PRIB Tutorial: Gaussian Processes and Gene Regulation

Neil D. Lawrence
present work with Magnus Rattray (co-PI), Pei Gao, Antti Honkela, Guido Sanguinetti, Jennifer Withers

Departments of Neuro- and Computer Science, University of Sheffield, U.K.
Tutorial at PRIB 2010, Nijmegen, Netherlands

22nd September 2010

Outline

- Motivation
- Probabilistic Model for $f(t)$
- Cascade Differential Equations
- Discussion

Outline

Motivation

Probabilistic Model for $f(t)$

Cascade Differential Equations

Discussion

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.

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.

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.

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.

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.

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.

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.

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).

General Approach

Broadly Speaking: Two approaches to modeling

data modeling

mechanistic modeling

General Approach

Broadly Speaking: Two approaches to modeling

data modeling

let the data “speak”

mechanistic modeling

General Approach

Broadly Speaking: Two approaches to modeling

data modeling	*mechanistic modeling*
let the data “speak”	impose physical laws

General Approach

Broadly Speaking: Two approaches to modeling

data modeling

let the data “speak”

computational models

mechanistic modeling

impose physical laws

General Approach

Broadly Speaking: Two approaches to modeling

data modeling	*mechanistic modeling*
let the data "speak"	impose physical laws
computational models	systems models

General Approach

Broadly Speaking: Two approaches to modeling

data modeling

let the data “speak”

computational models

adaptive models

mechanistic modeling

impose physical laws

systems models

General Approach

Broadly Speaking: Two approaches to modeling

data modeling	*mechanistic modeling*
let the data "speak"	impose physical laws
computational models	systems models
adaptive models	differential equations

General Approach

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

General Approach

Broadly Speaking: Two approaches to modeling

data modeling	*mechanistic modeling*
let the data “speak”	impose physical laws
computational models	systems models
adaptive models	differential equations
PCA, clustering	SDE, ODE models

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 damages 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 meitosis, chromosomes have divided.
    - S: Cell is undergoing meitosis (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 feeback loop with NF-κB.

p53 DNA Damage Repair

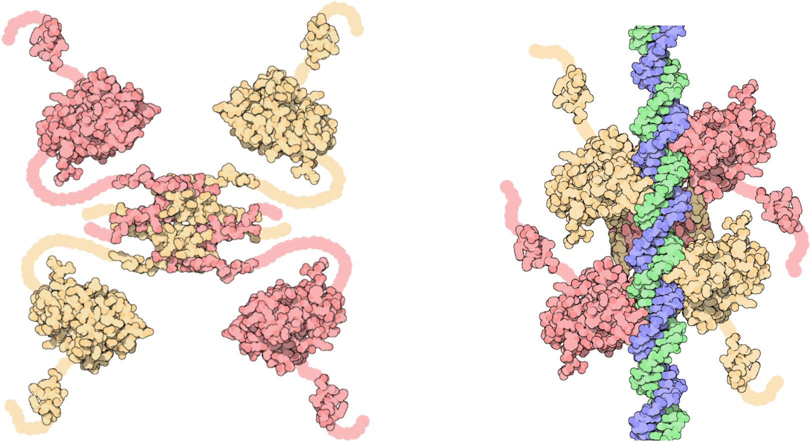


Figure: p53. *Left* unbound, *Right* bound to DNA. Images by David S. Goodsell from `http://www.rcsb.org/` (see the“Molecule of the Month” feature).

p53

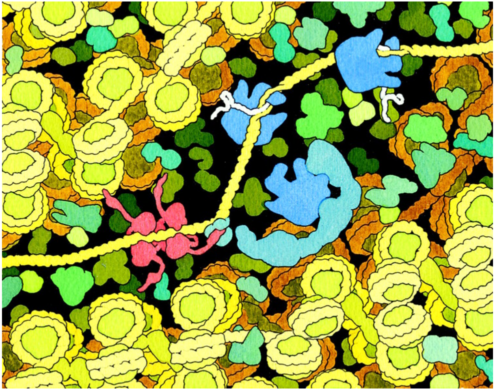


Figure: Repair of DNA damage by p53. Image from Goodsell (1999).

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 goverened 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).

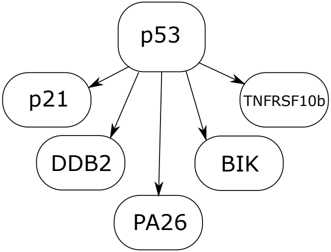


Figure: p53 SIM network motif as modelled by Barenco et al. 2006.

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.

Mathematical Model

- Differential equation model of system.

$$\frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = b_j + s_j f(t) - d_j x_j(t)$$

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

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

Mathematical Model

- Differential equation model of system.

$$\frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = b_j + s_j f(t) - d_j x_j(t)$$

$$d_j x_j(t) + \frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = b_j + s_j f(t)$$

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

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

Mathematical Model

- Differential equation model of system.

$$\frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = b_j + s_j f(t) - d_j x_j(t)$$

$$d_j x_j(t) + \frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = b_j + s_j f(t)$$

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

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

Mathematical Model

- Differential equation model of system.

$$\frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = b_j + s_j f(t) - d_j x_j(t)$$

$$d_j x_j(t) + \frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = b_j + s_j f(t)$$

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

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

Mathematical Model

- Differential equation model of system.

$$\frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = b_j + s_j f(t) - d_j x_j(t)$$

$$d_j x_j(t) + \frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = b_j + s_j f(t)$$

  rate of mRNA transcription, baseline transcription rate, transcription factor activity, mRNA decay
- We have observations of $x_j(t)$ from gene expression. .
- Reorder differential equation.
- An estimate of $\frac{\mathrm{d}x_j(t)}{\mathrm{d}t}$ is obtained through fitting polynomials.
- Jointly estimate $f(t)$ 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.

$$\frac{\mathrm{d}x_j\left(t\right)}{\mathrm{d}t} = b_j + s_j f\left(t\right) - d_j x_j\left(t\right)$$

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

- We have observations of $x_j\left(t\right)$ from gene expression.

Mathematical Model

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

$$\frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = b_j + s_j f(t) - d_j x_j(t)$$

$$d_j x_j(t) \approx b_j + s_j f(t)$$

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

- We have observations of $x_j(t)$ 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.

$$\frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = b_j + s_j f(t) - d_j x_j(t)$$

$$d_j x_j(t) \approx b_j + s_j f(t)$$

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

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

Mathematical Model

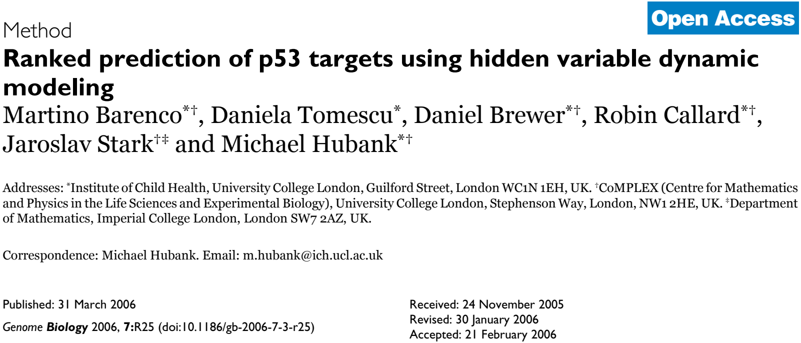
- Clustering model is equivalent to assuming d_j, b_j, and s_j are v. large.

$$\frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = b_j + s_j f(t) - d_j x_j(t)$$
$$d_j x_j(t) \approx b_j + s_j f(t)$$

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

- We have observations of $x_j(t)$ 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.

Mathematical Model



Method

Ranked prediction of p53 targets using hidden variable dynamic modeling

Martino Barenco[*†], Daniela Tomescu[*], Daniel Brewer[*†], Robin Callard[*†], Jaroslav Stark[†‡] and Michael Hubank[*†]

Addresses: [*]Institute of Child Health, University College London, Guilford Street, London WC1N 1EH, UK. [†]CoMPLEX (Centre for Mathematics and Physics in the Life Sciences and Experimental Biology), University College London, Stephenson Way, London, NW1 2HE, UK. [‡]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

Response of p53

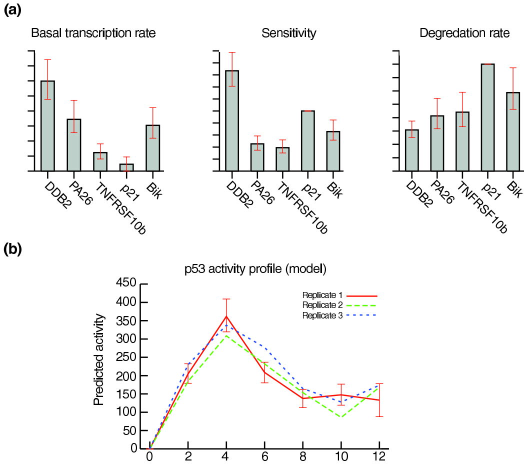


Figure: Results from Barenco et al. (2006). Top is parameter estimates. Bottom is inferred profile.

Respose to p53 ...

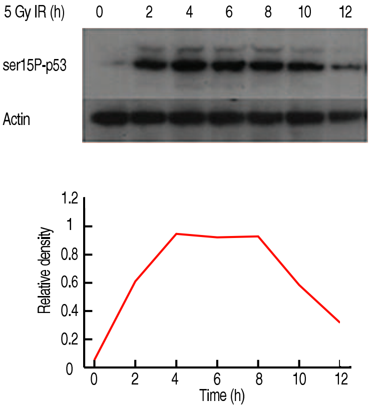


Figure: Results from Barenco et al. (2006). Activity profile of p53 was measured by Western blot to determine the levels of ser-15 phosphorylated p53 (ser15P-p53).

Outline

Motivation

Probabilistic Model for $f(t)$

Cascade Differential Equations

Discussion

Gaussian Distribution

Zero mean Gaussian distribution

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

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

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

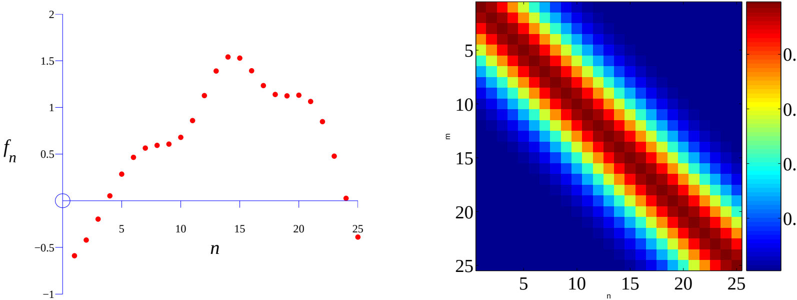
$$\mathcal{N}\left(\mathbf{f}|\mathbf{0}, \mathbf{K}\right) = \frac{1}{(2\pi)^{\frac{n}{2}} \left|\mathbf{K}\right|^{\frac{1}{2}}} \exp\left(-\frac{\mathbf{f}^{\top}\mathbf{K}^{-1}\mathbf{f}}{2}\right).$$

Sampling a Function

Multi-variate Gaussians

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

Gaussian Distribution Sample



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

(b) colormap showing correlations between dimensions

Figure: A sample from a 25 dimensional Gaussian distribution.

Covariance Function

The covariance matrix

- Covariance matrix shows correlation between points f_i and f_j if i is near to j.
- Less correlation if i is distant from j.
- Our ordering of points means that the *function appears smooth*.
- Let's focus on the joint distribution of two points from the 25.

Covariance Function

The covariance matrix

- Covariance matrix shows correlation between points f_i and f_j if i is near to j.
- Less correlation if i is distant from j.
- Our ordering of points means that the *function appears smooth*.
- Let's focus on the joint distribution of two points from the 25.

Covariance Function

The covariance matrix

- Covariance matrix shows correlation between points f_i and f_j if i is near to j.
- Less correlation if i is distant from j.
- Our ordering of points means that the *function appears smooth*.
- Let's focus on the joint distribution of two points from the 25.

Covariance Function

The covariance matrix

- Covariance matrix shows correlation between points f_i and f_j if i is near to j.
- Less correlation if i is distant from j.
- Our ordering of points means that the *function appears smooth*.
- Let's focus on the joint distribution of two points from the 25.

Prediction of f_2 from f_1

```
demGpCov2D([1 2])
```

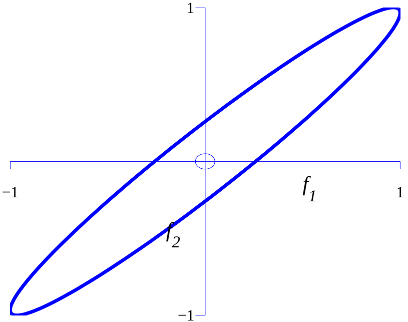


Figure: Covariance for $\begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$ is $\mathbf{K}_{12} = \begin{bmatrix} 1 & 0.966 \\ 0.966 & 1 \end{bmatrix}$.

Prediction of f_2 from f_1

`demGpCov2D([1 2])`

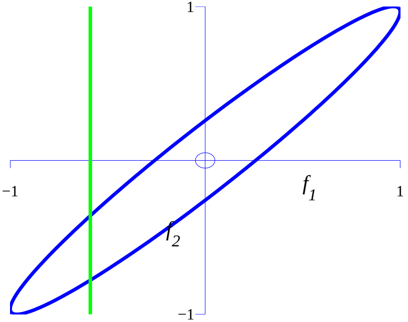


Figure: Covariance for $\begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$ is $\mathbf{K}_{12} = \begin{bmatrix} 1 & 0.966 \\ 0.966 & 1 \end{bmatrix}$.

Prediction of f_2 from f_1

`demGpCov2D([1 2])`

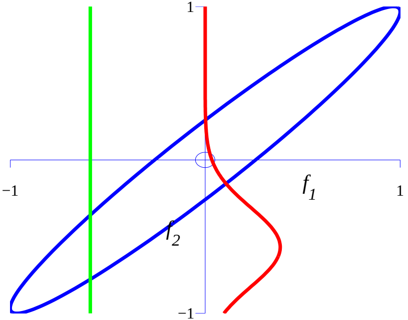


Figure: Covariance for $\begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$ is $\mathbf{K}_{12} = \begin{bmatrix} 1 & 0.966 \\ 0.966 & 1 \end{bmatrix}$.

Prediction of f_5 from f_1

```
demGpCov2D([1 5])
```

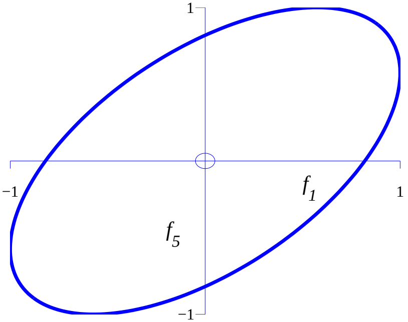


Figure: Covariance for $\begin{bmatrix} f_1 \\ f_5 \end{bmatrix}$ is $\mathbf{K}_{15} = \begin{bmatrix} 1 & 0.574 \\ 0.574 & 1 \end{bmatrix}$.

Prediction of f_5 from f_1

`demGpCov2D([1 5])`

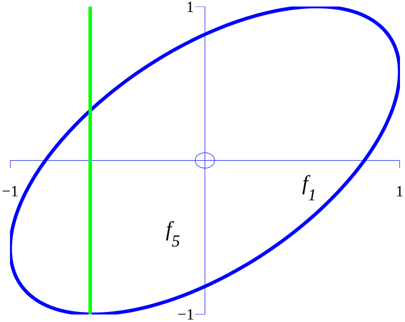


Figure: Covariance for $\begin{bmatrix} f_1 \\ f_5 \end{bmatrix}$ is $\mathbf{K}_{15} = \begin{bmatrix} 1 & 0.574 \\ 0.574 & 1 \end{bmatrix}$.

Prediction of f_5 from f_1

`demGpCov2D([1 5])`

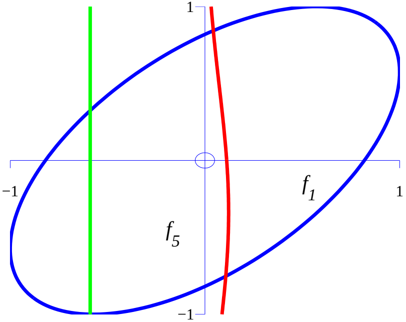


Figure: Covariance for $\begin{bmatrix} f_1 \\ f_5 \end{bmatrix}$ is $\mathbf{K}_{15} = \begin{bmatrix} 1 & 0.574 \\ 0.574 & 1 \end{bmatrix}$.

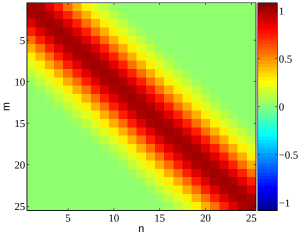
Covariance Functions

Where did this covariance matrix come from?

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

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

- 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 know as a kernel.



Covariance Samples

`demCovFuncSample`

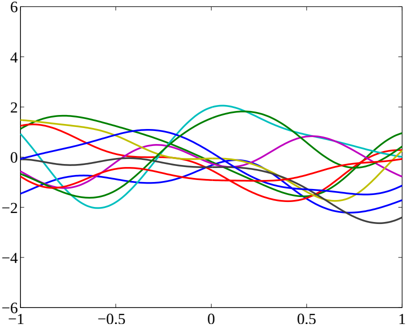


Figure: Exponentiated quadratic kernel with $\ell = 0.3$, $\alpha = 1$

Covariance Samples

`demCovFuncSample`

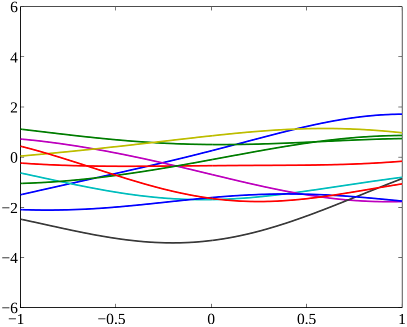


Figure: Exponentiated quadratic kernel with $\ell = 1$, $\alpha = 1$

Covariance Samples

`demCovFuncSample`

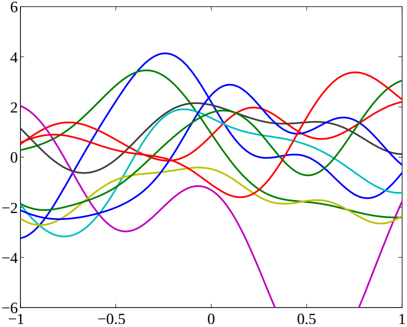


Figure: Exponentiated quadratic kernel with $\ell = 0.3$, $\alpha = 4$

Covariance Samples

`demCovFuncSample`

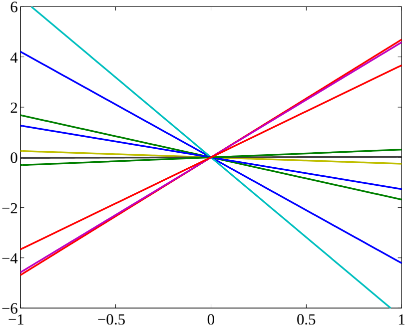


Figure: Linear covariance function, $\alpha = 16$.

Covariance Samples

`demCovFuncSample`

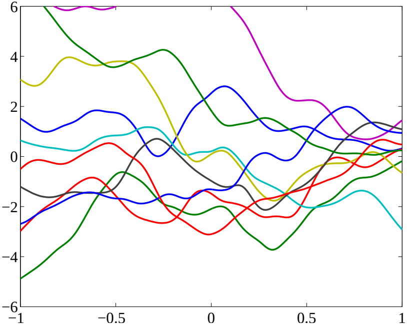


Figure: MLP covariance function, $\sigma_w^2 = 100$, $\sigma_b^2 = 100$, $\alpha = 8$.

