of Computer Science, University of Manchester, Kilburn Building, Oxford Road, Manchester, M13 9PL and

<sup>2</sup>Adaptive Informatics Research Centre, Helsinki University of Technology, PO Box 5400, FI-02015 TKK, Finland

---

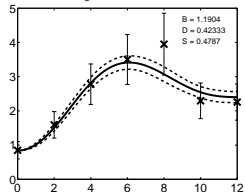
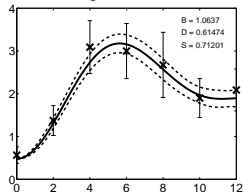
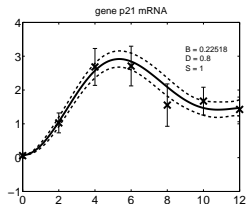
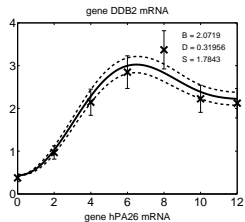
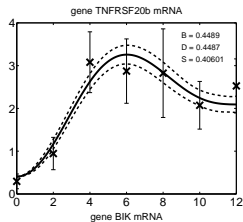
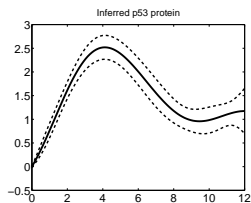
### ABSTRACT

**Motivation:** Inference of *latent chemical species* in biochemical interaction networks is a key problem in estimation of the structure

A challenging problem for parameter estimation in ODE models occurs where one or more chemical species influencing the dynamics are controlled outside of the sub-system being modelled. For

# p53 Results with GP

(Gao et al., 2008)

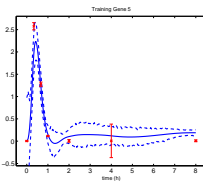
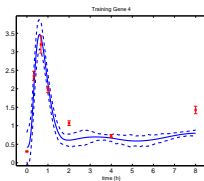
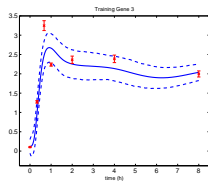
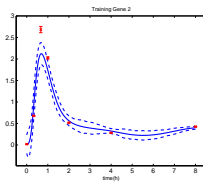
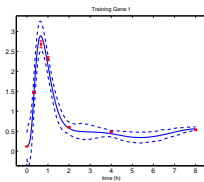
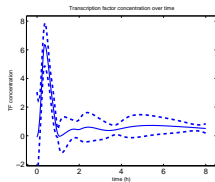


# Ranking with ERK Signalling

- ▶ 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.

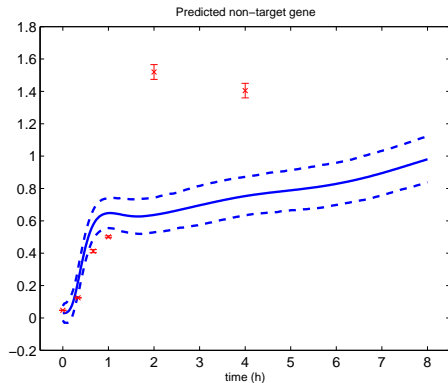
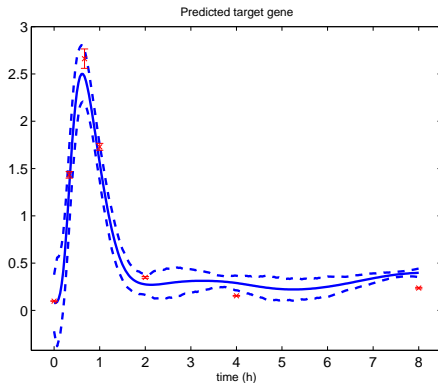
# Elk-1 (MLP covariance)

Jennifer Withers



# Elk-1 target selection

Fitted model used to rank potential targets of Elk-1



# Outline

Multivariate Gaussian Properties

**Cascade Differential Equations**

Multiple Transcription Factors

Conclusions

## Model-based method for transcription factor target identification with limited data

Antti Honkela<sup>a,1</sup>, Charles Girardot<sup>b</sup>, E. Hilary Gustafson<sup>b</sup>, Ya-Hsin Liu<sup>b</sup>, Eileen E. M. Furlong<sup>b</sup>, Neil D. Lawrence<sup>c,1</sup>, and Magnus Rattray<sup>c,1</sup>

<sup>a</sup>Department of Information and Computer Science, Aalto University School of Science and Technology, Helsinki, Finland; <sup>b</sup>Genome Biology U European Molecular Biology Laboratory, Heidelberg, Germany; and <sup>c</sup>School of Computer Science, University of Manchester, Manchester, United Kingdom

Edited by David Baker, University of Washington, Seattle, WA, and approved March 3, 2010 (received for review December 10, 2009)

**We present a computational method for identifying potential targets of a transcription factor (TF) using wild-type gene expression time series data. For each putative target gene we fit a simple differential equation model of transcriptional regulation, and the**

**used for genome-wide scoring of putative target genes. The method is required to apply our method is wild-type time series data collected over a period where TF activity is changing. Our method allows for complementary evidence from expression**



# Cascaded Differential Equations

**(Honkela et al., 2010)**

- ▶ 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*

## **Collaboration with Furlong Lab in EMBL Heidelberg.**

- ▶ Mesoderm development in *Drosophila melanogaster* (fruit fly).
- ▶ Mesoderm forms in triploblastic animals (along with ectoderm and endoderm). Mesoderm develops into muscles, and circulatory system.
- ▶ The transcription factor Twist initiates *Drosophila* mesoderm development, resulting in the formation of heart, somatic muscle, and other cell types.
- ▶ Wildtype microarray experiments publicly available.
- ▶ Can we use the cascade model to predict viable targets of Twist?

