Latent Force Models

Neil D. Lawrence
work with Magnus Rattray, **Mauricio Alvarez**, Pei Gao, Antti Honkela, David Luengo, Guido Sanguinetti, Michalis Titsias, Jennifer Withers

University of Sheffield
Talk at Functional Phylogenies Workshop, Oxford, U.K.

28th September 2010

Outline

Outline

Styles of Machine Learning

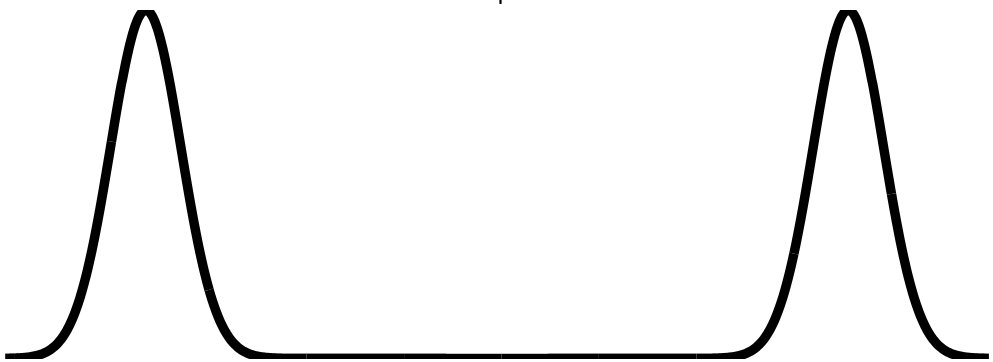
Background: interpolation is easy, extrapolation is hard

- Urs Hölzle keynote talk at NIPS 2005.
  - Emphasis on massive data sets.
  - Let the data do the work—more data, less extrapolation.
- Alternative paradigm:
  - Very scarce data: computational biology, human motion.
  - How to generalize from scarce data?
  - Need to include more assumptions about the data (e.g. invariances).

General Approach

Broadly Speaking: Two approaches to modeling

data modeling | *mechanistic modeling*



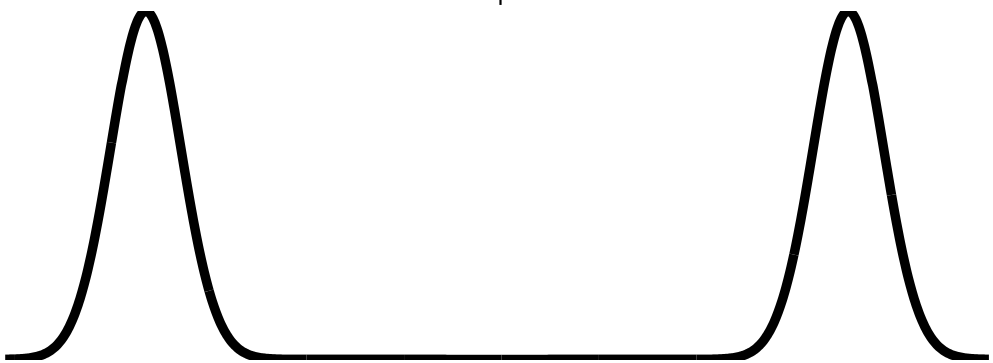
General Approach

Broadly Speaking: Two approaches to modeling

data modeling

let the data "speak"

mechanistic modeling



General Approach

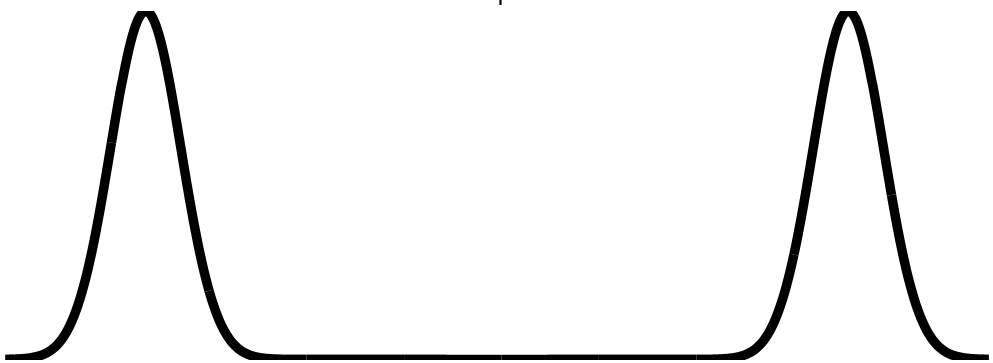
Broadly Speaking: Two approaches to modeling