Covariance Samples

`demCovFuncSample`

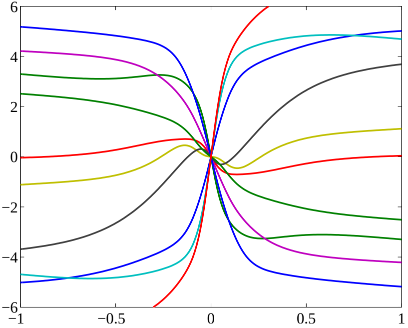


Figure: MLP covariance function, $\sigma_w^2 = 100$, $\sigma_b^2 = 0$, $\alpha = 8$.

Covariance Samples

`demCovFuncSample`

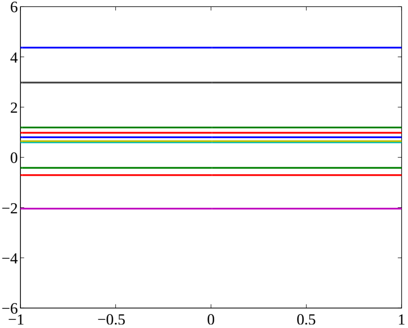


Figure: Bias term, $\alpha = 4$

Covariance Samples

`demCovFuncSample`

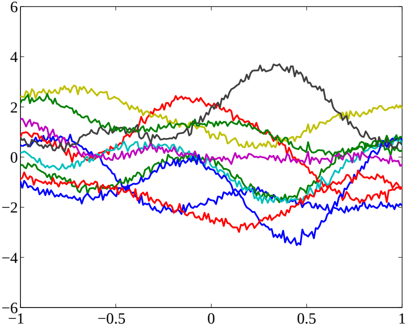


Figure: Exponentiated quadratic $\ell = 0.3$, $\alpha = 1$ plus bias term with $\alpha = 1$ plus white noise with $\alpha = 0.01$.

Covariance Samples

`demCovFuncSample`

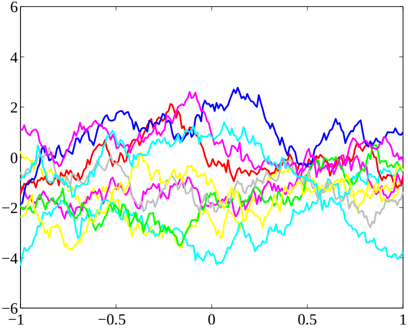


Figure: Ornstein-Uhlenbeck (stationary Gauss-Markov) covariance function $\ell = 1$, $\alpha = 4$.

Gaussian Process Interpolation

`demInterpolation`

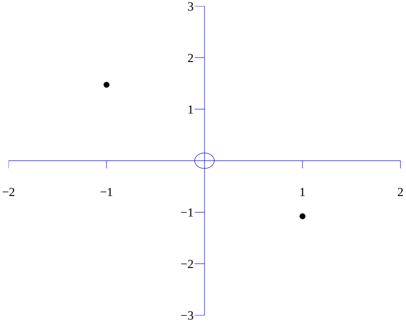


Figure: Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian Process Interpolation

`demInterpolation`

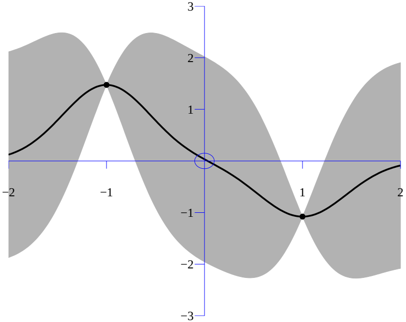


Figure: Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian Process Interpolation

`demInterpolation`

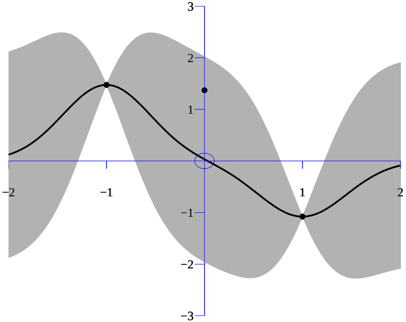


Figure: Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian Process Interpolation

`demInterpolation`

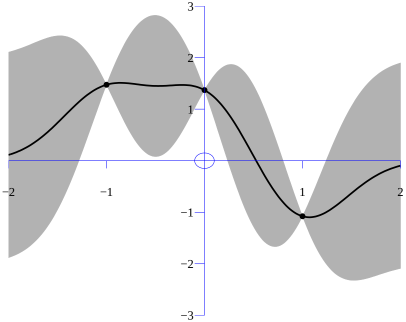


Figure: Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian Process Interpolation

`demInterpolation`

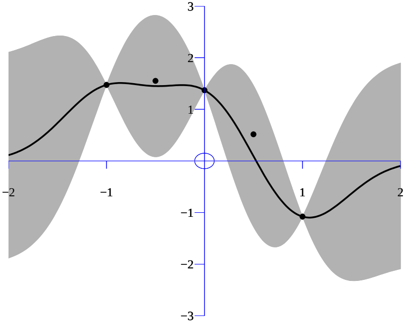


Figure: Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian Process Interpolation

`demInterpolation`

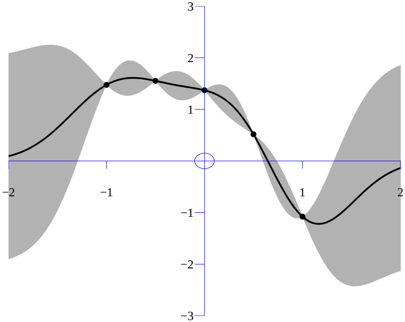


Figure: Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian Process Interpolation

`demInterpolation`

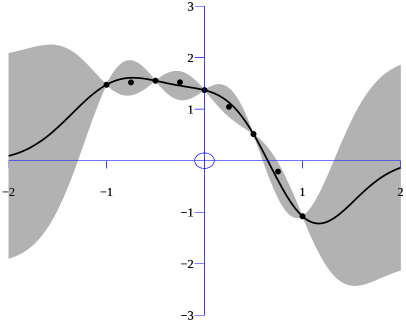


Figure: Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian Process Interpolation

`demInterpolation`

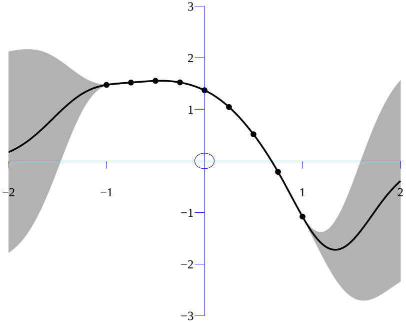


Figure: Real example: BACCO (see *e.g.* (Oakley and O'Hagan, 2002)). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Noise Models

Graph of a GP

- Relates input variables, **t**, to vector, **x**, through **f** given kernel parameters $\boldsymbol{\theta}$.
- Plate notation indicates independence of $x_i | f_i$.
- Noise model, $p(x_i | f_i)$ can take several forms.
- Simplest is Gaussian noise.

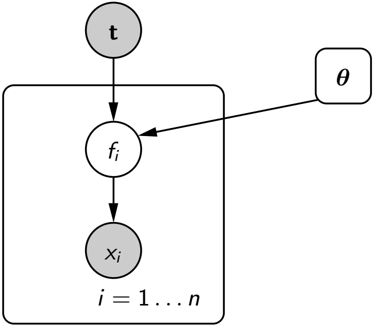


Figure: The Gaussian process depicted graphically.

Gaussian Noise

- Gaussian noise model,

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

  where σ^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

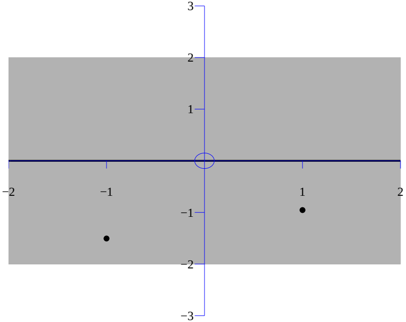


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

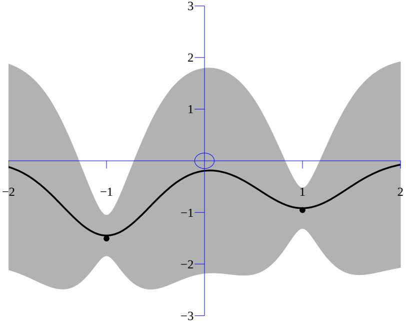


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

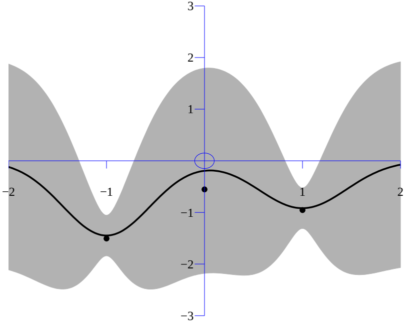


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

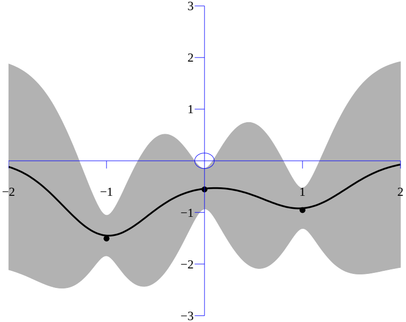


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

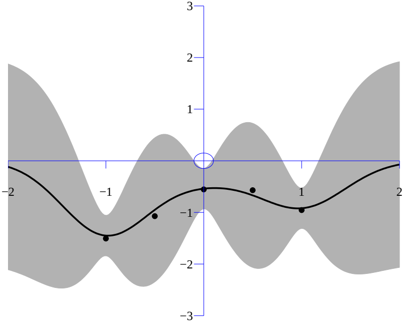


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

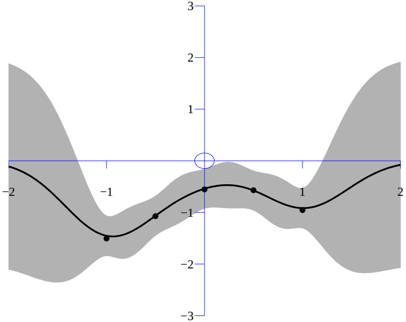


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

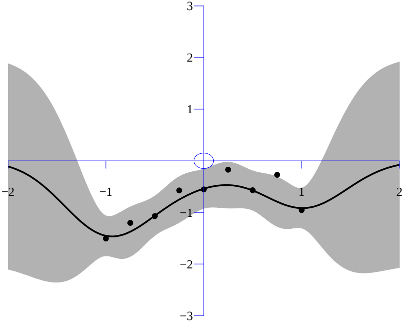


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

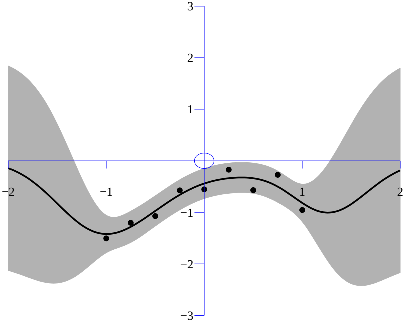
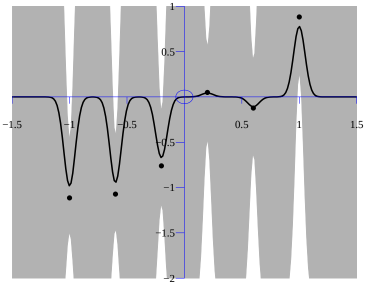
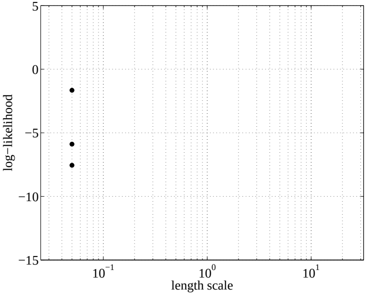


Figure: Examples include WiFi localization, C14 callibration curve.

Learning Kernel Parameters

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

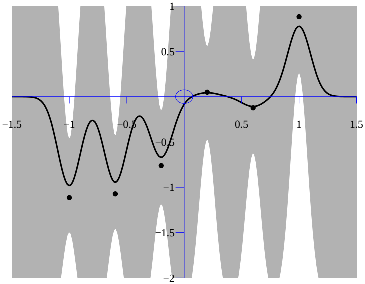


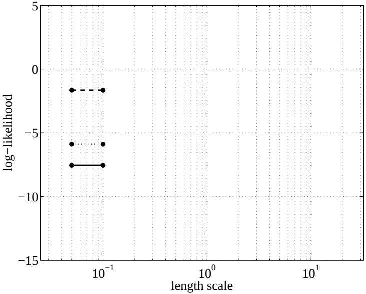


$$\log \mathcal{N}(\mathbf{x}|\mathbf{0}, \mathbf{K}) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}| - \frac{\mathbf{x}^\top \mathbf{K}^{-1}\mathbf{x}}{2}$$

Learning Kernel Parameters

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

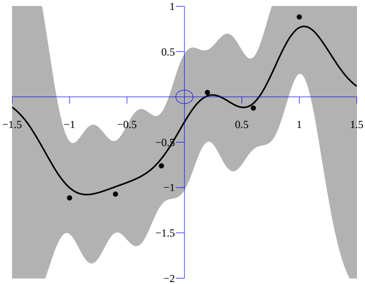


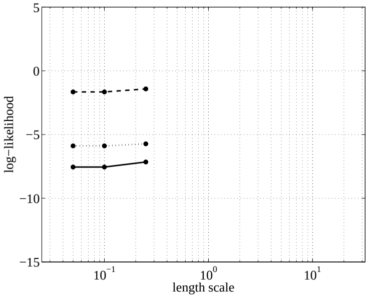


$$\log \mathcal{N}(\mathbf{x}|\mathbf{0},\mathbf{K}) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}| - \frac{\mathbf{x}^\top\mathbf{K}^{-1}\mathbf{x}}{2}$$

Learning Kernel Parameters

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

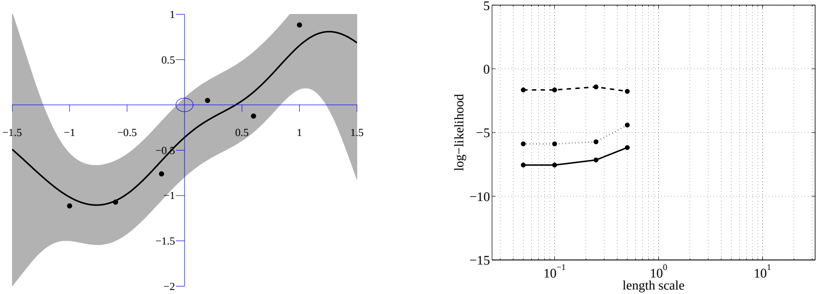




$$\log \mathcal{N}\left(\mathbf{x}|\mathbf{0},\mathbf{K}\right) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}| - \frac{\mathbf{x}^\top\mathbf{K}^{-1}\mathbf{x}}{2}$$

Learning Kernel Parameters

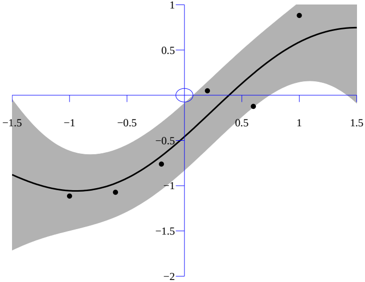
Can we determine length scales and noise levels from the data?

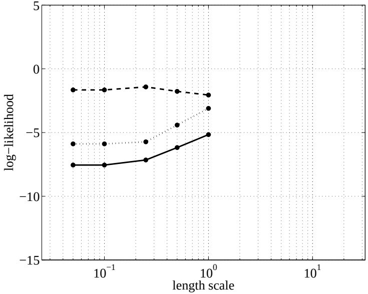


$$\log \mathcal{N}\left(\mathbf{x}|\mathbf{0},\mathbf{K}\right) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}| - \frac{\mathbf{x}^\top\mathbf{K}^{-1}\mathbf{x}}{2}$$

Learning Kernel Parameters

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

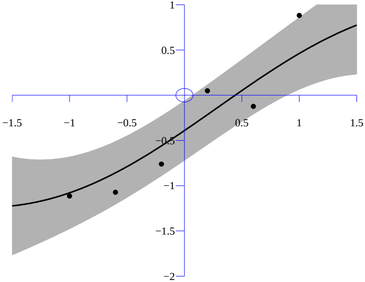


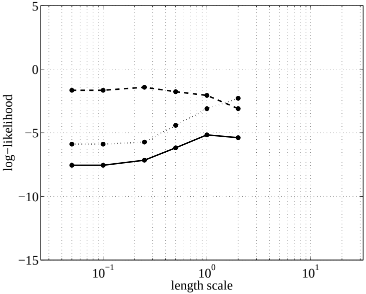


$$\log \mathcal{N}\left(\mathbf{x}|\mathbf{0},\mathbf{K}\right) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}| - \frac{\mathbf{x}^\top\mathbf{K}^{-1}\mathbf{x}}{2}$$

Learning Kernel Parameters

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

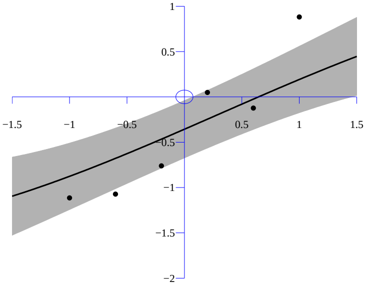


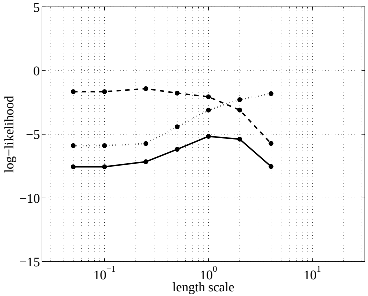


$$\log \mathcal{N}\left(\mathbf{x}|\mathbf{0},\mathbf{K}\right) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\log\left|\mathbf{K}\right| - \frac{\mathbf{x}^\top \mathbf{K}^{-1}\mathbf{x}}{2}$$

Learning Kernel Parameters

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

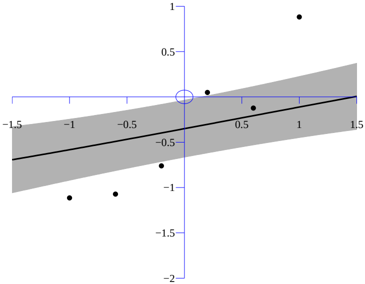


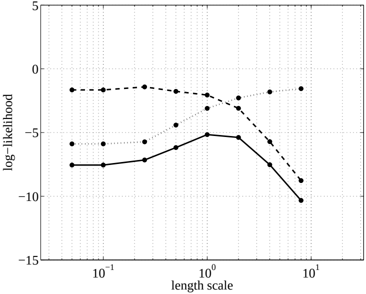


$$\log \mathcal{N}(\mathbf{x}|\mathbf{0},\mathbf{K}) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}| - \frac{\mathbf{x}^\top\mathbf{K}^{-1}\mathbf{x}}{2}$$

Learning Kernel Parameters

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

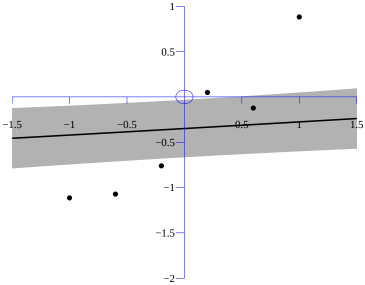


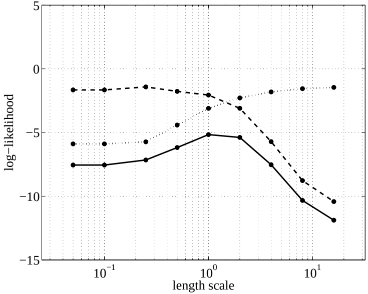


$$\log \mathcal{N}(\mathbf{x}|\mathbf{0},\mathbf{K}) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}| - \frac{\mathbf{x}^\top\mathbf{K}^{-1}\mathbf{x}}{2}$$

Learning Kernel Parameters

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





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

Example: Transcriptional Regulation

- First Order Differential Equation

$$\frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = b_j + s_j f(t) - d_j x_j(t)$$

- It turns out that our Gaussian process assumption for $f(t)$, implies $x(t)$ is also a Gaussian process.
- The new Gaussian process is over $f(t)$ and all its targets: $x_1(t), x_2(t), \ldots$ 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

$$\frac{\mathrm{d}x_j\left(t\right)}{\mathrm{d}t} = b_j + s_j f\left(t\right) - d_j x_j\left(t\right)$$

- It turns out that our Gaussian process assumption for $f(t)$, implies $x(t)$ is also a Gaussian process.
- The new Gaussian process is over $f(t)$ and all its targets: $x_1(t), x_2(t), \ldots$ 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

$$\frac{\mathrm{d}x_j\left(t\right)}{\mathrm{d}t} = b_j + s_j f\left(t\right) - d_j x_j\left(t\right)$$

- It turns out that our Gaussian process assumption for $f(t)$, implies $x(t)$ is also a Gaussian process.
- The new Gaussian process is over $f(t)$ and all its targets: $x_1(t), x_2(t), \ldots$ 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

$$\frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = b_j + s_j f(t) - d_j x_j(t)$$

- It turns out that our Gaussian process assumption for $f(t)$, implies $x(t)$ is also a Gaussian process.
- The new Gaussian process is over $f(t)$ and all its targets: $x_1(t), x_2(t), \dots$ 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

$$\frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = b_j + s_j f(t) - d_j x_j(t)$$

- It turns out that our Gaussian process assumption for $f(t)$, implies $x(t)$ is also a Gaussian process.
- The new Gaussian process is over $f(t)$ and all its targets: $x_1(t), x_2(t), \ldots$ etc.
- Our new covariance matrix gives correlations between all these functions.
- This gives us a *probabilistic* model for transcriptional regulation.