# Cascaded Differential Equations

(Honkela et al., 2010)

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

$$\begin{aligned}\frac{dp(t)}{dt} &= \sigma f(t) - \delta p(t) \\ \frac{dm_j(t)}{dt} &= b_j + s_j p(t) - d_j m_j(t)\end{aligned}$$

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

$$p(t) = \sigma \exp(-\delta t) \int_0^t f(u) \exp(\delta u) du .$$

# Covariance for Translation/Transcription Model

## RBF covariance function for $f(t)$

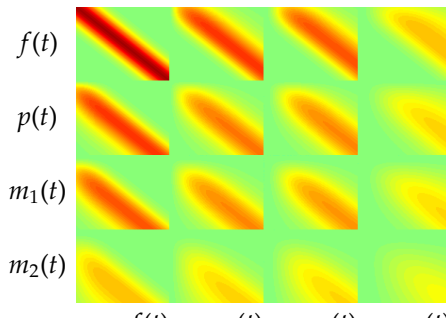
$$p(t) = \sigma \exp(-\delta t) \int_0^t f(u) \exp(\delta u) du$$

$$m_i(t) = \frac{b_i}{d_i} + s_i \exp(-d_i t) \int_0^t p(u) \exp(d_i u) du.$$

- ▶ Joint distribution for  $m_1(t)$ ,  $m_2(t)$ ,  $p(t)$  and  $f(t)$ .

- ▶ Here:

$\delta$	$d_1$	$s_1$	$d_2$	$s_2$
1	5	5	0.5	0.5



# Joint Sampling of $f(t)$ , $p(t)$ , and $m(t)$

## ► `disimSample`

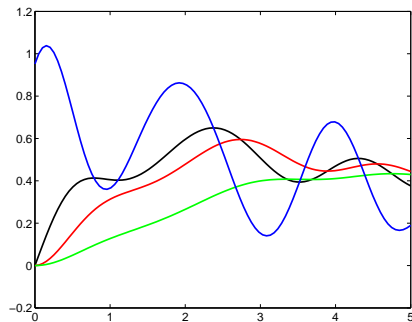


Figure : Joint samples from the ODE covariance, *blue*:  $f(t)$  (mRNA of TF), *black*:  $p(t)$  (TF concentration), *red*:  $m_1(t)$  (high decay target) and *green*:  $m_2(t)$  (low decay target)

# Joint Sampling of $f(t)$ , $p(t)$ , and $m(t)$

## ► `disimSample`

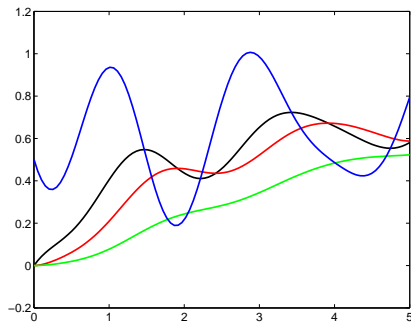


Figure : Joint samples from the ODE covariance, *blue*:  $f(t)$  (mRNA of TF), *black*:  $p(t)$  (TF concentration), *red*:  $m_1(t)$  (high decay target) and *green*:  $m_2(t)$  (low decay target)

# Joint Sampling of $f(t)$ , $p(t)$ , and $m(t)$

## ► `disimSample`

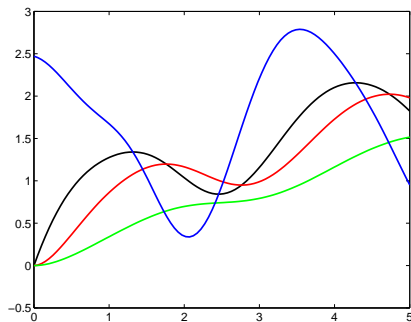


Figure : Joint samples from the ODE covariance, *blue*:  $f(t)$  (mRNA of TF), *black*:  $p(t)$  (TF concentration), *red*:  $m_1(t)$  (high decay target) and *green*:  $m_2(t)$  (low decay target)

# Joint Sampling of $f(t)$ , $p(t)$ , and $m(t)$

## ► `disimSample`

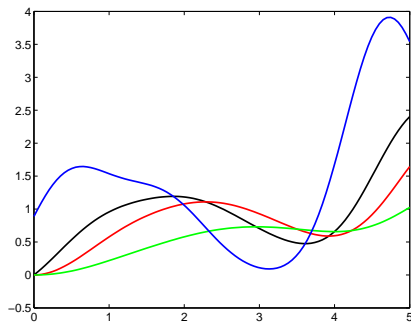


Figure : Joint samples from the ODE covariance, *blue*:  $f(t)$  (mRNA of TF), *black*:  $p(t)$  (TF concentration), *red*:  $m_1(t)$  (high decay target) and *green*:  $m_2(t)$  (low decay target)



# Twist Results

- ▶ Use mRNA of Twist as driving input.
- ▶ For each gene build a cascade model that forces Twist to be the only TF.
- ▶ Compare fit of this model to a baseline (*e.g.* similar model but sensitivity zero).
- ▶ Rank according to the likelihood above the baseline.
- ▶ Compare with correlation, knockouts and time series network identification (TSNI) (Della Gatta et al., 2008).