data modeling

let the data "speak"

mechanistic modeling

impose physical laws



General Approach

Broadly Speaking: Two approaches to modeling

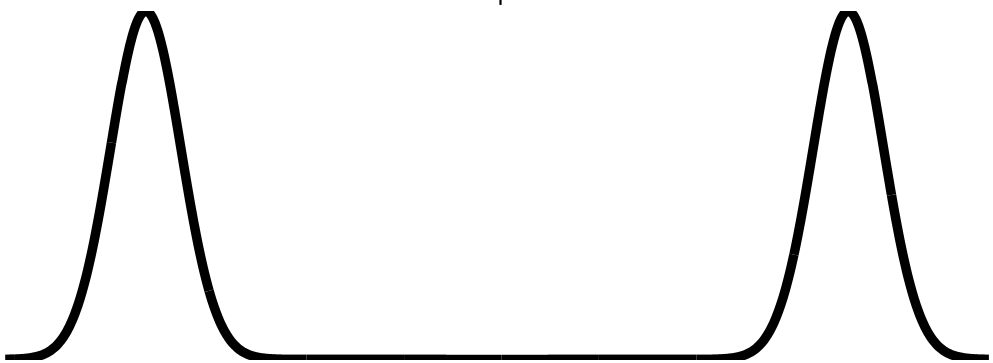
data modeling

let the data "speak"

data driven

mechanistic modeling

impose physical laws



General Approach

Broadly Speaking: Two approaches to modeling

data modeling

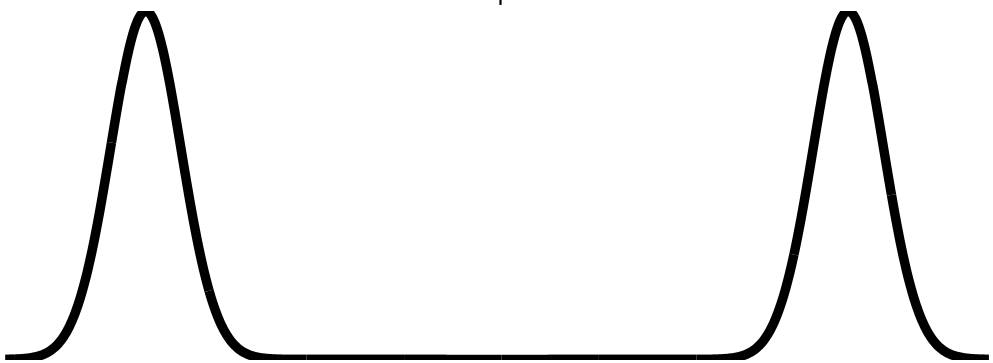
let the data "speak"

data driven

mechanistic modeling

impose physical laws

knowledge driven



General Approach

Broadly Speaking: Two approaches to modeling

data modeling

let the data "speak"