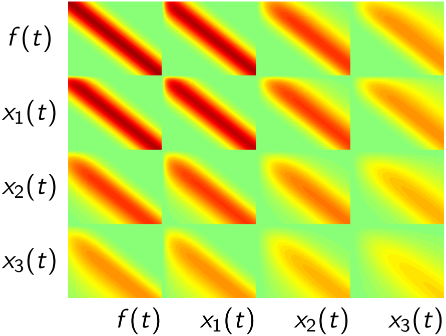
Covariance for Transcription Model

RBF covariance function for $f(t)$

$$x_i(t) = \frac{b_i}{d_i} + s_i \exp(-d_i t) \int_0^t f(u) \exp(d_i u) \, \mathrm{d}u.$$

- Joint distribution for $x_1(t)$, $x_2(t)$, $x_3(t)$, and $f(t)$.
- Here:

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



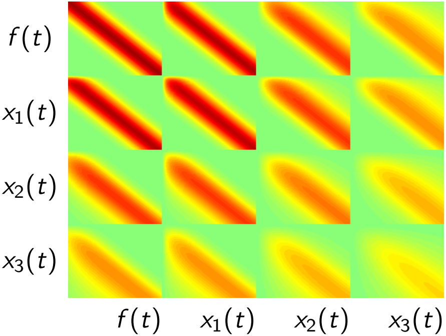
Covariance for Transcription Model

RBF covariance function for $f(t)$

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

- Joint distribution for $x_1(t)$, $x_2(t)$, $x_3(t)$, and $f(t)$.
- Here:

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



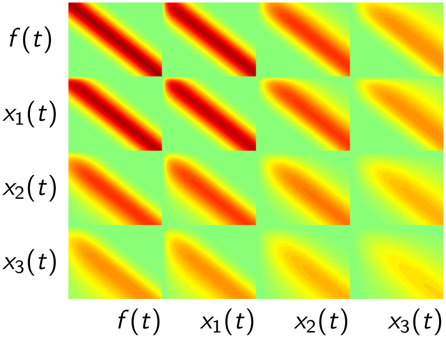
Covariance for Transcription Model

RBF covariance function for $f(t)$

$$x_i(t) = \frac{b_i}{d_i} + s_i \exp(-d_i t) \int_0^t f(u) \exp(d_i u) \, \mathrm{d}u.$$

- Joint distribution for $x_1(t)$, $x_2(t)$, $x_3(t)$, and $f(t)$.
- Here:

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



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

- `simSample`

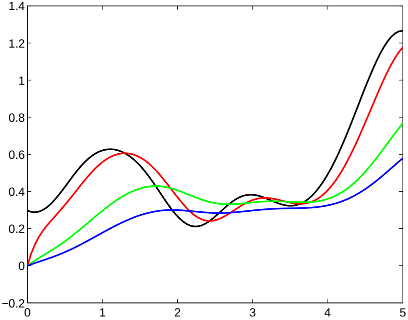


Figure: Joint samples from the ODE covariance, *black*: $f(t)$, *red*: $x_1(t)$ (high decay/sensitivity), *green*: $x_2(t)$ (medium decay/sensitivity) and *blue*: $x_3(t)$ (low decay/sensitivity).

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

- `simSample`

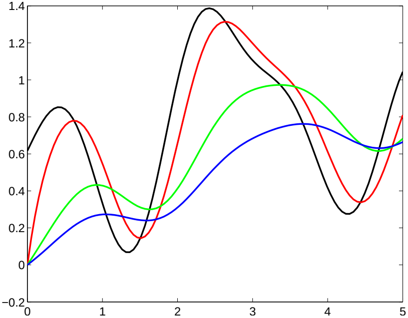


Figure: Joint samples from the ODE covariance, *black*: $f(t)$, *red*: $x_1(t)$ (high decay/sensitivity), *green*: $x_2(t)$ (medium decay/sensitivity) and *blue*: $x_3(t)$ (low decay/sensitivity).

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

- `simSample`

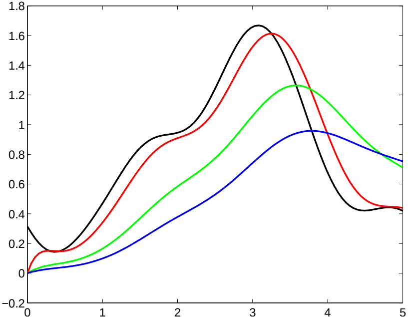


Figure: Joint samples from the ODE covariance, *black*: $f(t)$, *red*: $x_1(t)$ (high decay/sensitivity), *green*: $x_2(t)$ (medium decay/sensitivity) and *blue*: $x_3(t)$ (low decay/sensitivity).

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

- `simSample`

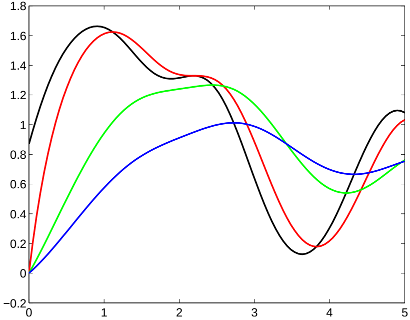
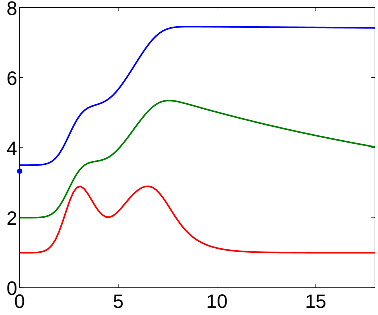


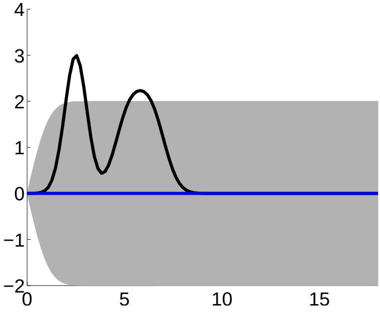
Figure: Joint samples from the ODE covariance, *black*: $f(t)$, *red*: $x_1(t)$ (high decay/sensitivity), *green*: $x_2(t)$ (medium decay/sensitivity) and *blue*: $x_3(t)$ (low decay/sensitivity).

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



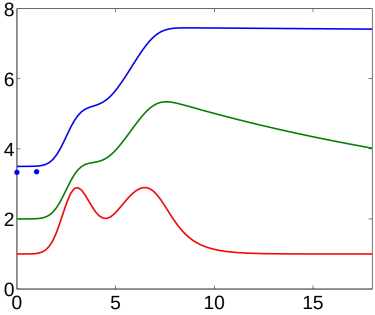
True “gene profiles” and noisy observations.



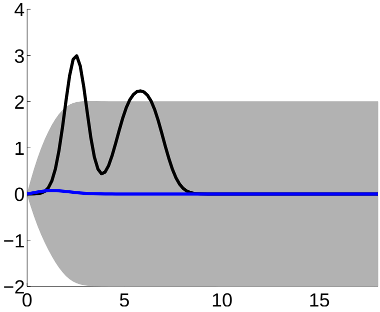
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



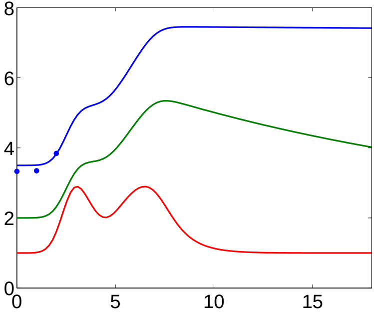
True "gene profiles" and noisy observations.



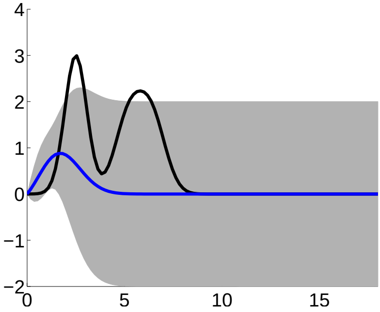
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



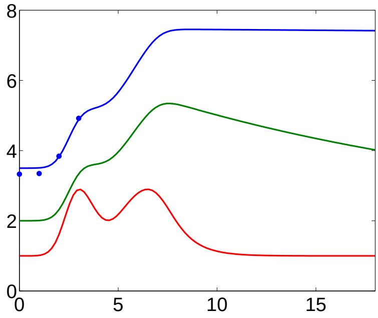
True “gene profiles” and noisy observations.



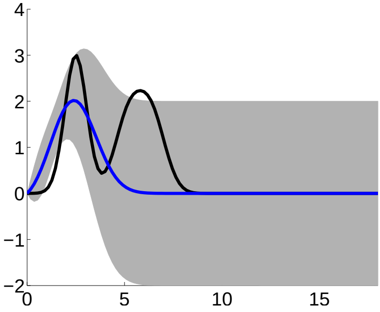
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



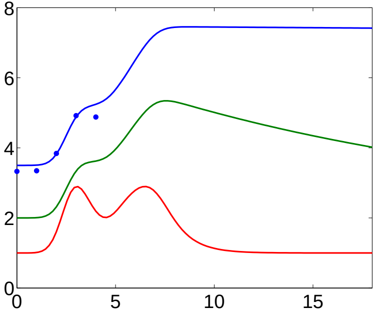
True "gene profiles" and noisy observations.



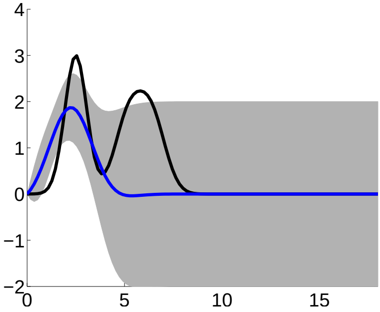
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



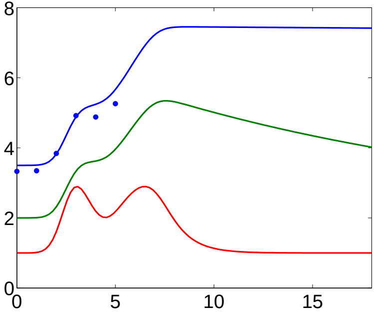
True "gene profiles" and noisy observations.



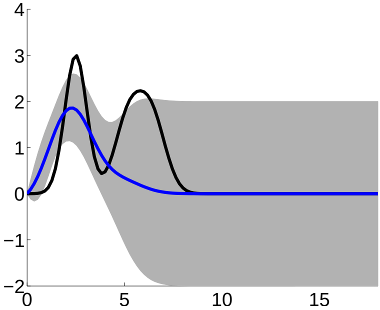
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



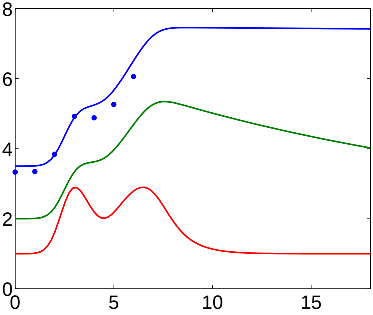
True "gene profiles" and noisy observations.



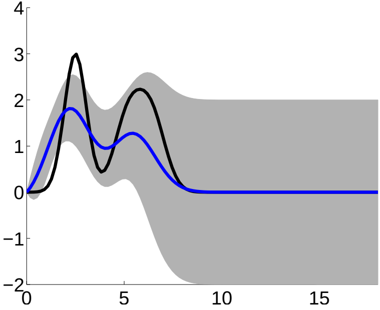
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



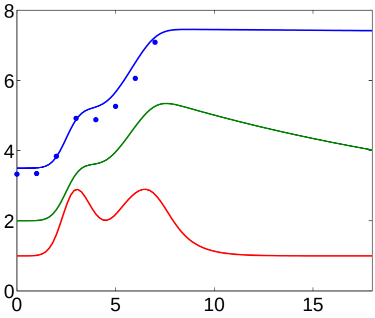
True “gene profiles” and noisy observations.



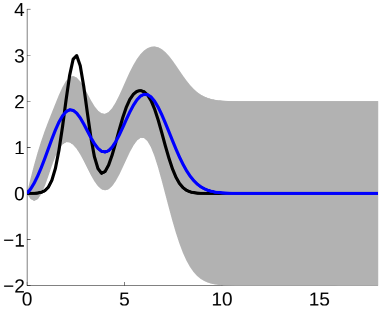
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



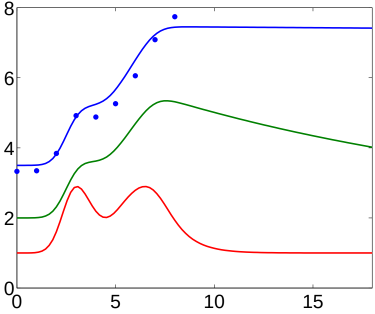
True "gene profiles" and noisy observations.



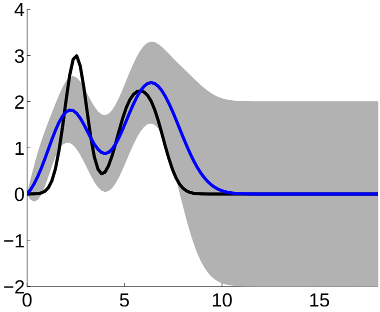
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



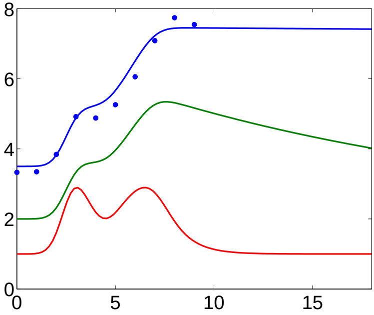
True “gene profiles” and noisy observations.



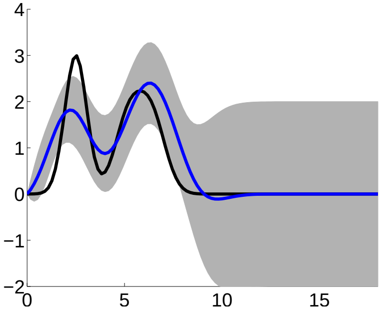
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



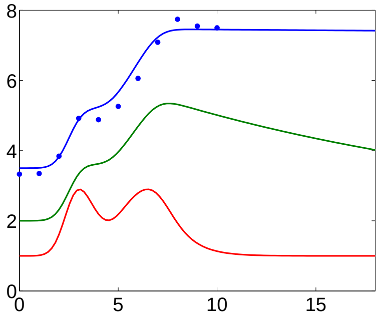
True "gene profiles" and noisy observations.



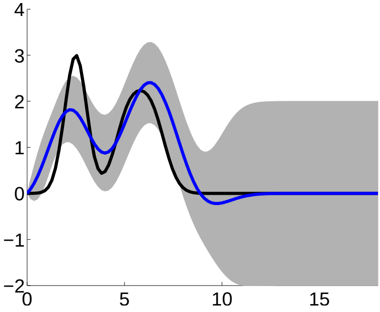
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



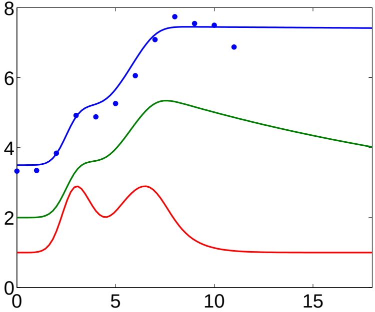
True "gene profiles" and noisy observations.



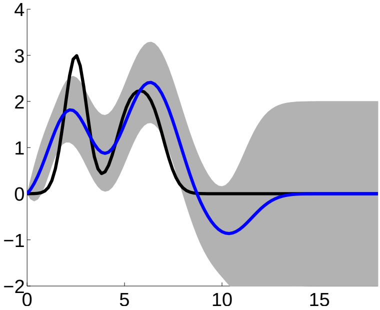
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



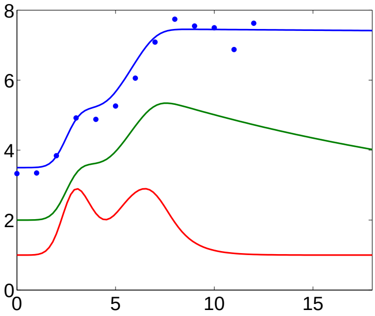
True "gene profiles" and noisy observations.



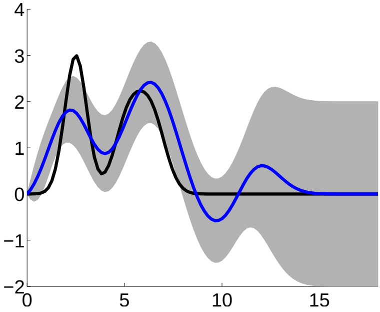
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



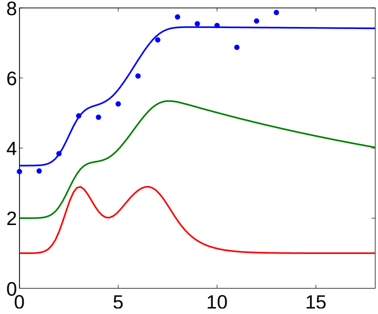
True "gene profiles" and noisy observations.