# Results for Twi using the Cascade model

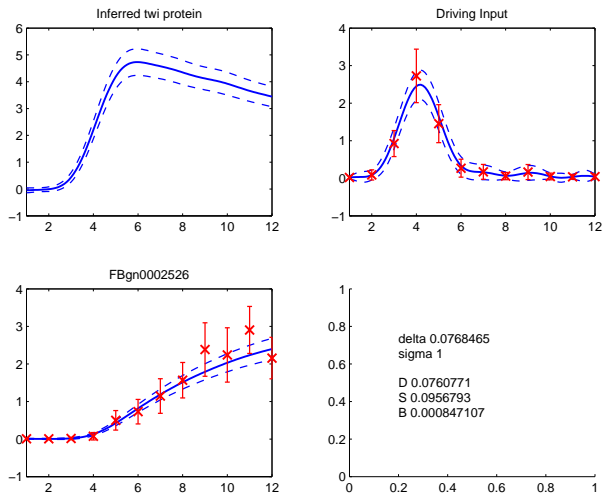


Figure : Model for flybase gene identity FBgn0002526.

# Results for Twi using the Cascade model

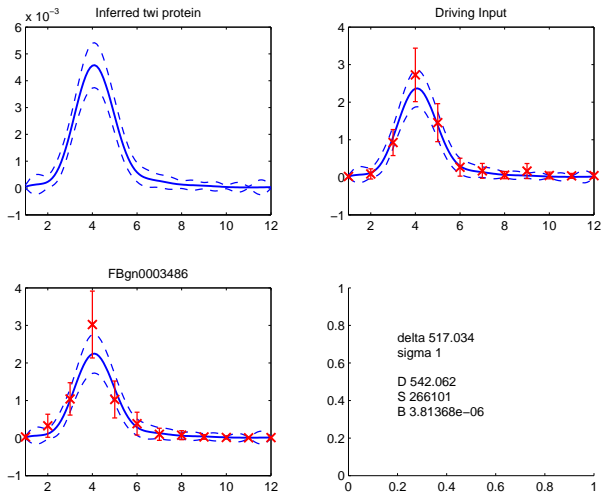


Figure : Model for flybase gene identity FBgn0003486.

# Results for Twi using the Cascade model

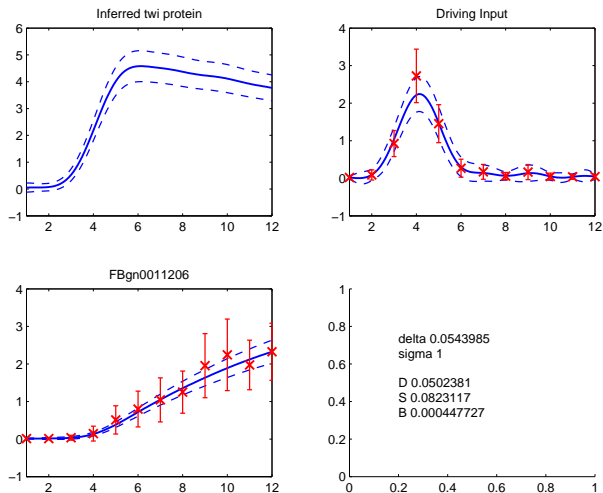


Figure : Model for flybase gene identity FBgn0011206.

# Results for Twi using the Cascade model

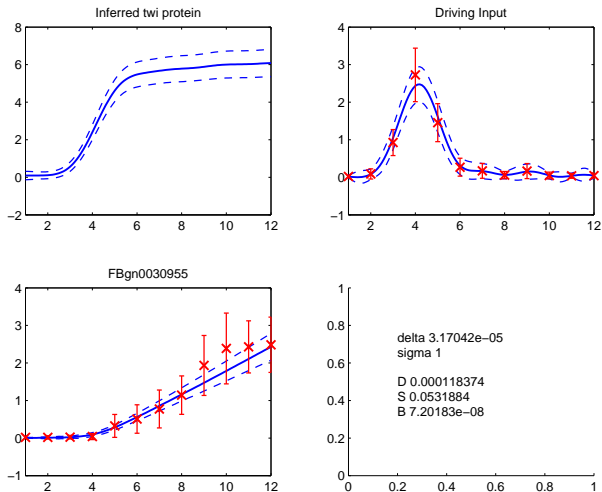


Figure : Model for flybase gene identity FBgn00309055.

# Results for Twi using the Cascade model

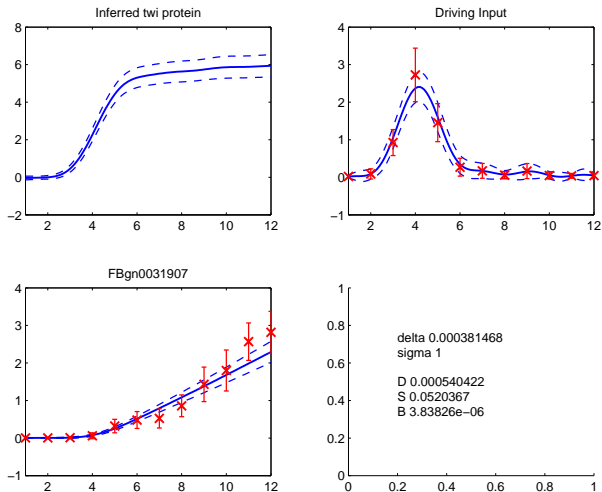


Figure : Model for flybase gene identity FBgn0031907.

# Results for Twi using the Cascade model

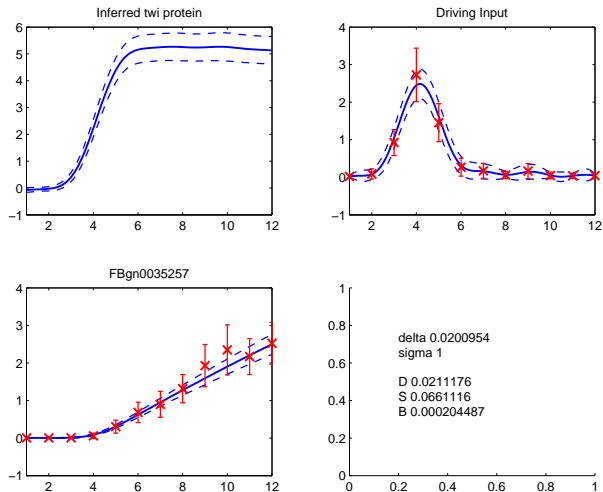


Figure : Model for flybase gene identity FBgn0035257.