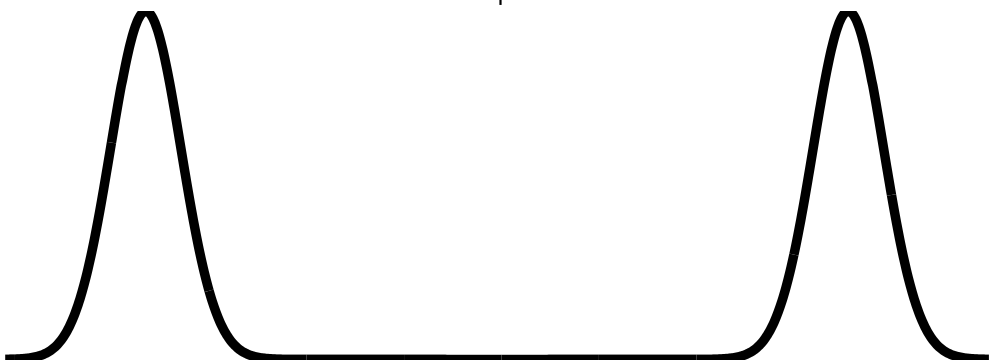
data driven

adaptive models

mechanistic modeling

impose physical laws

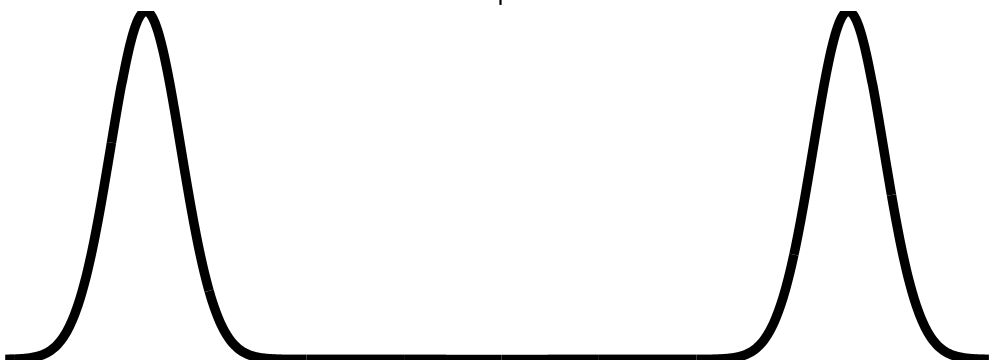
knowledge driven



General Approach

Broadly Speaking: Two approaches to modeling

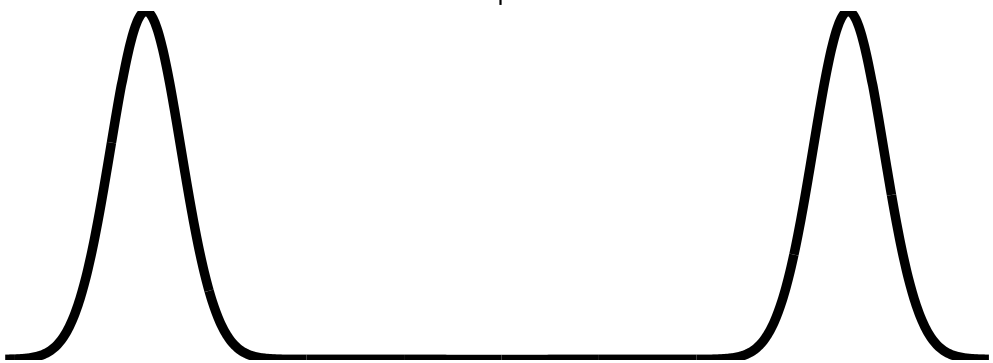
data modeling	*mechanistic modeling*
let the data "speak"	impose physical laws
data driven	knowledge driven
adaptive models	differential equations



General Approach

Broadly Speaking: Two approaches to modeling

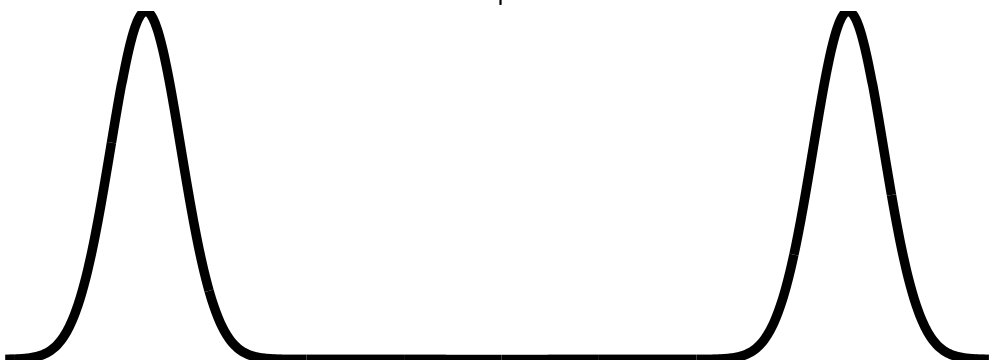
data modeling	*mechanistic modeling*
let the data "speak"	impose physical laws
data driven	knowledge driven
adaptive models	differential equations
digit recognition	