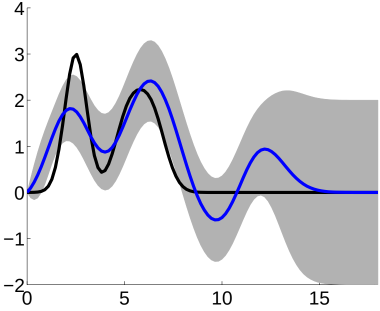
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



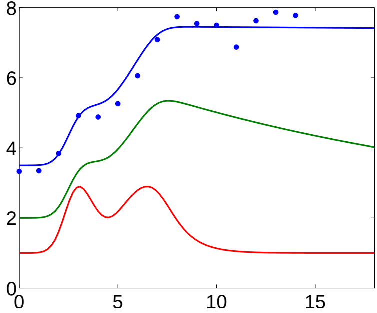
True "gene profiles" and noisy observations.



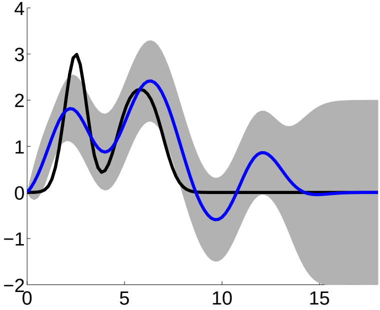
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



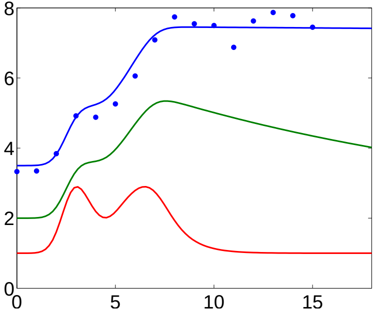
True “gene profiles” and noisy observations.



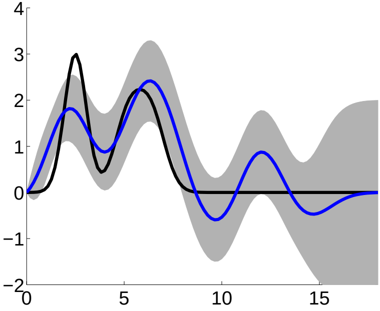
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



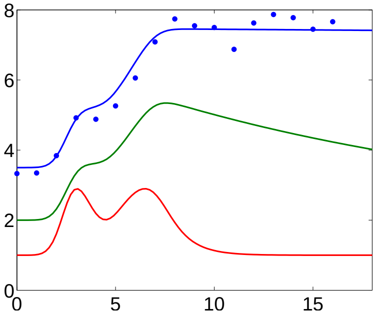
True "gene profiles" and noisy observations.



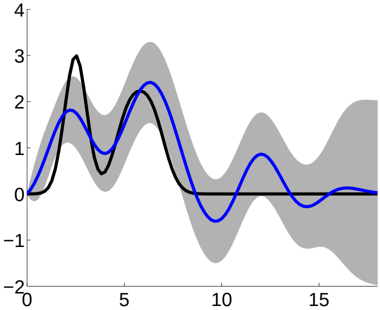
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



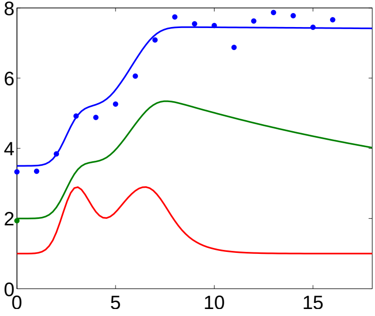
True “gene profiles” and noisy observations.



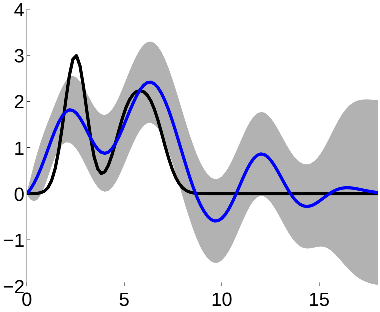
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



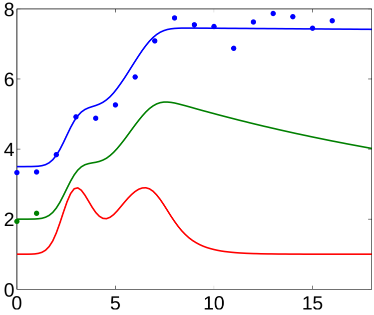
True “gene profiles” and noisy observations.



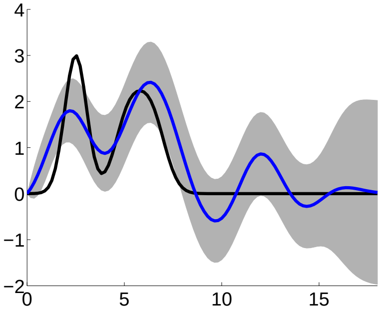
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



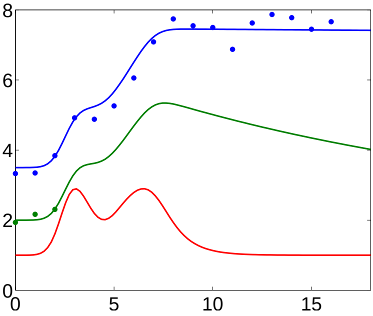
True “gene profiles” and noisy observations.



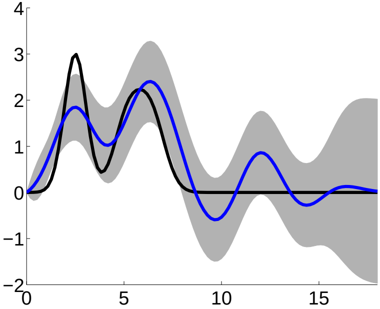
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



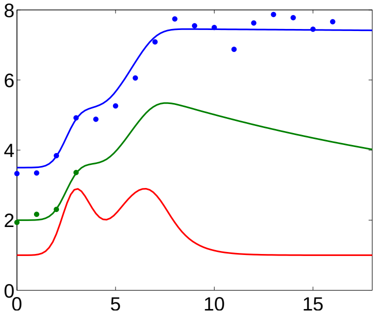
True “gene profiles” and noisy observations.



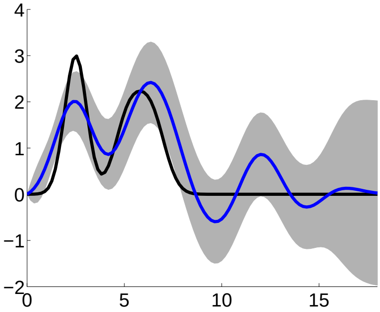
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



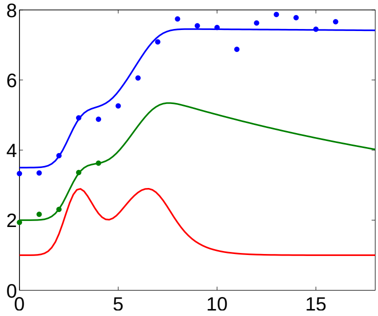
True "gene profiles" and noisy observations.



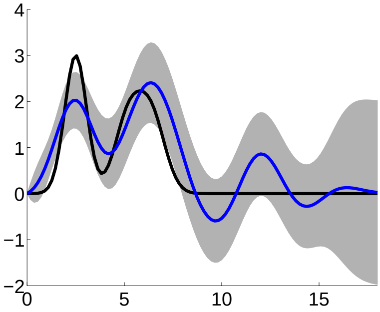
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



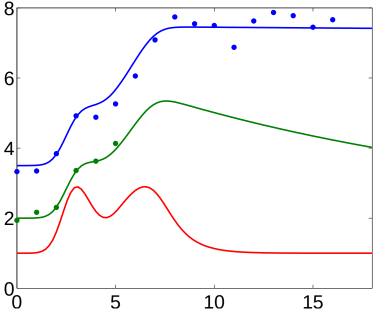
True "gene profiles" and noisy observations.



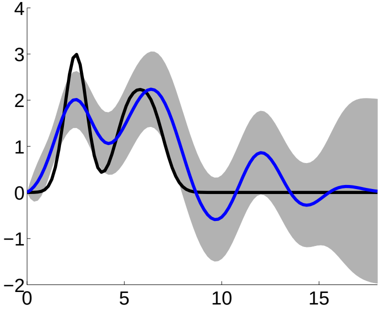
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



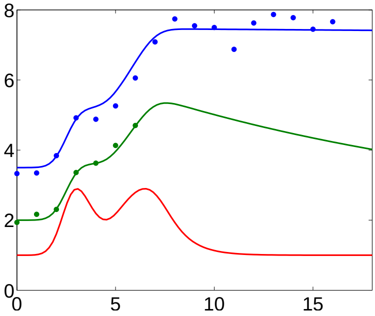
True “gene profiles” and noisy observations.



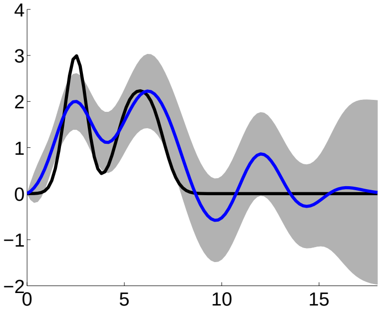
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



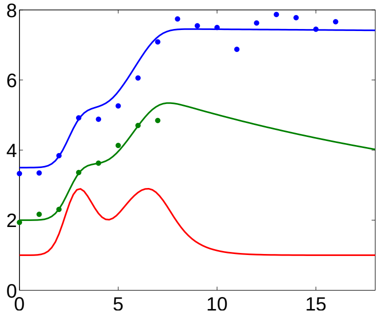
True “gene profiles” and noisy observations.



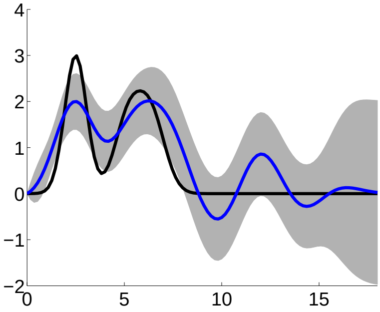
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



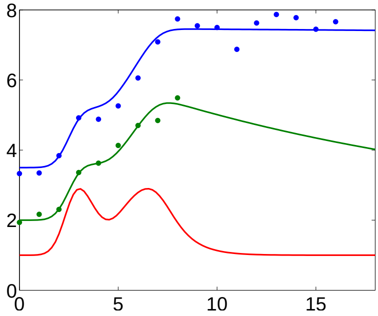
True “gene profiles” and noisy observations.



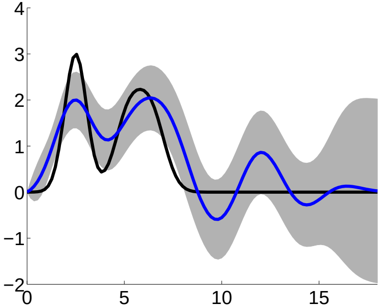
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



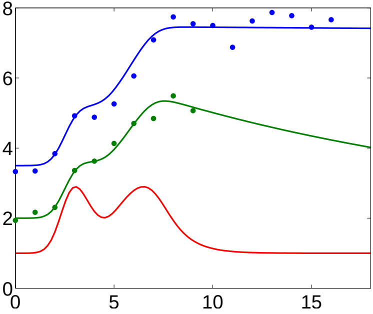
True "gene profiles" and noisy observations.



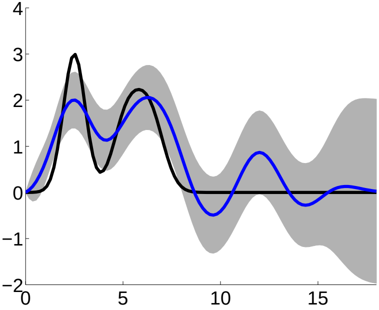
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



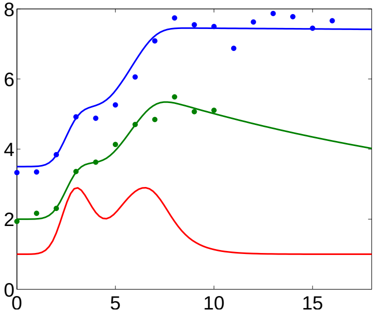
True "gene profiles" and noisy observations.



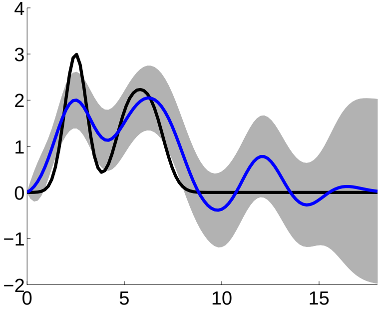
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



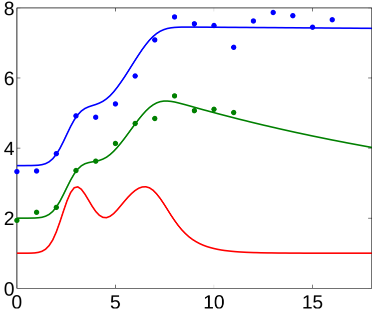
True “gene profiles” and noisy observations.



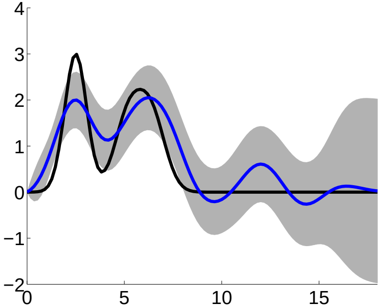
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



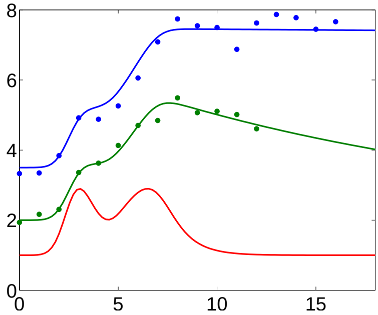
True "gene profiles" and noisy observations.



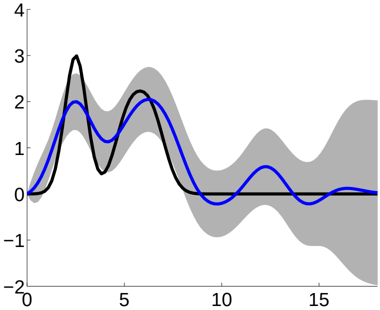
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



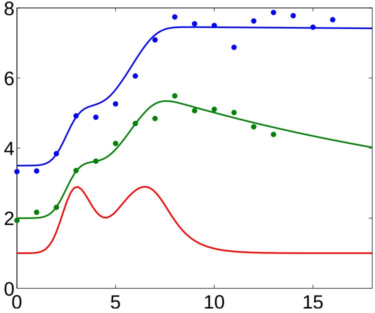
True "gene profiles" and noisy observations.



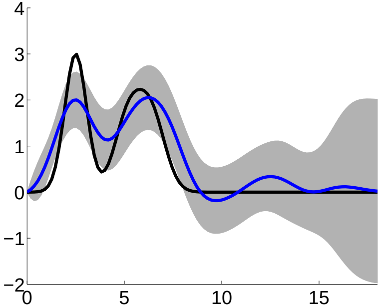
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



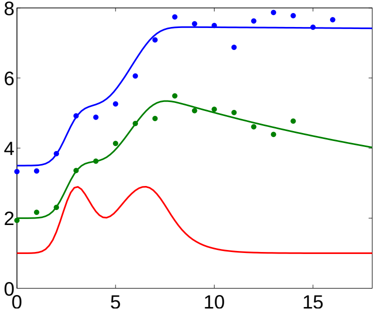
True “gene profiles” and noisy observations.



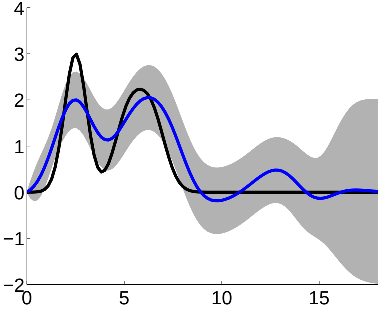
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



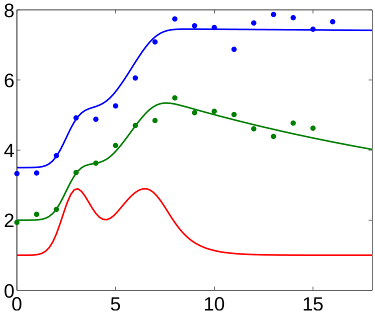
True "gene profiles" and noisy observations.



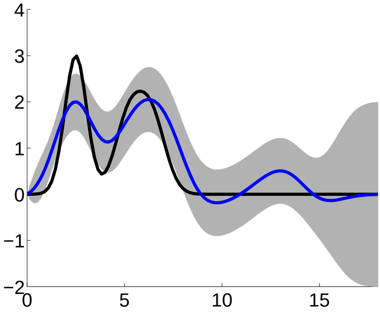
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



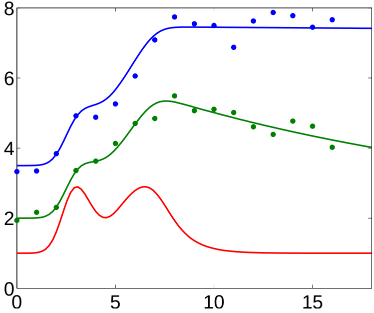
True "gene profiles" and noisy observations.



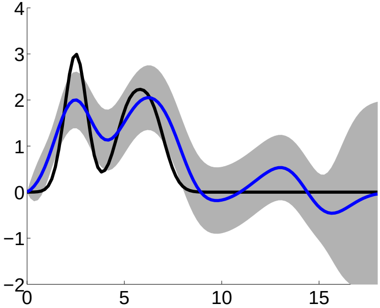
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



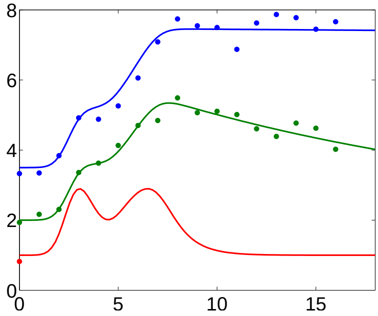
True “gene profiles” and noisy observations.



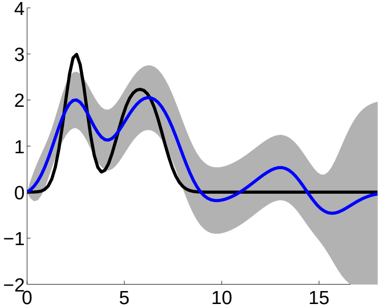
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



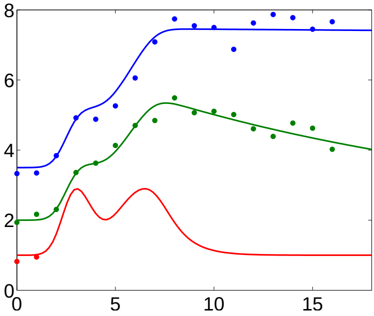
True “gene profiles” and noisy observations.