# Results for Twi using the Cascade model

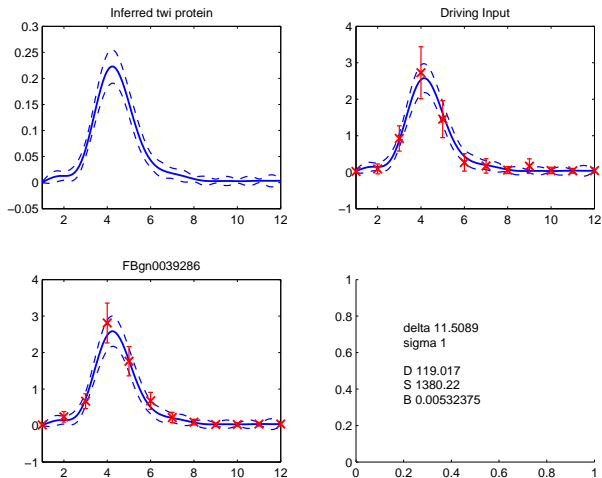


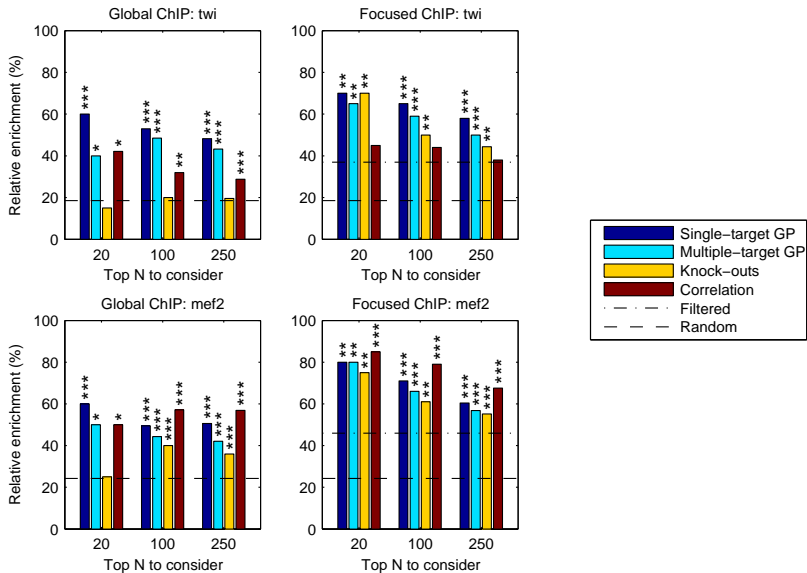
Figure : Model for flybase gene identity FBgn0039286.



# Evaluation methods

- ▶ Evaluate the ranking methods by taking a number of top-ranked targets and record the number of “positives” (Zinzen et al., 2009):
  - ▶ targets with ChIP-chip binding sites within 2 kb of gene
  - ▶ (targets differentially expressed in TF knock-outs)
- ▶ Compare against
  - ▶ Ranking by correlation of expression profiles
  - ▶ Ranking by  $q$ -value of differential expression in knock-outs
- ▶ Optionally focus on genes with annotated expression in tissues of interest

# Results



\*\*\*\*:  $p < 0.001$ , \*\*\*:  $p < 0.01$ , \*\*:  $p < 0.05$

# Summary

- ▶ Cascade models allow genomewide analysis of potential targets given only expression data.
- ▶ Once a set of potential candidate targets have been identified, they can be modelled in a more complex manner.
- ▶ We don't have ground truth, but evidence indicates that the approach *can* perform as well as knockouts.

# Outline

Multivariate Gaussian Properties

Cascade Differential Equations

Multiple Transcription Factors

Conclusions

This Provisional PDF corresponds to the article as it appeared upon acceptance. Fully formatted PDF and full text (HTML) versions will be made available soon.

## **Identifying targets of multiple co-regulating transcription factors from expression time-series by Bayesian model comparison**

*BMC Systems Biology* 2012, **6**:53 doi:10.1186/1752-0509-6-53

Michalis K Titsias (mtitsias@well.ox.ac.uk)

Antti Honkela (antti.honkela@hiit.fi)

Neil D Lawrence (n.lawrence@sheffield.ac.uk)

Magnus Rattray (m.rattray@sheffield.ac.uk)

# A “middle-out” approach for inferring regulatory networks

Task: find targets of a small number of co-regulating transcription factors (TFs) from time-series expression data:

- ▶ Stage 1: Sub-network training (~100 targets):
  - ▶ Fit regulation model on sub-network of known structure
  - ▶ Infer TF protein concentration functions
- ▶ Stage 2: Genome-wide scanning:
  - ▶ Fit alternative regulation models to all potential targets
  - ▶ Score models and identify well supported TF-target links
- ▶ Challenges:
  - ▶ Fitting and scoring >10000 models
  - ▶ Not all regulation is modelled: an open system

# A “middle-out” approach for inferring regulatory networks

Task: find targets of a small number of co-regulating transcription factors (TFs) from time-series expression data:

- ▶ Stage 1: Sub-network training (~100 targets):
  - ▶ Fit regulation model on sub-network of known structure
  - ▶ Infer TF protein concentration functions
- ▶ Stage 2: Genome-wide scanning:
  - ▶ Fit alternative regulation models to all potential targets
  - ▶ Score models and identify well supported TF-target links
- ▶ Challenges:
  - ▶ Fitting and scoring >10000 models
  - ▶ Not all regulation is modelled: an open system