General Approach

Broadly Speaking: Two approaches to modeling

data modeling	*mechanistic modeling*
let the data "speak"	impose physical laws
data driven	knowledge driven
adaptive models	differential equations
digit recognition	climate, weather models

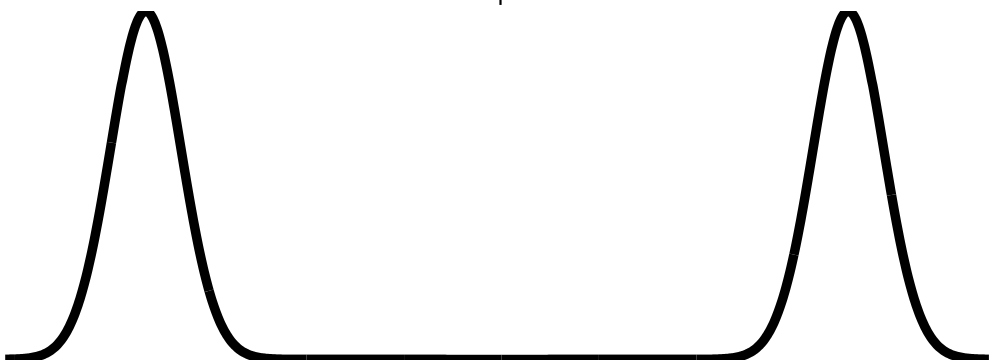


General Approach

Broadly Speaking: Two approaches to modeling

data modeling	*mechanistic modeling*
let the data "speak"	impose physical laws
data driven	knowledge driven
adaptive models	differential equations
digit recognition	climate, weather models

Weakly Mechanistic



General Approach

Broadly Speaking: Two approaches to modeling

data modeling

let the data "speak"

data driven

adaptive models

digit recognition

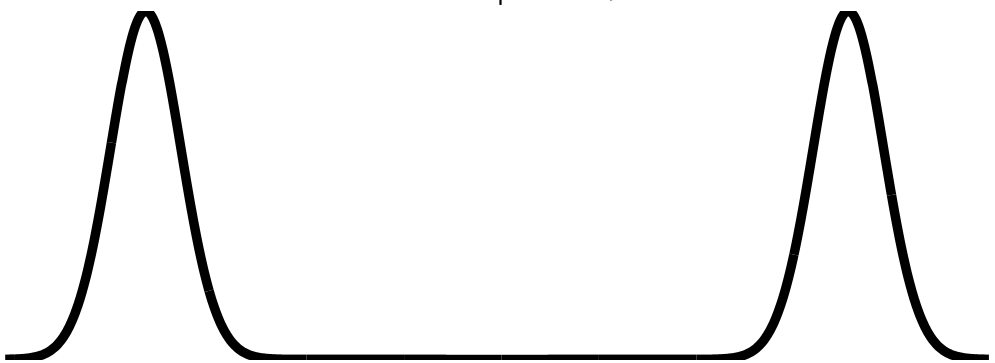
mechanistic modeling

impose physical laws

knowledge driven

differential equations

climate, weather models



Weakly Mechanistic vs Strongly Mechanistic

- Underlying data modeling techniques there are *weakly mechanistic* principles (e.g. smoothness).
- In physics the models are typically *strongly mechanistic*.
- In principle we expect a range of models which vary in the strength of their mechanistic assumptions.
- This work is one part of that spectrum: add further mechanistic ideas to weakly mechanistic models.

What is Machine Learning?