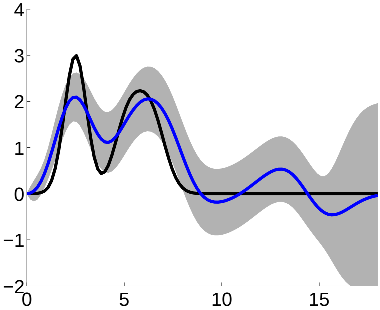
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



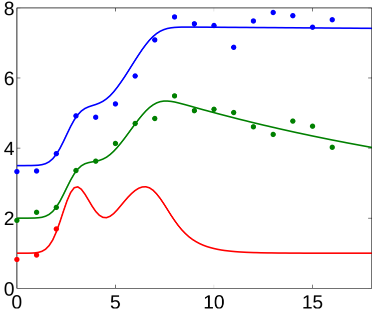
True “gene profiles” and noisy observations.



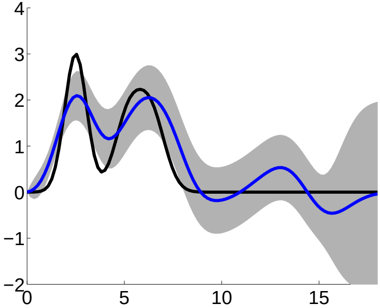
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



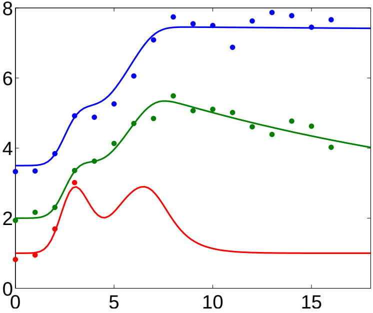
True "gene profiles" and noisy observations.



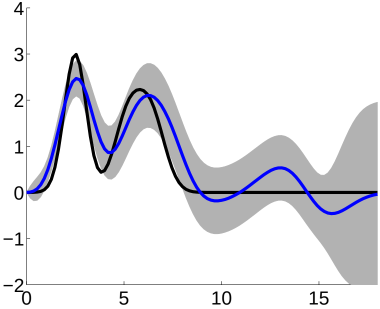
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



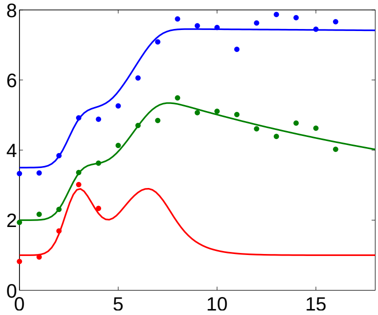
True "gene profiles" and noisy observations.



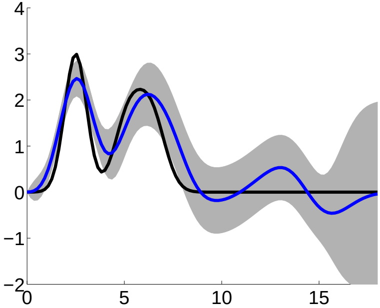
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



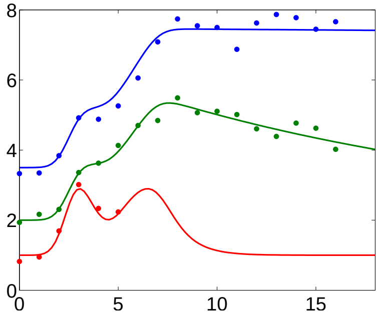
True "gene profiles" and noisy observations.



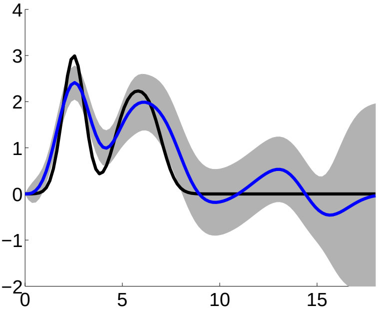
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



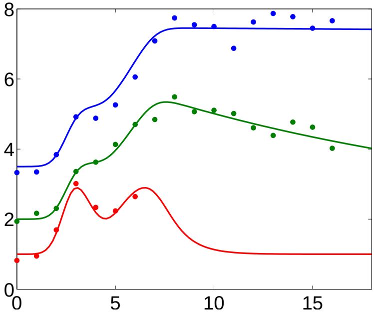
True "gene profiles" and noisy observations.



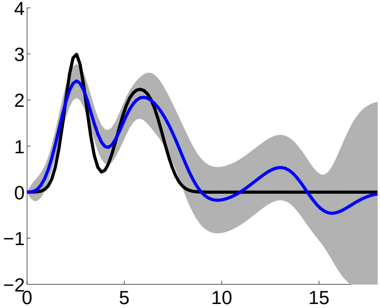
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



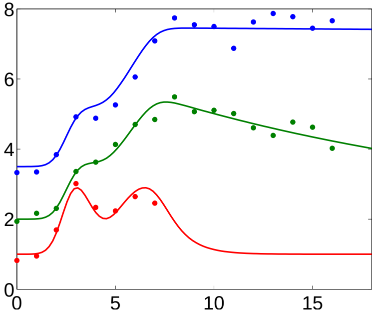
True "gene profiles" and noisy observations.



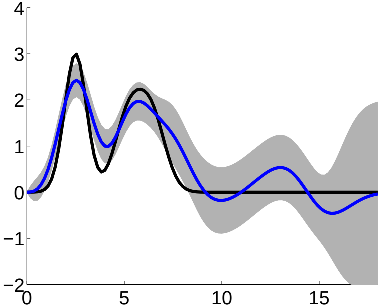
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



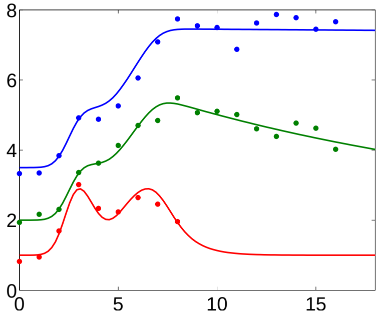
True "gene profiles" and noisy observations.



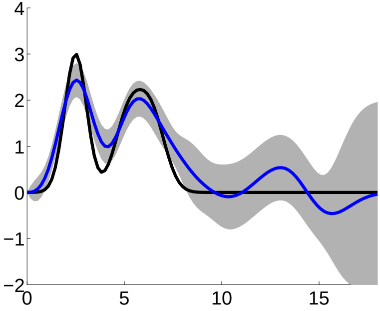
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



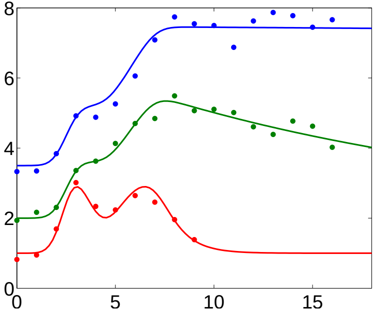
True "gene profiles" and noisy observations.



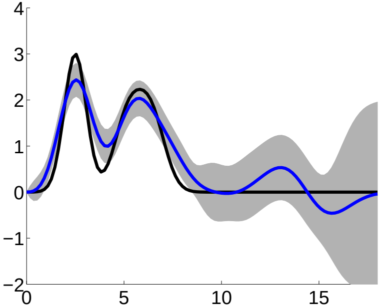
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



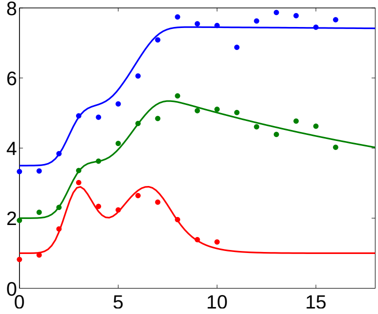
True "gene profiles" and noisy observations.



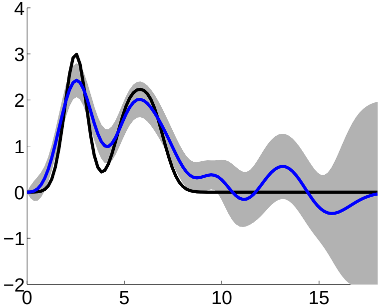
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



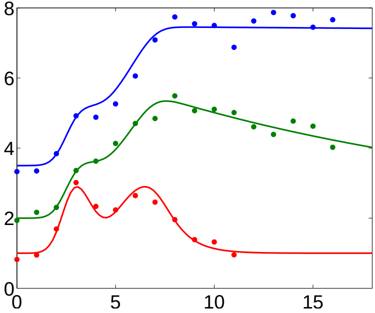
True “gene profiles” and noisy observations.



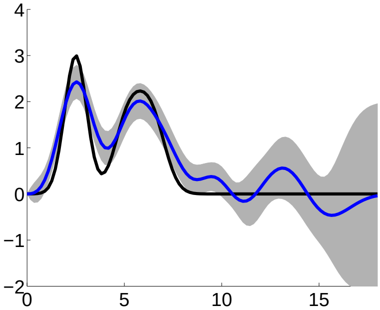
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



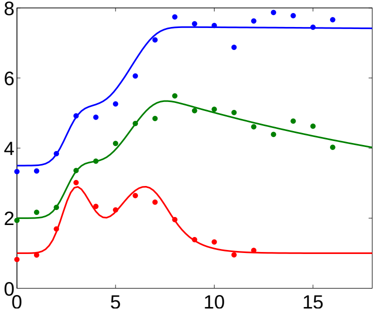
True “gene profiles” and noisy observations.



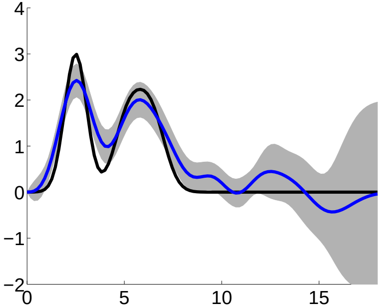
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



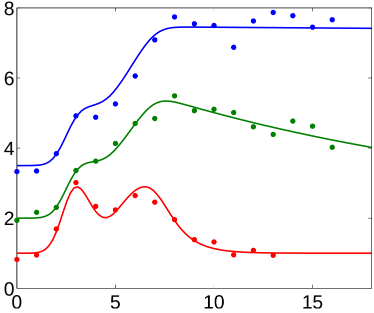
True "gene profiles" and noisy observations.



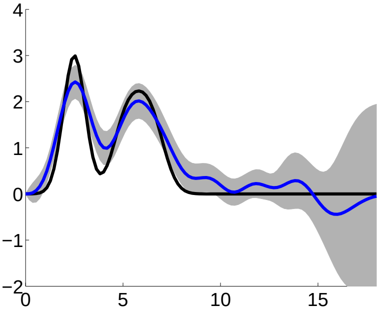
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



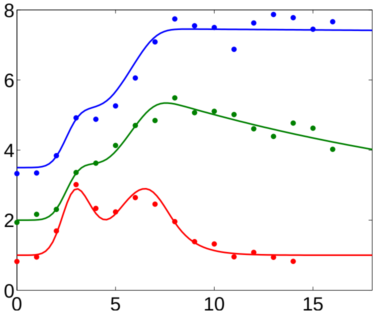
True “gene profiles” and noisy observations.



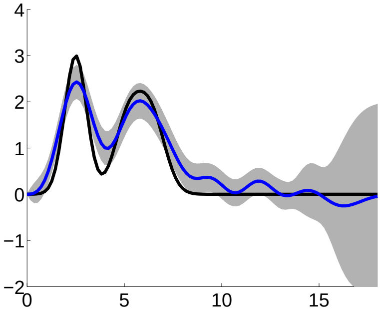
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



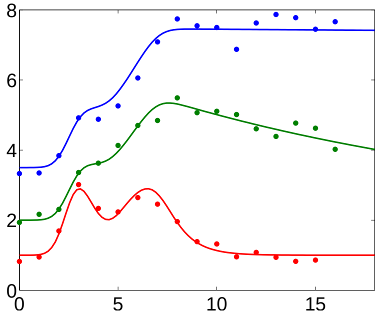
True "gene profiles" and noisy observations.



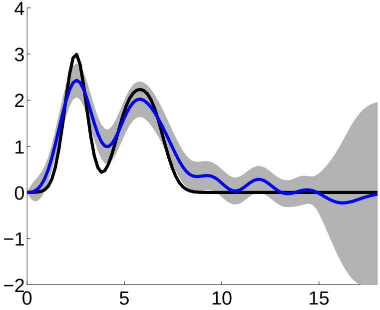
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



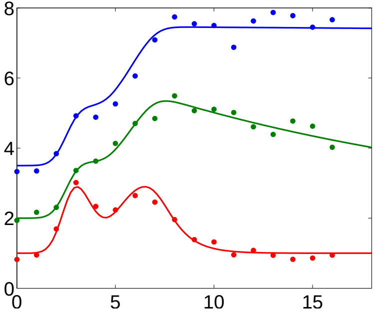
True "gene profiles" and noisy observations.



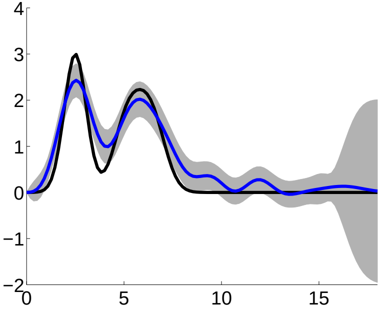
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.

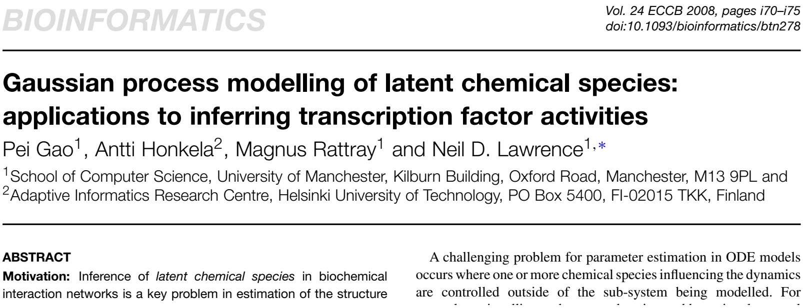


True "gene profiles" and noisy observations.



Inferred transcription factor activity.

p53 Results with GP



BIOINFORMATICS

Vol. 24 ECCB 2008, pages i70–i75
doi:10.1093/bioinformatics/btn278

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

Pei Gao[1], Antti Honkela[2], Magnus Rattray[1] and Neil D. Lawrence[1,*]

[1]School of Computer Science, University of Manchester, Kilburn Building, Oxford Road, Manchester, M13 9PL and
[2]Adaptive Informatics Research Centre, Helsinki University of Technology, PO Box 5400, FI-02015 TKK, Finland

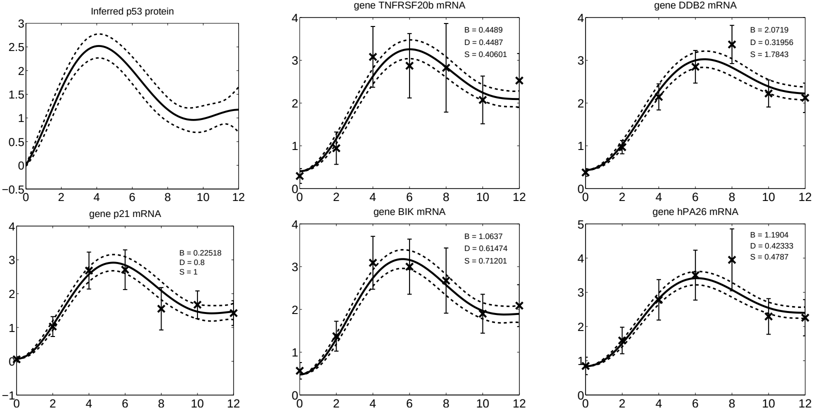
ABSTRACT

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)

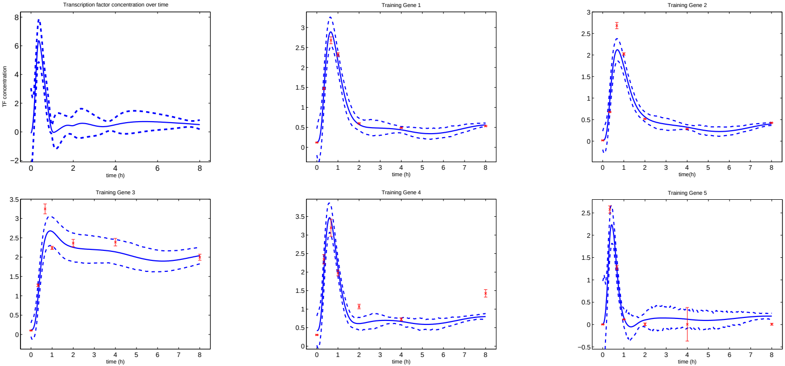


Ranking with ERK Signalling

- 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.

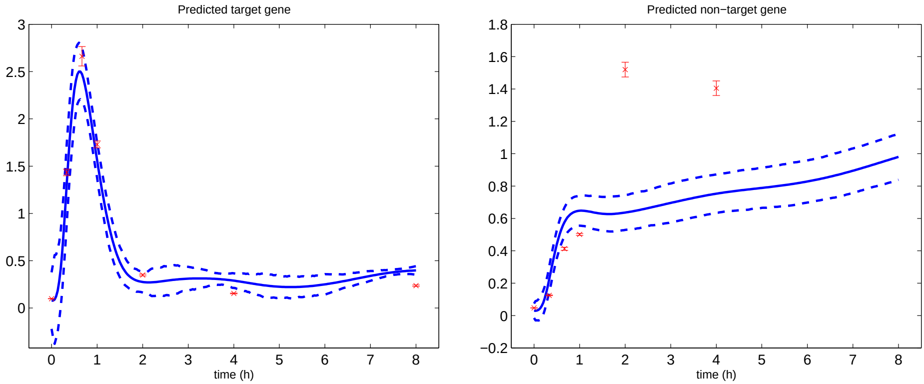
Elk-1 (MLP covariance)

Jennifer Withers



Elk-1 target selection

Fitted model used to rank potential targets of Elk-1



Outline

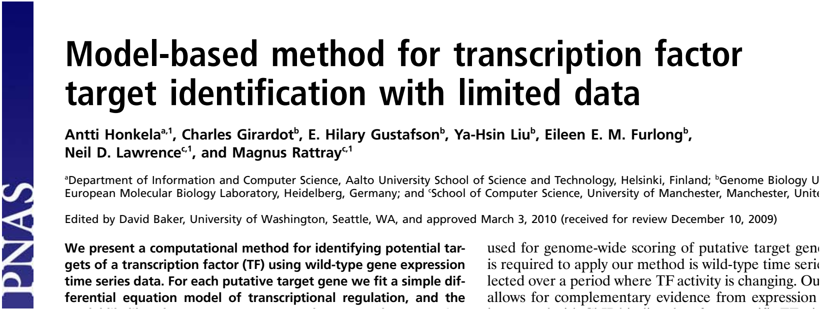
Motivation

Probabilistic Model for $f(t)$

Cascade Differential Equations

Discussion

Cascaded Differential Equations



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

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

[a]Department of Information and Computer Science, Aalto University School of Science and Technology, Helsinki, Finland; [b]Genome Biology U
European Molecular Biology Laboratory, Heidelberg, Germany; and [c]School of Computer Science, University of Manchester, Manchester, Unite

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 gen
is required to apply our method is wild-type time seri
lected over a period where TF activity is changing. Ou
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 triplobastic 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

$$\frac{\mathrm{d}f\left(t\right)}{\mathrm{d}t} = \sigma y\left(t\right) - \delta f\left(t\right)$$
$$\frac{\mathrm{d}x_j\left(t\right)}{\mathrm{d}t} = b_j + s_j f\left(t\right) - d_j x_j\left(t\right)$$

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

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

Covariance for Translation/Transcription Model

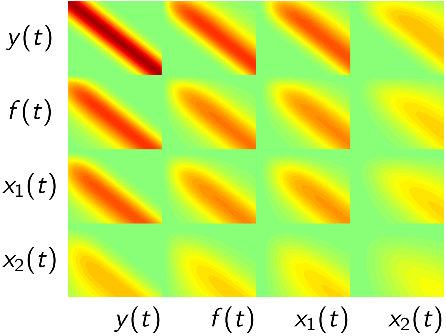
RBF covariance function for $y(t)$

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

$$x_i(t) = \frac{b_i}{d_i} + s_i \exp(-d_i t) \int_0^t f(u) \exp(d_i u) \, \mathrm{d}u.$$

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

δ	d_1	s_1	d_2	s_2
1	5	5	0.5	0.5



Joint Sampling of $y(t)$, $f(t)$, and $x(t)$

- `disimSample`

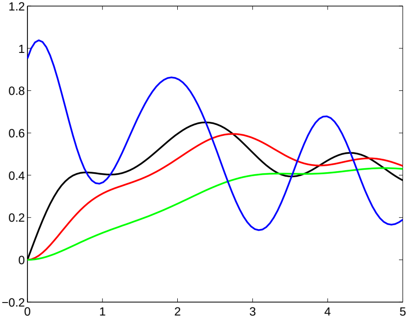


Figure: Joint samples from the ODE covariance, *blue*: $y(t)$ (mRNA of TF), *black*: $f(t)$ (TF concentration), *red*: $x_1(t)$ (high decay target) and *green*: $x_2(t)$ (low decay target)

Joint Sampling of $y(t)$, $f(t)$, and $x(t)$

- `disimSample`

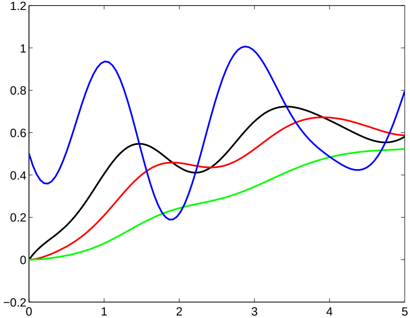


Figure: Joint samples from the ODE covariance, *blue*: $y(t)$ (mRNA of TF), *black*: $f(t)$ (TF concentration), *red*: $x_1(t)$ (high decay target) and *green*: $x_2(t)$ (low decay target)

.