# A “middle-out” approach for inferring regulatory networks

Task: find targets of a small number of co-regulating transcription factors (TFs) from time-series expression data:

- ▶ Stage 1: Sub-network training (~100 targets):
  - ▶ Fit regulation model on sub-network of known structure
  - ▶ Infer TF protein concentration functions
- ▶ Stage 2: Genome-wide scanning:
  - ▶ Fit alternative regulation models to all potential targets
  - ▶ Score models and identify well supported TF-target links
- ▶ Challenges:
  - ▶ Fitting and scoring >10000 models
  - ▶ Not all regulation is modelled: an open system



# A “middle-out” approach for inferring regulatory networks

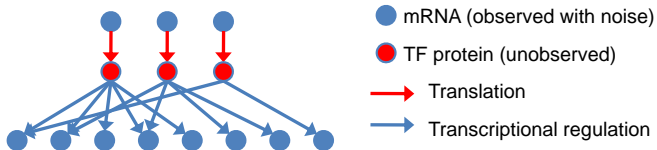
Task: find targets of a small number of co-regulating transcription factors (TFs) from time-series expression data:

- ▶ Stage 1: Sub-network training (~100 targets):
  - ▶ Fit regulation model on sub-network of known structure
  - ▶ Infer TF protein concentration functions
- ▶ Stage 2: Genome-wide scanning:
  - ▶ Fit alternative regulation models to all potential targets
  - ▶ Score models and identify well supported TF-target links
- ▶ Challenges:
  - ▶ Fitting and scoring >10000 models
  - ▶ Not all regulation is modelled: an open system

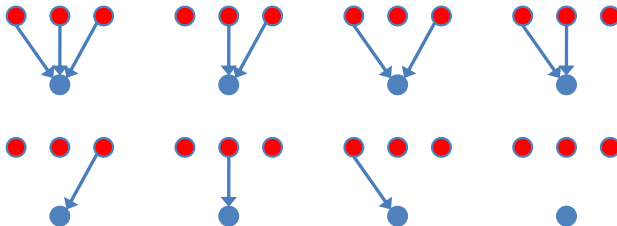
# A “middle-out” approach for inferring regulatory networks

- Training stage: Parameter estimation on known network

(a): Training phase



(b): Prediction phase

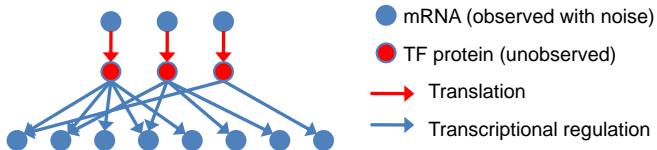


- Scanning stage: Bayesian evidence model scoring for

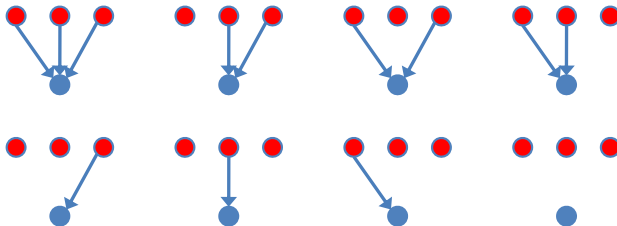
# A “middle-out” approach for inferring regulatory networks

- ▶ Training stage: Parameter estimation on known network

(a): Training phase



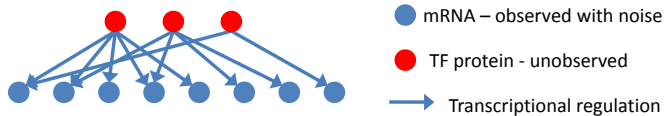
(b): Prediction phase



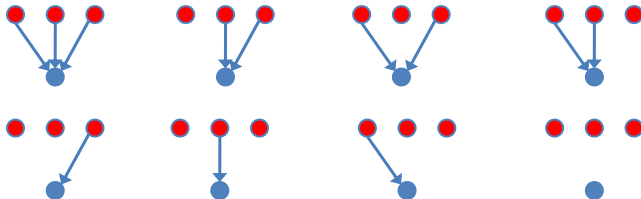
- ▶ Scanning stage: Bayesian evidence model scoring for

# A “middle-out” approach for inferring regulatory networks

- ▶ Training stage with post-translational modification



- ▶ Scanning stage: Bayesian evidence model scoring for target inference



# Model of transcriptional regulation

- Transcription

$$\frac{dm_j(t)}{dt} = F(p_1(t), \dots, p_K(t); \theta_j) - d_j m_j(t)$$

$m_j(t)$  – target gene  $j$  mRNA concentration function

$p_i(t)$  – transcription factor  $i$  protein concentration function

$F(\mathbf{p}; \theta_j)$  – regulation model,  $d_j$  – mRNA decay rate

- Translation (optional)

$$\frac{dp_i(t)}{dt} = f_i(t) - \delta_i p_i(t)$$

$f_i(t)$  – transcription factor  $i$  mRNA concentration function

$\delta_i$  – protein decay rate

# Model of transcriptional regulation

- Transcription

$$\frac{dm_j(t)}{dt} = F(p_1(t), \dots, p_K(t); \theta_j) - d_j m_j(t)$$

$m_j(t)$  – target gene  $j$  mRNA concentration function

$p_i(t)$  – transcription factor  $i$  protein concentration function

$F(\mathbf{p}; \theta_j)$  – regulation model,  $d_j$  – mRNA decay rate

- Translation (optional)

$$\frac{dp_i(t)}{dt} = f_i(t) - \delta_i p_i(t)$$

$f_i(t)$  – transcription factor  $i$  mRNA concentration function

$\delta_i$  – protein decay rate

# Gaussian process inference over latent functions

- ▶ Transcription factors considered **inputs** to the system
- ▶ Modelled as samples from a Gaussian process prior distribution
- ▶ Equations linear in  $m(t)$  can be solved as a function of  $p(t)$  so no need for numerical ODE solver to compute likelihood
- ▶ Useful way to close an open system
- ▶ Can ignore TF mRNA data and treat  $p(t)$  as latent function
- ▶ Bayesian MCMC used to infer  $p(t)$  and all model parameters