Joint Sampling of $y(t)$, $f(t)$, and $x(t)$

- `disimSample`

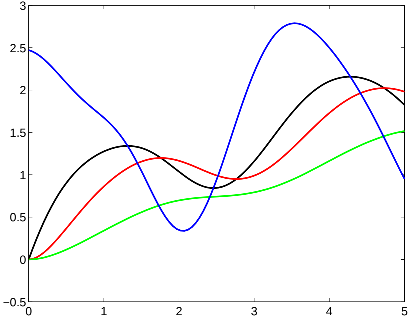


Figure: Joint samples from the ODE covariance, *blue*: $y(t)$ (mRNA of TF), *black*: $f(t)$ (TF concentration), *red*: $x_1(t)$ (high decay target) and *green*: $x_2(t)$ (low decay target)

.

Joint Sampling of $y(t)$, $f(t)$, and $x(t)$

- `disimSample`

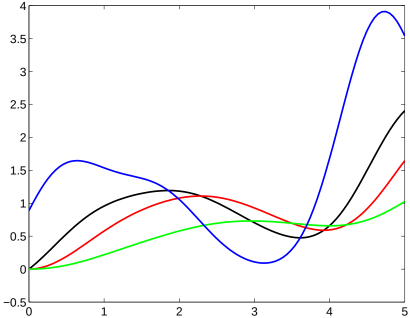


Figure: Joint samples from the ODE covariance, *blue*: $y(t)$ (mRNA of TF), *black*: $f(t)$ (TF concentration), *red*: $x_1(t)$ (high decay target) and *green*: $x_2(t)$ (low decay target)

.

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

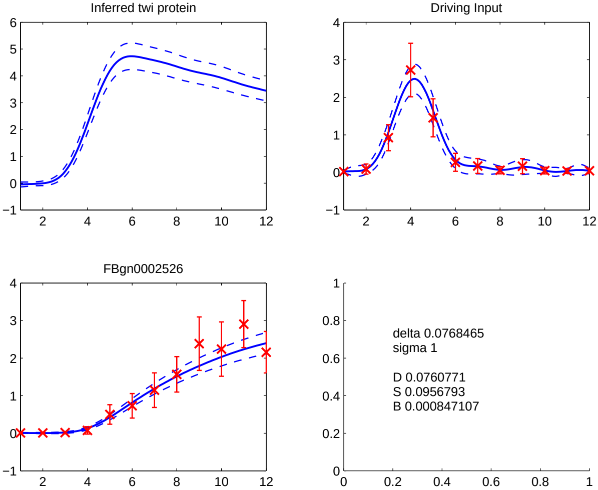


Figure: Model for flybase gene identity FBgn0002526.

Results for Twi using the Cascade model

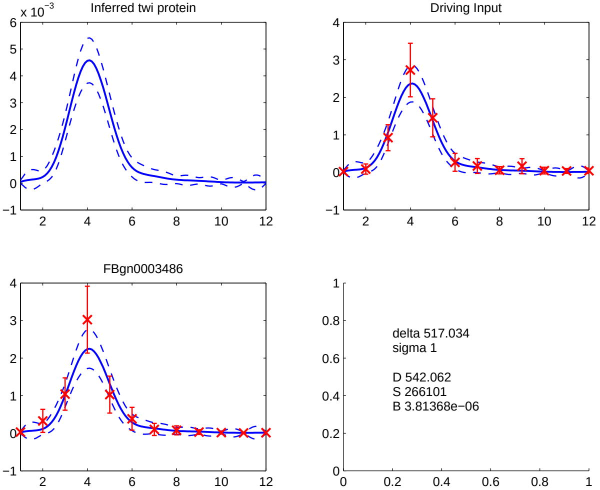
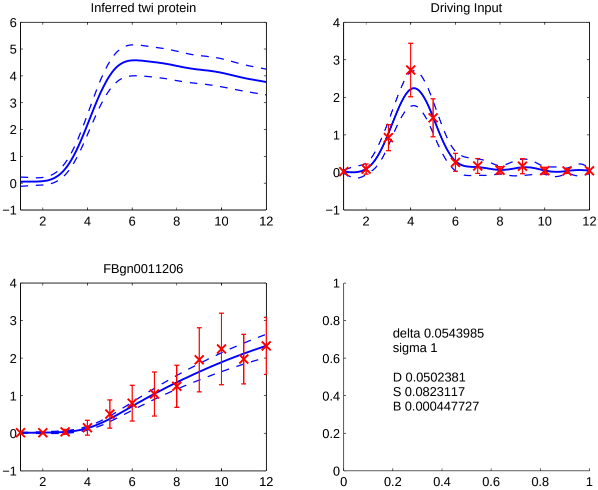


Figure: Model for flybase gene identity FBgn0003486.

Results for Twi using the Cascade model



delta 0.0543985
sigma 1

D 0.0502381
S 0.0823117
B 0.000447727

Figure: Model for flybase gene identity FBgn0011206.

Results for Twi using the Cascade model

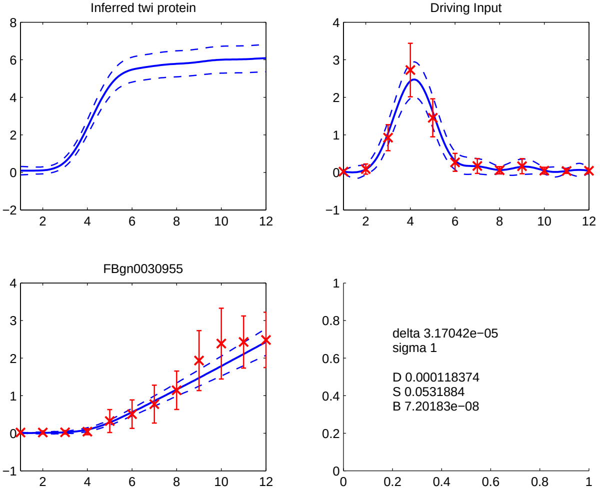


Figure: Model for flybase gene identity FBgn00309055.

Results for Twi using the Cascade model

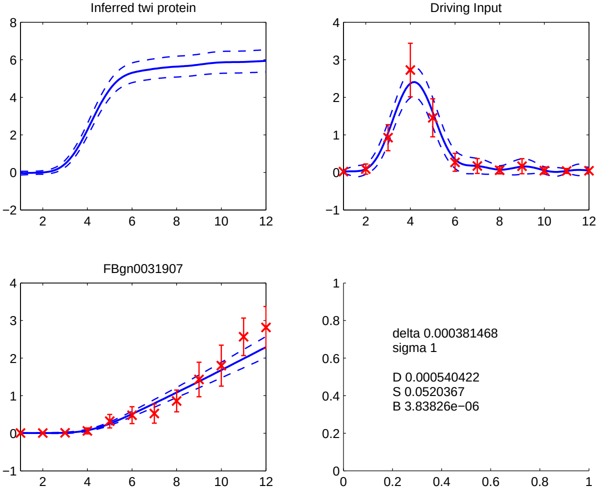
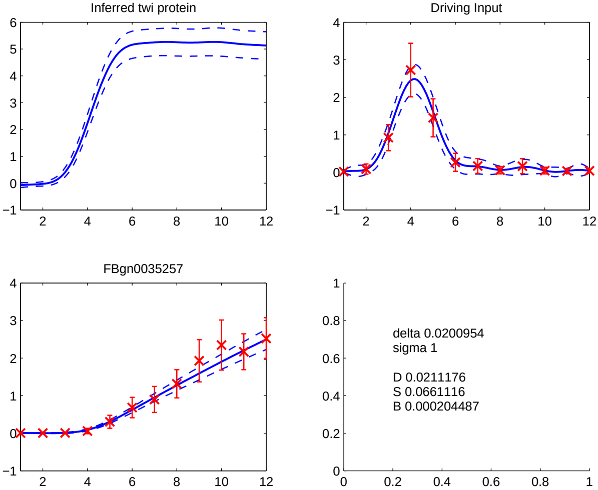


Figure: Model for flybase gene identity FBgn0031907.

Results for Twi using the Cascade model


Inferred twi protein

Driving Input

FBgn0035257

delta 0.0200954
sigma 1

D 0.0211176
S 0.0661116
B 0.000204487

Figure: Model for flybase gene identity FBgn0035257.

Results for Twi using the Cascade model

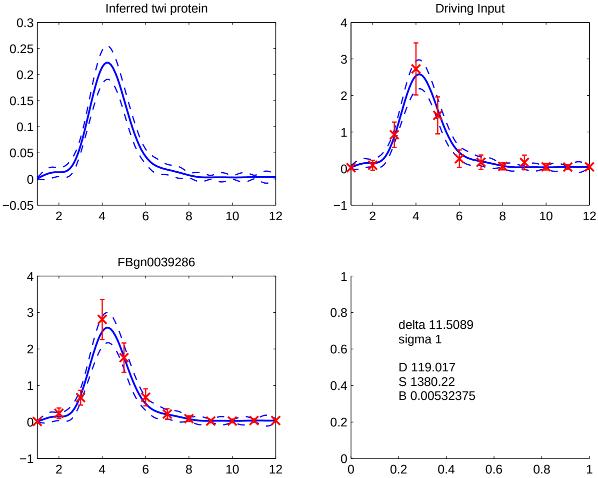
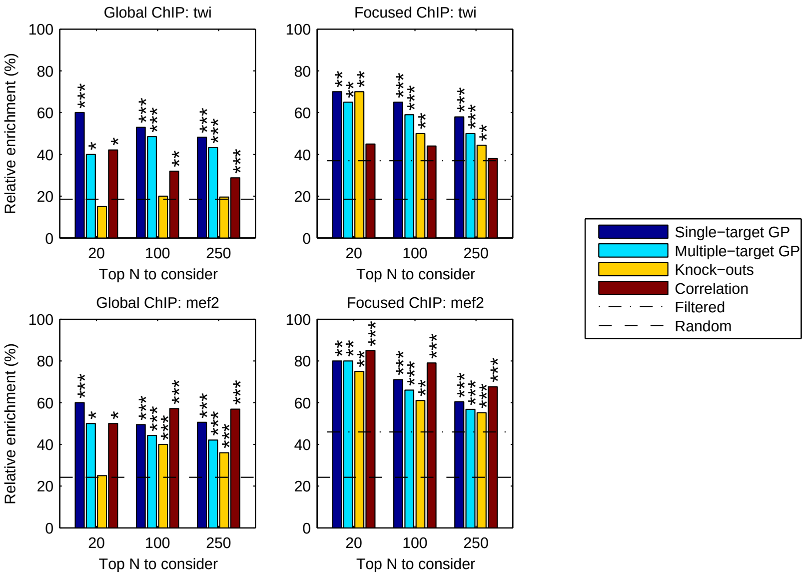


Figure: Model for flybase gene identity FBgn0039286.

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

Discussion and Future Work

- Integration of probabilistic inference with mechanistic models.
- Applications in modeling gene expression.
- Cascade model introduces model of translation.
- Challenges:
    - Non linear response and non linear differential equations.
    - Scaling up to larger systems.
    - Stochastic differential equations.

Acknowledgements

- Investigators: Neil Lawrence and Magnus Rattray
- Researchers: Pei Gao, Antti Honkela, Guido Sanguinetti, and Jennifer Withers
- Martino Barenco and Mike Hubank at the Institute of Child Health in UCL (p53 pathway).
- Charles Girardot and Eileen Furlong of EMBL in Heidelberg (mesoderm development in *D. Melanogaster*).

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

References I

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

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

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

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].

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

Y. Lazebnik. Can a biologist fix a radio? or, what I learned while studying apoptosis. *Cancer Cell*, 2:179–182, 2002.

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

J. Oakley and A. O'Hagan. Bayesian inference for the uncertainty distribution of computer model outputs. *Biometrika*, 89(4):769–784, 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].

Outline

Experimental Structure of Arrays

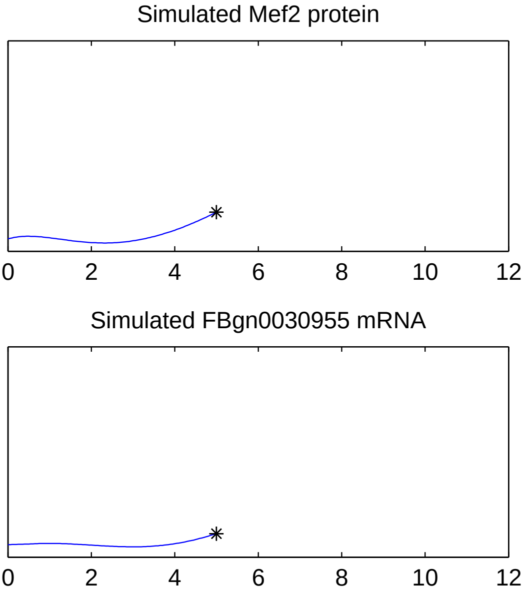
Nonlinear Response

Molecular biology time series

Antti Honkela

- Biological systems are dynamic, observing their time evolution very helpful
- Time series measurements of gene expression, protein activity, protein binding, ...
- Problem: most of these assays are highly disruptive to the sample
- Therefore: time series = series of independent experiments run for different lengths of time
- This has implications for modelling...

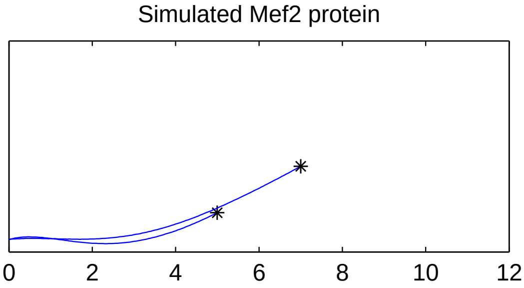
Simulated molecular biology time series

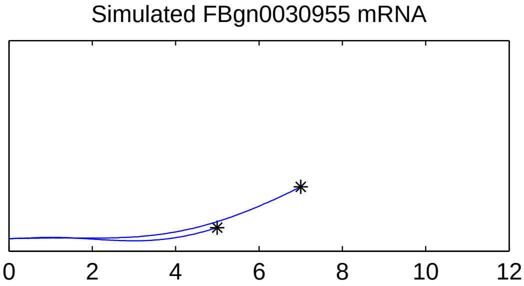


Simulated Mef2 protein

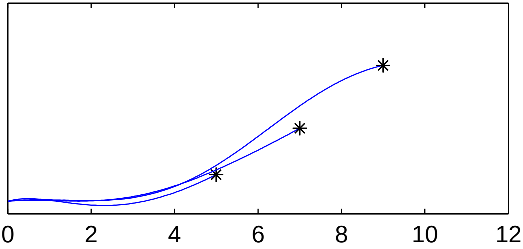
Simulated FBgn0030955 mRNA

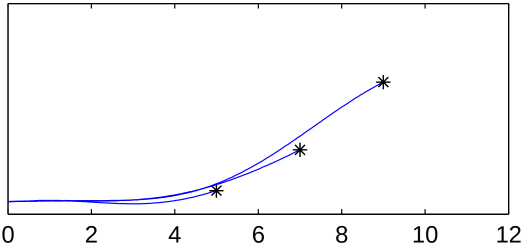
Simulated molecular biology time series



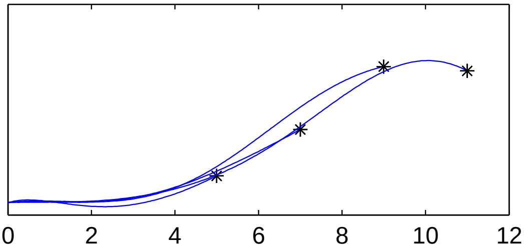


Simulated molecular biology time series

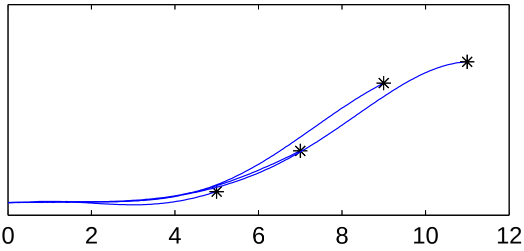

Simulated Mef2 protein


Simulated FBgn0030955 mRNA

Simulated molecular biology time series

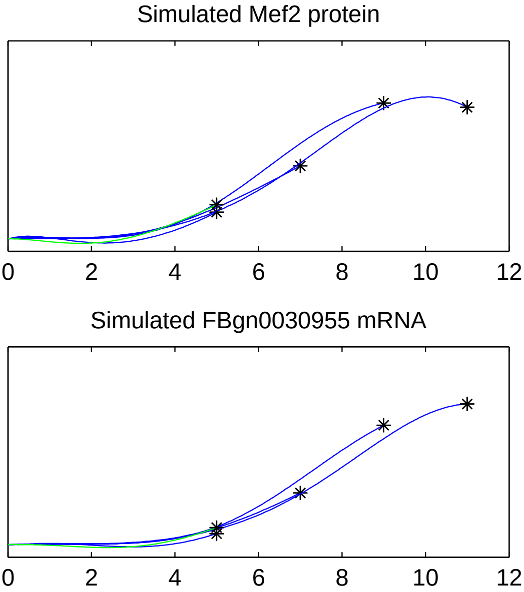


Simulated Mef2 protein

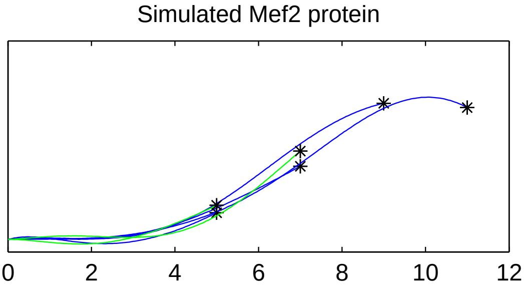


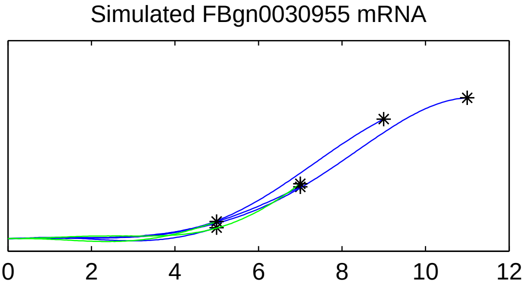
Simulated FBgn0030955 mRNA

Simulated molecular biology time series



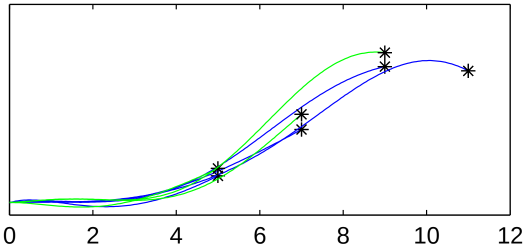
Simulated molecular biology time series


Simulated Mef2 protein

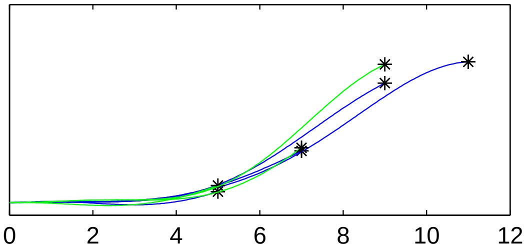

Simulated FBgn0030955 mRNA

Simulated molecular biology time series

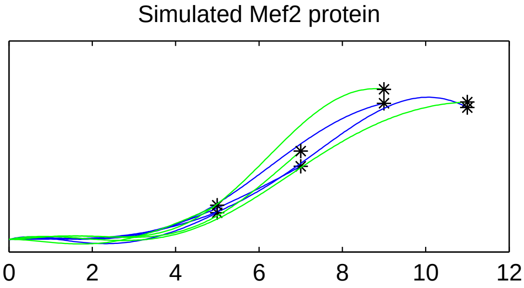
Simulated Mef2 protein

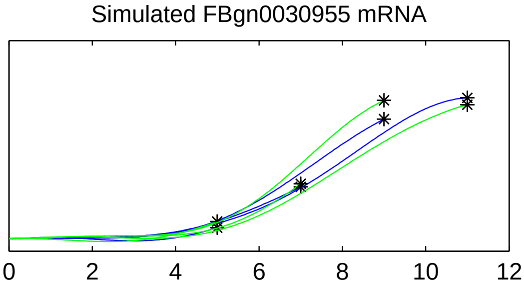


Simulated FBgn0030955 mRNA

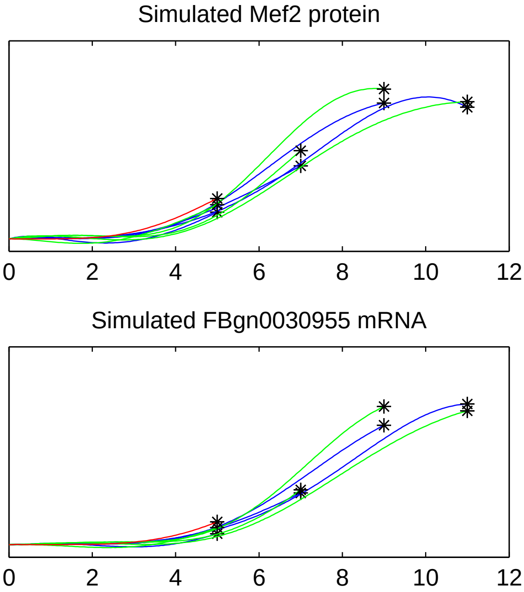


Simulated molecular biology time series

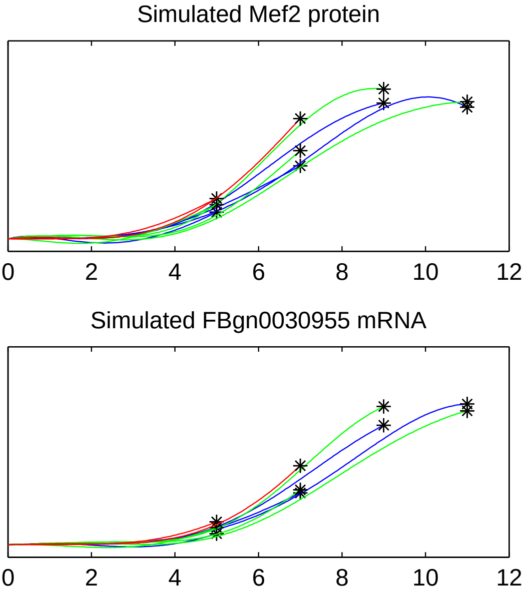




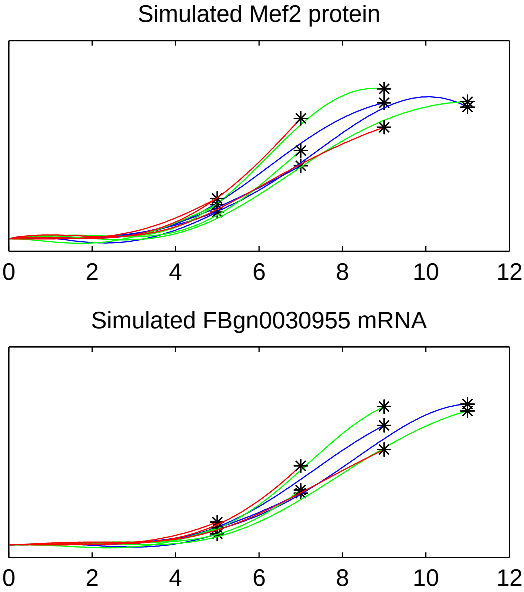
Simulated molecular biology time series



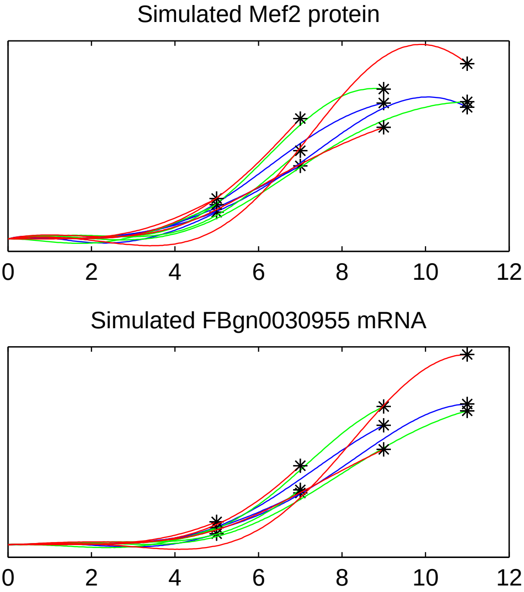
Simulated molecular biology time series



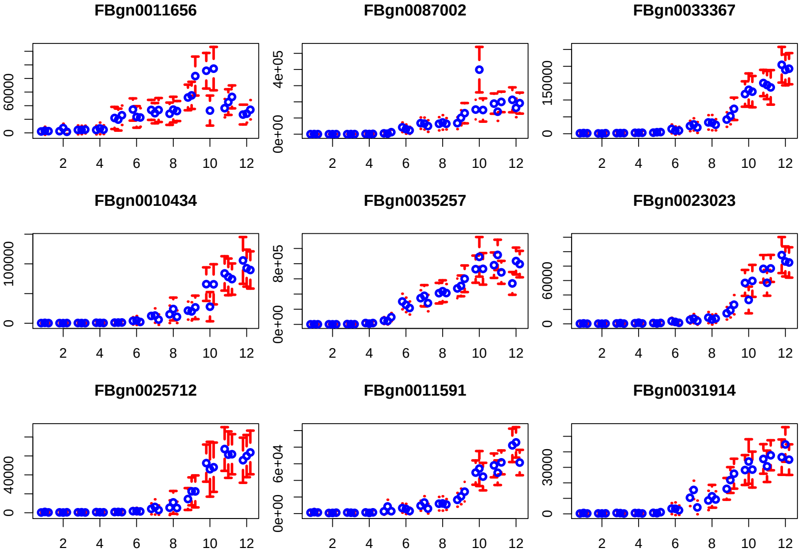
Simulated molecular biology time series



Simulated molecular biology time series



Real gene expression time series



Example model: Linear ODE model of transcription

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

$$\frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = b_j + s_j f(t) - d_j x_j(t)$$

- $x_j(t)$ – concentration of gene j's mRNA
- $f(t)$ – concentration of active transcription factor
- Model parameters: baseline b_j, sensitivity s_j and decay d_j
- Placing a Gaussian process (GP) prior on $f(t)$ leads to a joint GP over all concentration profiles (Gao et al., 2008, Bioinformatics)

How to connect the model to data?

1. Assume independent profiles for each complete (biological) repeat
    - Loses statistical power for extra independence assumptions
    - Is it meaningful to order the repeats?
2. Assume one shared underlying profile with independent observations
    - Potentially sensitive to outliers

Exchangeability analysis

Assume $x_j^k(t_i)$ observation of kth repeat of jth gene at ith time

	$x_{:}^{k}(t_i) \leftrightarrow x_{:}^{k'}(t_i)$ “swap arrays”	$x_j^{k}(t_i) \leftrightarrow x_j^{k'}(t_i)$ “swap single gene”
“Reality”	Yes	No
1. Independent profiles	No	No
2. Shared profile	Yes	Yes

Solution: hierarchical GP model

- Assume the underlying $f(t)$ is composed of a shared and an experiment-specific part $f_{ik}(t)$

$$\frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = b_j + s_j[f_{\mathrm{shared}}(t) + f_{ik}(t)] - d_j x_j(t)$$

- Covariance is of the same form as usual
- Introduces additional covariance terms for measurements from the same experiment
- Alternative parametrisations of variance of $f_{ik}(t)$
    - Shared across all experiments
    - Sampled independently for each experiment

Exchangeability analysis revisited

Assume $x_j^k(t_i)$ observation of kth repeat of jth gene at ith time

	$x_:^k(t_i) \leftrightarrow x_:^{k'}(t_i)$ "swap arrays"	$x_j^k(t_i) \leftrightarrow x_j^{k'}(t_i)$ "swap single gene"
"Reality"	Yes	No
1. Independent profiles	No	No
2. Shared profile	Yes	Yes
3. Hierarchical model	Yes	No

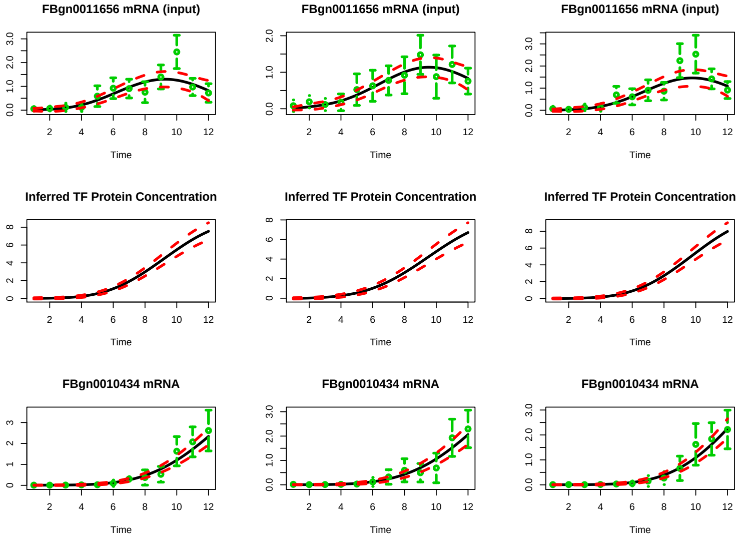
ODE model of translation and transcription

- Assume TF is transcriptionally regulated with related mRNA $y(t)$
- This yields a system of ODEs (Gao et al., 2008)

$$\begin{aligned}\frac{\mathrm{d}f(t)}{\mathrm{d}t} &= \sigma y(t) - \delta f(t) \\ \frac{\mathrm{d}x_j(t)}{\mathrm{d}t} &= b_j + s_j f(t) - d_j x_j(t)\end{aligned}$$

- The corresponding GP model can be derived analogously to the previous case

Independent profiles

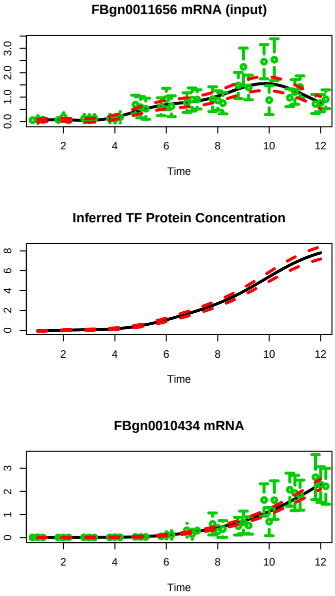


FBgn0011656 mRNA (input) FBgn0011656 mRNA (input) FBgn0011656 mRNA (input)

Inferred TF Protein Concentration Inferred TF Protein Concentration Inferred TF Protein Concentration

FBgn0010434 mRNA FBgn0010434 mRNA FBgn0010434 mRNA

Hierarchical model



Outline

Experimental Structure of Arrays

Nonlinear Response

Nonlinear Response Models

Consider the model of transcription,

$$\frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = b_j + s_j g(f(t)) - d_j x_j(t),$$

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

$$x_j(t) = \frac{b_j}{d_j} + s_j \int_0^t e^{-d_j(t-u)} g_j(f(u))\,\mathrm{d}u$$

MAP-Laplace Approximation

Laplace's method: approximate posterior mode as Gaussian

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

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

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

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

MAP-Laplace Approximation

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

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

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

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

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

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

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

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

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

Model Parameter Estimation

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

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

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

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

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

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

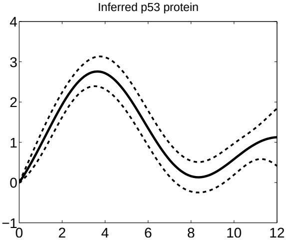
Michaelis-Menten Kinetics

Pei Gao

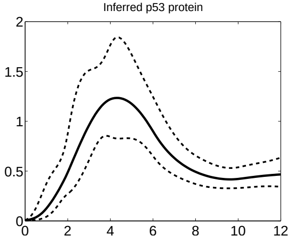
- The Michaelis-Menten activation model uses the following non-linearity

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

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



(a) Linear Response



(b) Laplace Approximation Nonlinear

Valdiation of Laplace Approximation

Michalis Titsias

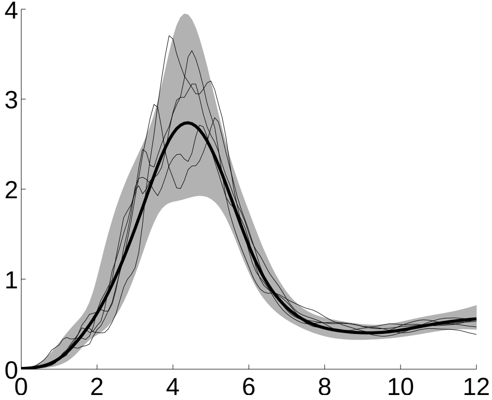


Figure: Laplace approximation error bars along with samples from the true posterior distribution.

SOS Response

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

LexA Experimental Description

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

Khanin et al. Model

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

$$x_i\left(t\right) = x_i^0 e^{-d_i t} + \frac{b_i}{d_i} + s_i e^{-d_i t} \int_0^t g(f\left(u\right)) e^{d_i u} \mathrm{d}u.$$

$$x_i\left(t\right) = x_i^0 e^{-d_i t} + \frac{b_i}{d_i} + s_i e^{-d_i t} \frac{1}{t_{j+1} - t_j} \sum_{j=0}^{N-2} g(\bar{f}_j) \left(e^{d_i t_{j+1}} - e^{d_i t_j}\right)$$

where $\bar{f}_j = \frac{\left(f\left(t_j\right) + f\left(t_{j+1}\right)\right)}{2}$ on each subinterval $\left(t_j, t_j + 1\right), j = 0, \ldots, N-2.$ This is under the simplifying assumption that $f\left(t\right)$ is a piece-wise constant function on each subinterval $\left(t_j, t_j + 1\right).$ Repression model: $g(f(t)) = \frac{1}{\gamma + e^{f(t)}}$.

Khanin et al. Results

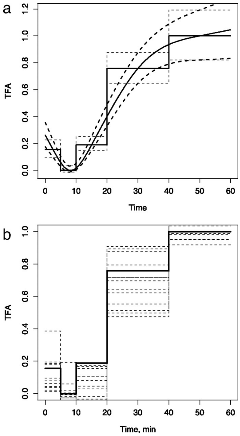


Figure: Fig. 2 from Khanin et al. (2006): Reconstructed activity level of master repressor LexA, following a UV dose of 40 J/m2.

Khanin et al. Results

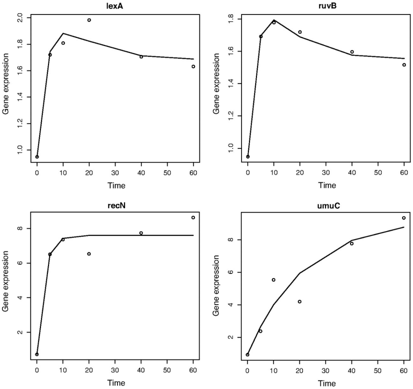


Figure: Fig. 3 from Khanin et al. (2006): Reconstructed profiles for four genes in the LexA SIM.

Repression Model

Pei Gao

- We can use the same model of repression,

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

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

$$x_j\left(t\right) = \alpha_j e^{-d_j t} + \frac{b_j}{d_j} + s_j \int_0^t e^{-d_j(t-u)} g_j(f\left(u\right))\mathrm{d}u$$

Results for the repressor LexA

Pei Gao

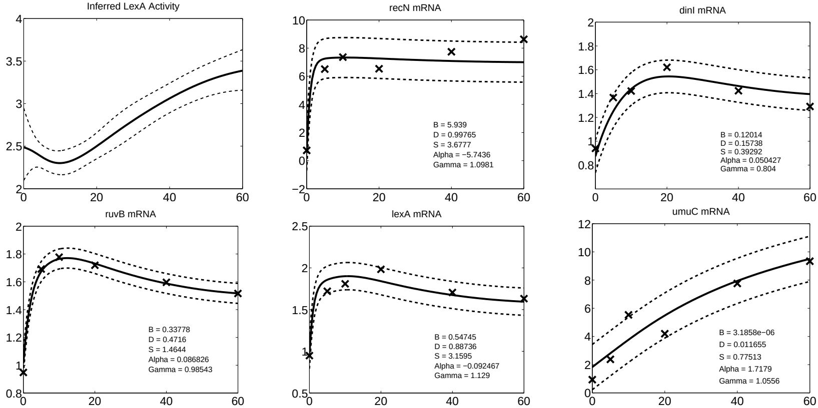


Figure: Our results using an MLP kernel. From Gao et al. (2008).

Use Samples to Represent Posterior

Michalis Titsias

- Sample in Gaussian processes

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

- Likelihood relates GP to data through

$$x_j\left(t\right) = \alpha_j e^{-d_j t} + \frac{b_j}{d_j} + s_j \int_0^t e^{-d_j(t-u)} g_j(f\left(u\right)) \mathrm{d}u$$

- We use *control points* for fast sampling.