Gao et al. (2008); Titsias et al. (2009); Honkela et al. (2010);  
Titsias et al. (2012)

# Gaussian process inference over latent functions

- ▶ Transcription factors considered **inputs** to the system
- ▶ Modelled as samples from a Gaussian process prior distribution
- ▶ Equations linear in  $m(t)$  can be solved as a function of  $p(t)$  so no need for numerical ODE solver to compute likelihood
- ▶ Useful way to close an open system
- ▶ Can ignore TF mRNA data and treat  $p(t)$  as latent function
- ▶ Bayesian MCMC used to infer  $p(t)$  and all model parameters

Gao et al. (2008); Titsias et al. (2009); Honkela et al. (2010);  
Titsias et al. (2012)



# Gaussian process inference over latent functions

- ▶ Transcription factors considered **inputs** to the system
- ▶ Modelled as samples from a Gaussian process prior distribution
- ▶ Equations linear in  $m(t)$  can be solved as a function of  $p(t)$  so no need for numerical ODE solver to compute likelihood
- ▶ Useful way to close an open system
- ▶ Can ignore TF mRNA data and treat  $p(t)$  as latent function
- ▶ Bayesian MCMC used to infer  $p(t)$  and all model parameters

Gao et al. (2008); Titsias et al. (2009); Honkela et al. (2010);  
Titsias et al. (2012)

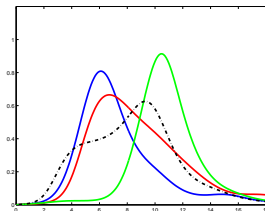
# Gaussian process inference over latent functions

- ▶ Transcription factors considered **inputs** to the system
- ▶ Modelled as samples from a Gaussian process prior distribution
- ▶ Equations linear in  $m(t)$  can be solved as a function of  $p(t)$  so no need for numerical ODE solver to compute likelihood
- ▶ Useful way to close an open system
- ▶ Can ignore TF mRNA data and treat  $p(t)$  as latent function
- ▶ Bayesian MCMC used to infer  $p(t)$  and all model parameters

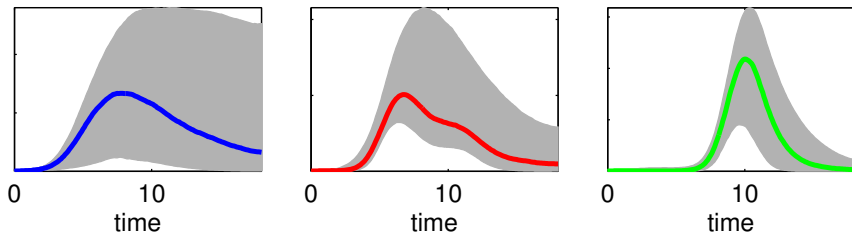
Gao et al. (2008); Titsias et al. (2009); Honkela et al. (2010);  
Titsias et al. (2012)

# Artificial data: one experimental condition

**Ground Truth TFs**

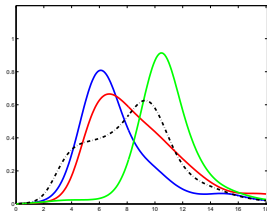


**Inferred TF concentrations after training stage**

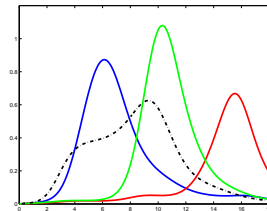


# Artificial data: two experimental conditions

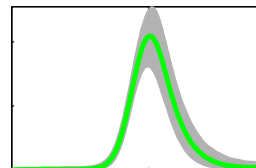
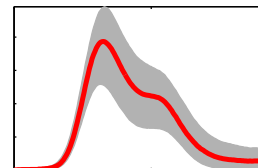
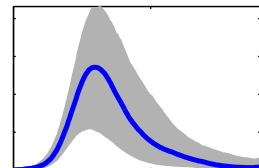
**True TFs condition 1**



**True TFs condition 2**



**Inferred TF concentrations for condition 1**



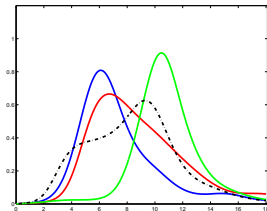
0 10  
time

0 10  
time

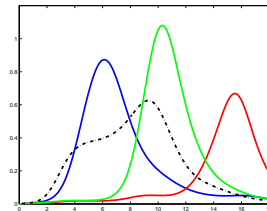
0 10  
time

# Artificial data: two experimental conditions

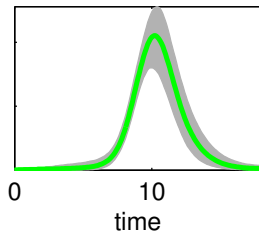
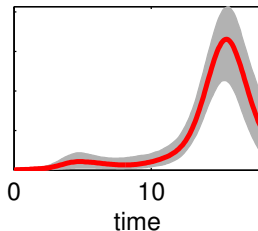
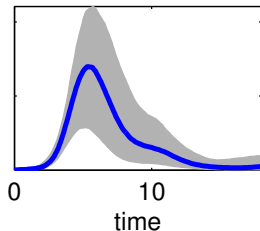
**True TFs condition 1**



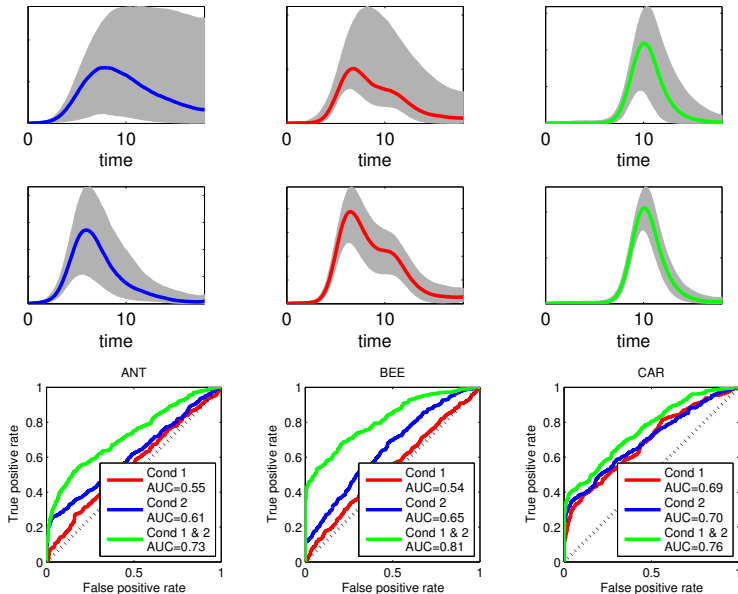
**True TFs condition 2**



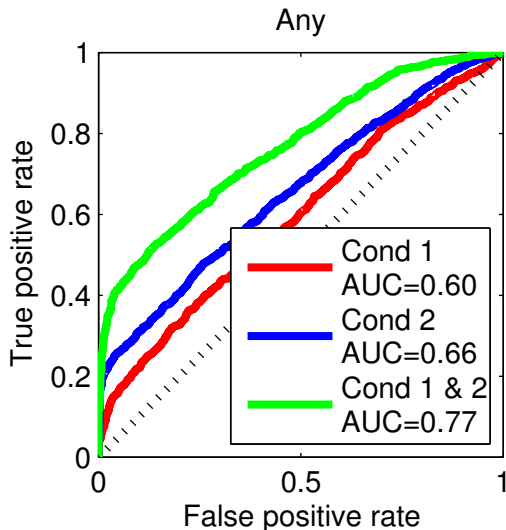
**Inferred TF concentrations for condition 2**



# Artificial data: scanning performance for each TF

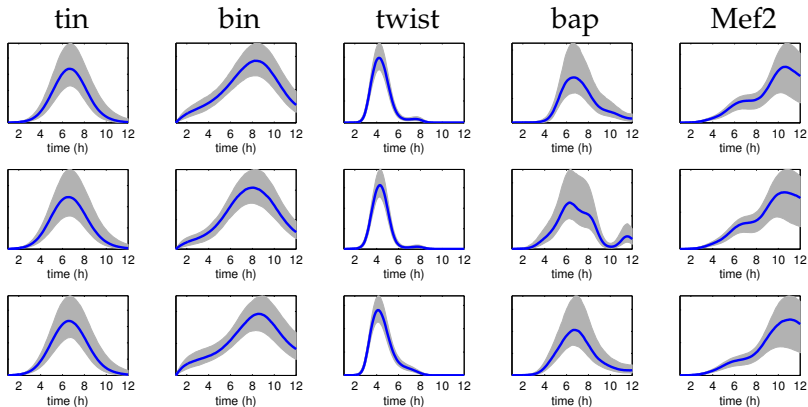


## Artificial data: scanning performance for all TFs



# Drosophila training

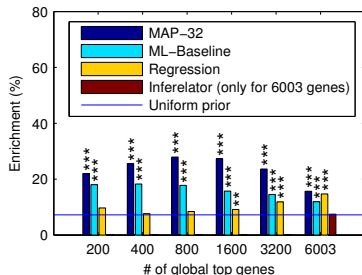
- ▶ Sub-network of 96 genes targeted by 5 TFs during *Drosophila* mesoderm development (Zinzen et al., 2009).
- ▶ Data: wild-type times series, 3 replicates (Tomancak et al., 2002).



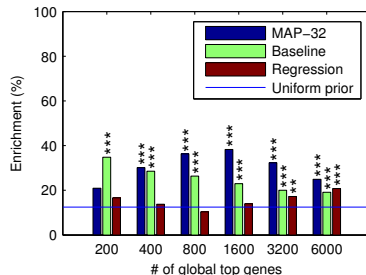


# Drosophila scanning: model ranking

- ▶ Rank target gene regulation models by their posterior probability across all  $2^5 = 32$  possible models
- ▶ Validate predicted links by enrichment for genes within 2kb of ChIP-chip TF binding predictions from Zinzen et al. (2009).