MCMC for Non Linear Response

The Metropolis-Hastings algorithm

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

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

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

Sampling using control points

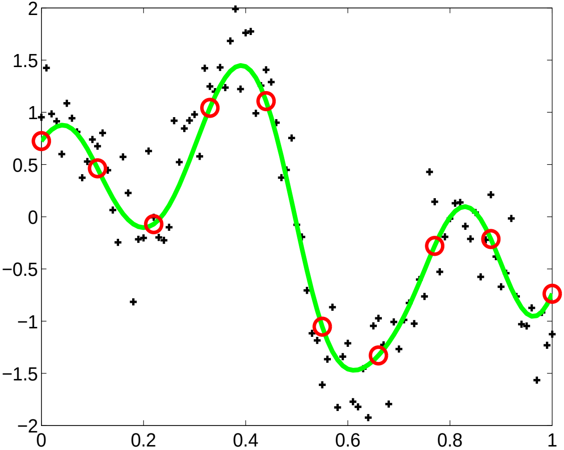
- Separate the points in $\mathbf{f}$ into two groups:
  - few control points $\mathbf{f}_c$
  - and the large majority of the remaining points $\mathbf{f}_\rho = \mathbf{f} \setminus \mathbf{f}_c$
- Sample the control points $\mathbf{f}_c$ using a proposal $q\left(\mathbf{f}_c^{(t+1)} | \mathbf{f}_c^{(t)}\right)$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

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

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

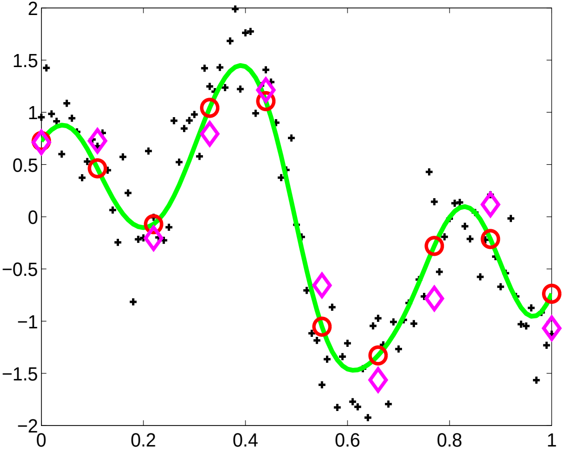
Sampling using control points: Regression-Examples

Sample 121 points using 10 control points



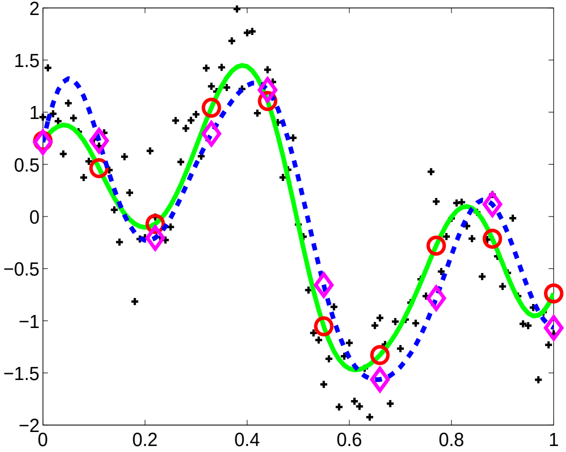
Sampling using control points: Regression-Examples

Sample 121 points using 10 control points



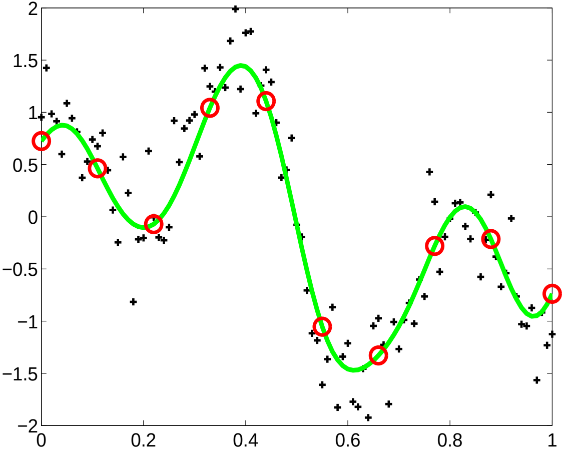
Sampling using control points: Regression-Examples

Sample 121 points using 10 control points



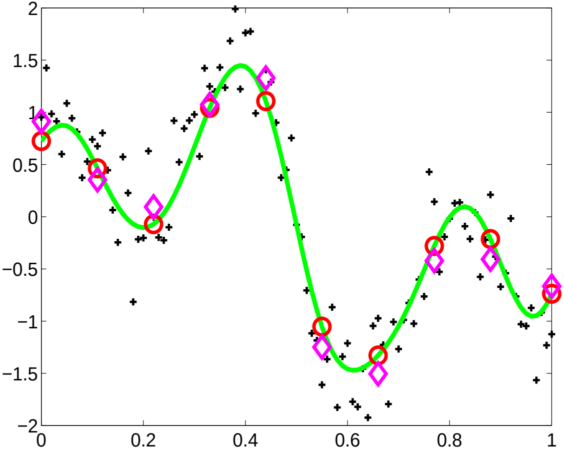
Sampling using control points: Regression-Examples

Sample 121 points using 10 control points



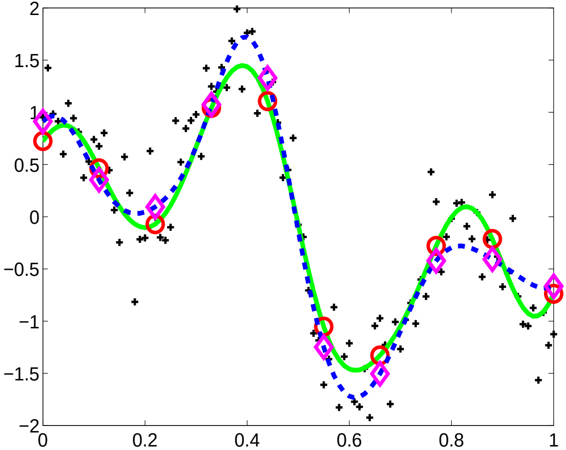
Sampling using control points: Regression-Examples

Sample 121 points using 10 control points



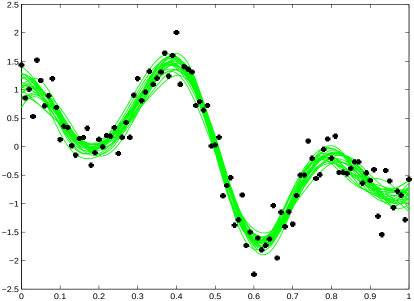
Sampling using control points: Regression-Examples

Sample 121 points using 10 control points



Sampling using control points

Few samples drawn during MCMC



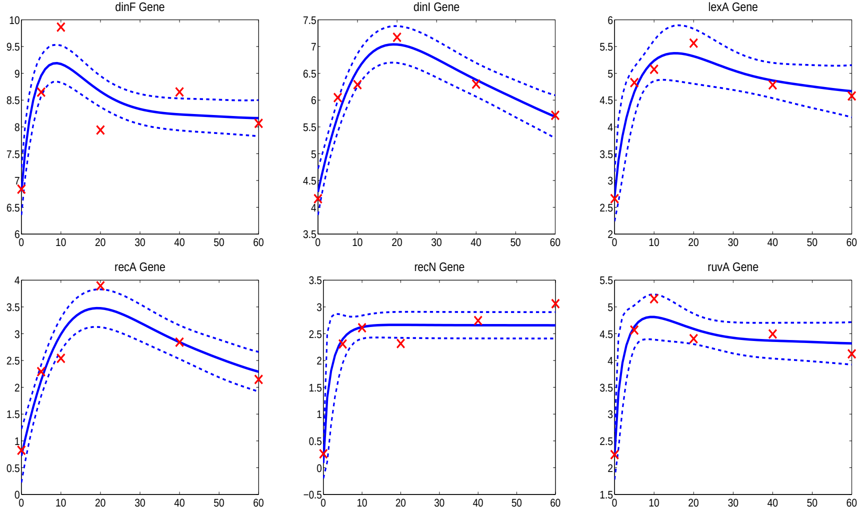
Results on SOS System

- Again consider the Michaelis-Menten kinetic equation

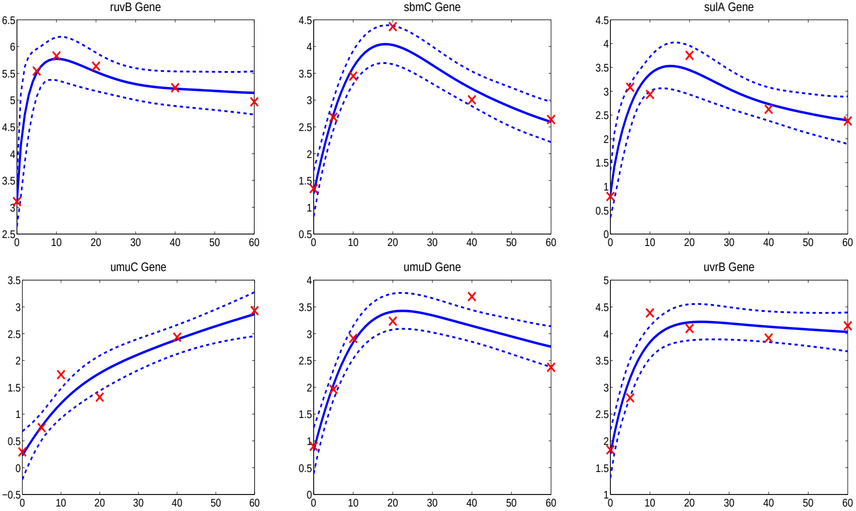
$$\frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = b_j + s_j \frac{1}{\exp(f(t)) + \gamma_j} - d_j x_j(t)$$

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

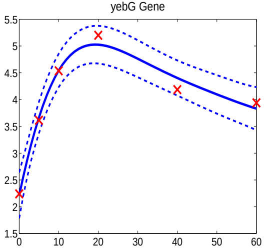
Results in E.coli data: Predicted gene expressions

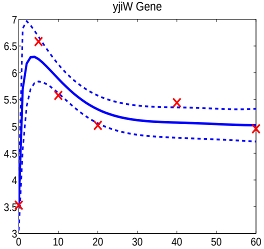


Results in E.coli data: Predicted gene expressions

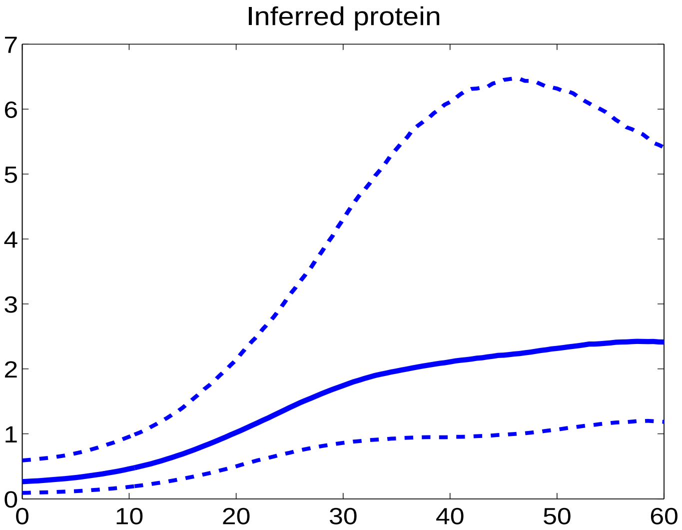


Results in E.coli data: Predicted gene expressions

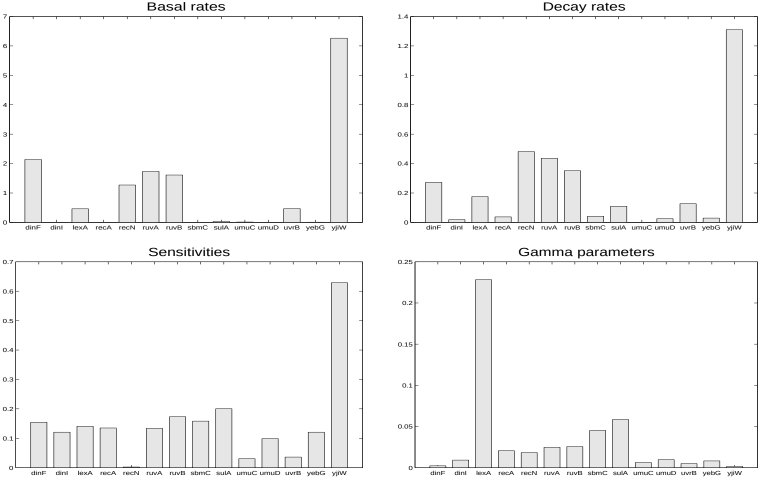




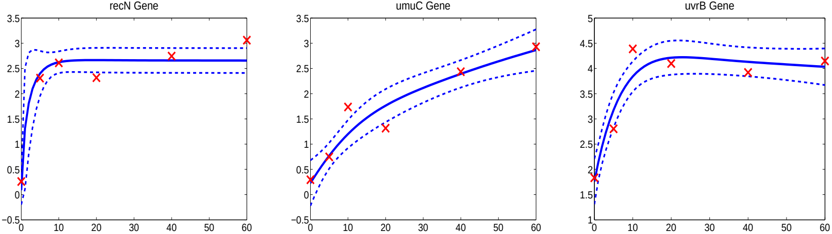
Results in E.coli data: Protein concentration

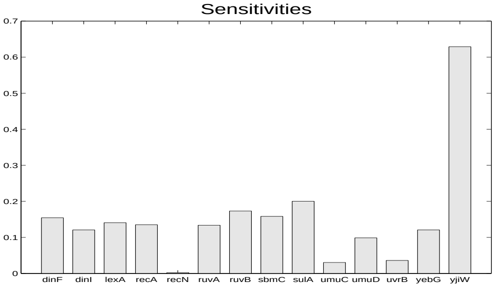


Results in E.coli data: Kinetic parameters

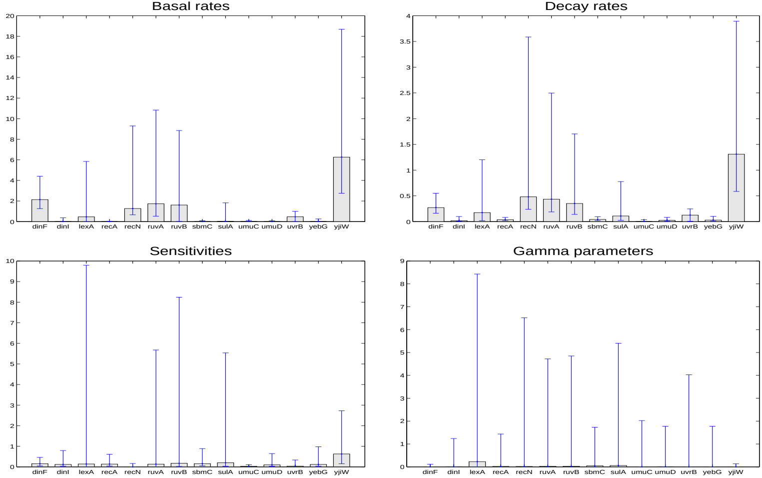


Results in E.coli data: Genes with low sensitivity value





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



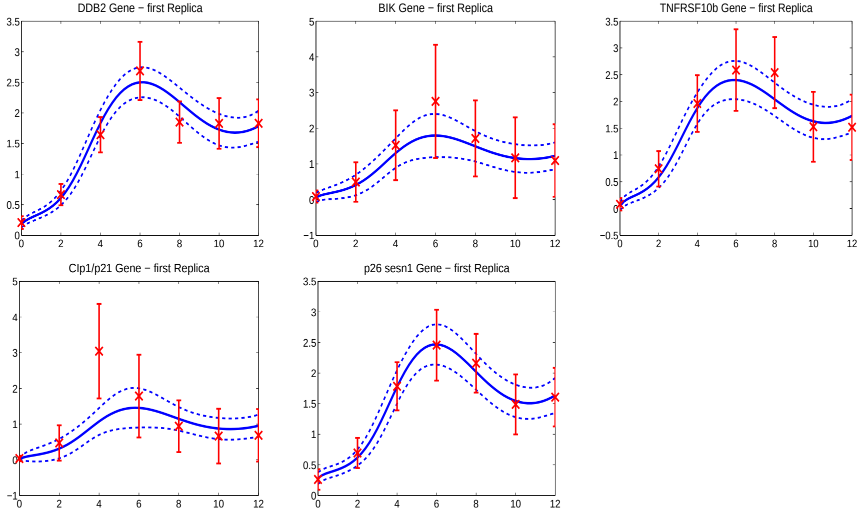
p53 System Again

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

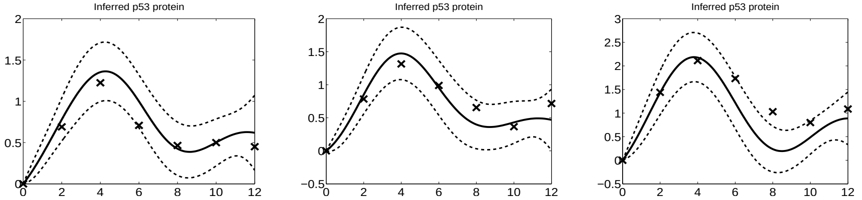
$$\frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = b_j + s_j \frac{\exp(f(t))}{\exp(f(t)) + \gamma_j} - d_j x_j(t)$$

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

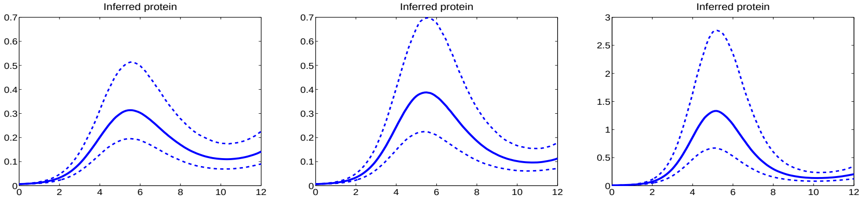
Data used by Barenco et al. (2006): Predicted gene expressions for the 1st replica



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

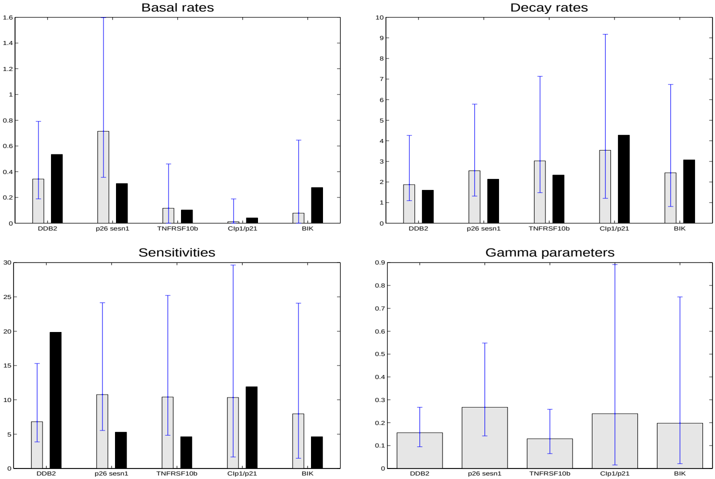


Linear model (Barenco et al. predictions are shown as crosses)



Nonlinear (Michaelis-Menten kinetic equation)

p53 Data Kinetic parameters



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