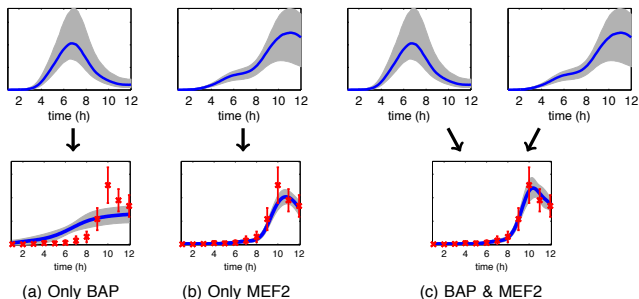


All “non-quiet” genes



All targets with in situ evidence

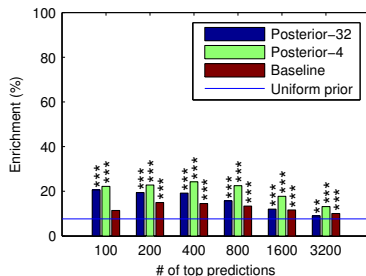
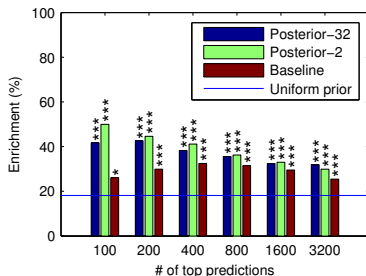
# Coregulated Target Example



A highly ranked putative joint target of BAP and MEF2. The candidate gene is confirmed as a joint target by independent ChIP-chip studies Zinzen et al. (2009).

# Drosophila scanning: link ranking

- ▶ TF-target link and link-pair ranking according to posterior probability of particular single TF or double TF regulations
- ▶ Validate predicted links by enrichment for genes within 2kb of ChIP-chip TF binding predictions from Zinzen et al. (2009).



# Summary and Conclusion

- ▶ Middle-out approach: sub-network training followed by genome-wide scanning
- ▶ Training: Bayesian inference of regulation model parameters and TF protein concentration functions
- ▶ Scanning: Bayesian model scoring for inferring TF-target link probabilities
- ▶ More informative conditions → better performance
- ▶ Robust to existence of some unknown regulating TFs
- ▶ Significant enrichment of inferred TF-target links for nearby ChIP-chip binding in drosophila development example

# Summary and Conclusion

- ▶ Middle-out approach: sub-network training followed by genome-wide scanning
- ▶ Training: Bayesian inference of regulation model parameters and TF protein concentration functions
- ▶ Scanning: Bayesian model scoring for inferring TF-target link probabilities
- ▶ More informative conditions → better performance
- ▶ Robust to existence of some unknown regulating TFs
- ▶ Significant enrichment of inferred TF-target links for nearby ChIP-chip binding in drosophila development example

# Summary and Conclusion

- ▶ Middle-out approach: sub-network training followed by genome-wide scanning
- ▶ Training: Bayesian inference of regulation model parameters and TF protein concentration functions
- ▶ Scanning: Bayesian model scoring for inferring TF-target link probabilities
- ▶ More informative conditions → better performance
- ▶ Robust to existence of some unknown regulating TFs
- ▶ Significant enrichment of inferred TF-target links for nearby ChIP-chip binding in drosophila development example

# Summary

- ▶ Flexible method for probability densities over functions.
- ▶ Covariance function is key: defines how different data interrelate.
- ▶ Problems occur if there are discontinuities in the function.
- ▶ Applications in Transcriptional Regulation Provide Examples

# 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.
- G. Della Gatta, M. Bansal, A. Ambesi-Impiombato, D. Antonini, C. Missero, and D. di Bernardo. Direct targets of the trp63 transcription factor revealed by a combination of gene expression profiling and reverse engineering. *Genome Research*, 18(6):939–948, Jun 2008. [[URL](#)]. [[DOI](#)].
- P. Gao, A. Honkela, M. Rattray, and N. D. Lawrence. Gaussian process modelling of latent chemical species: Applications to inferring transcription factor activities. *Bioinformatics*, 24:i70–i75, 2008. [[PDF](#)]. [[DOI](#)].
- D. S. Goodsell. The molecular perspective: p53 tumor suppressor. *The Oncologist*, Vol. 4, No. 2, 138–139, April 1999, 4 (2):138–139, 1999.
- A. Honkela, C. Girardot, E. H. Gustafson, Y.-H. Liu, E. E. M. Furlong, N. D. Lawrence, and M. Rattray. Model-based method for transcription factor target identification with limited data. *Proc. Natl. Acad. Sci. USA*, 107(17):7793–7798, Apr 2010. [[DOI](#)].
- A. A. Kalaitzis and N. D. Lawrence. A simple approach to ranking differentially expressed gene expression time courses through Gaussian process regression. *BMC Bioinformatics*, 12(180), 2011. [[DOI](#)].
- Y. Lazebnik. Can a biologist fix a radio? or, what I learned while studying apoptosis. *Cancer Cell*, 2:179–182, 2002.
- J. Oakley and A. O’Hagan. Bayesian inference for the uncertainty distribution of computer model outputs. *Biometrika*, 89(4):769–784, 2002.
- C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006. [[Google Books](#)].
- M. K. Titsias, A. Honkela, N. D. Lawrence, and M. Rattray. Identifying targets of multiple co-regulated transcription factors from expression time-series by Bayesian model comparison. *BMC Systems Biology*, 6(53), 2012. [[DOI](#)].
- M. K. Titsias, N. D. Lawrence, and M. Rattray. Efficient sampling for Gaussian process inference using control variables. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 21, pages 1681–1688, Cambridge, MA, 2009. MIT Press. [[PDF](#)].
- P. Tomancak, A. Beaton, R. Weiszmman, E. Kwan, S. Shu, S. E. Lewis, S. Richards, M. Ashburner, V. Hartenstein, S. E. Celniker, and G. M. Rubin. Systematic determination of patterns of gene expression during *Drosophila* embryogenesis. *Genome Biology*, 3(12):RESEARCH0088, 2002.
- R. P. Zinzen, C. Girardot, J. Gagneur, M. Braun, and E. E. M. Furlong. Combinatorial binding predicts spatio-temporal cis-regulatory activity. *Nature*, 462(7269):65–70, Nov 2009. [[URL](#)]. [[DOI](#)].