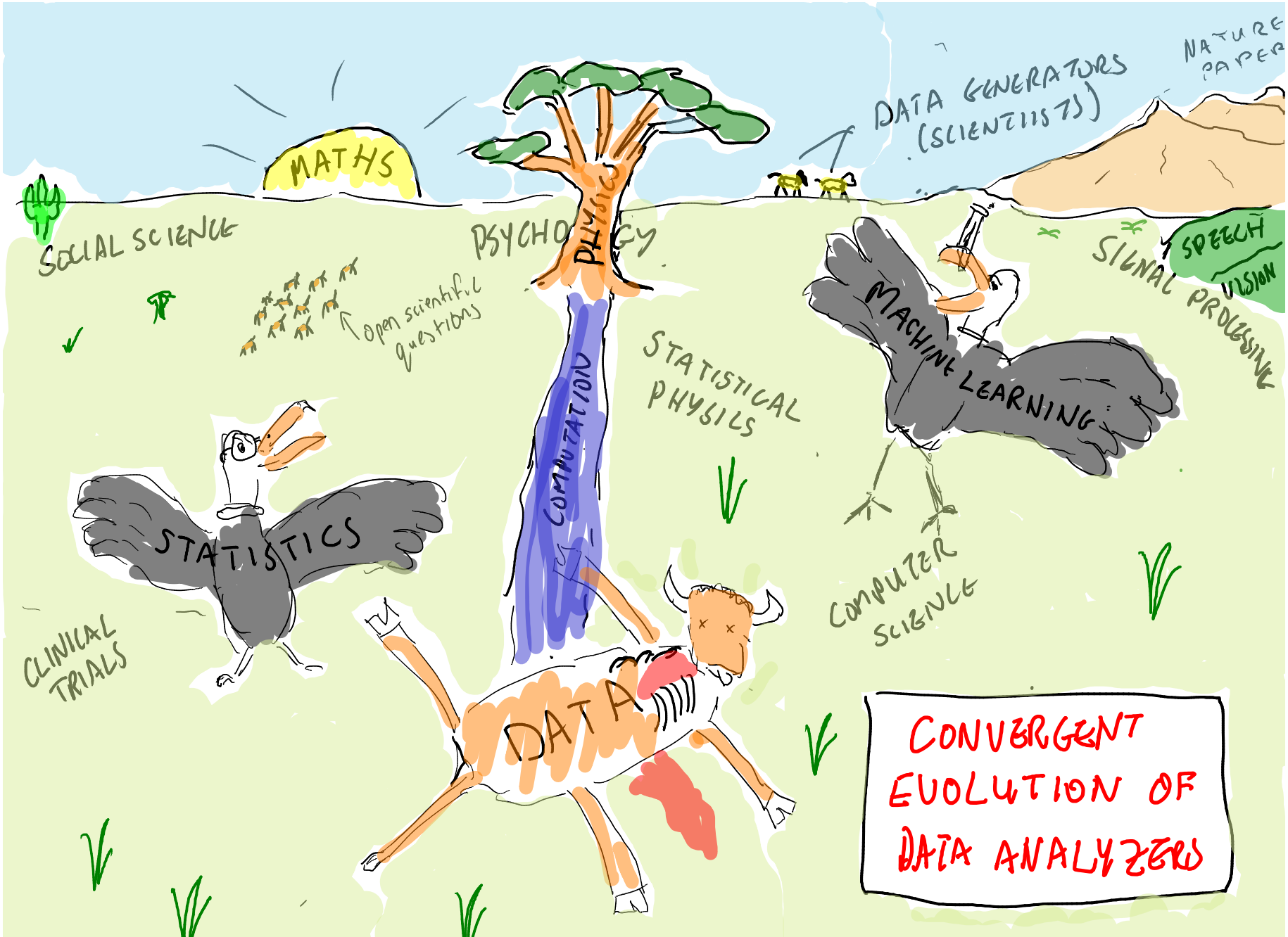
- Arises from the Artifical Intelligence community.
  - Objective: endow computers with the ability to learn like humans.
  - Actuality: fitting models to data to make predictions. In particular classification.
- Modern machine learning has its routes in "Connectionism". Originally neural networks but now dominated by:
  - Kernel methods (support vector machines)
  - Probabilistic methods (graphical models, Bayesian approaches, Bayesian non-parametrics)
- No probabilist/statistician distinction.

Vultures

New World vultures are not closely genetically related to the superficially similar family of Old World vultures; similarities between the two groups are due to convergent evolution. Just how closely related they are is a matter of debate ...

- New World Vultures detect prey through smelling ethyl mercaptan
- Old World Vultures detect prey through sight.
- Convergent Evolution means the two birds are morphologically similar.

The Academic Serengeti



Outline

Dimensionality Reduction

- Linear relationship between the data, $\mathbf{X} \in \Re^{n \times p}$, and a reduced dimensional representation, $\mathbf{F} \in \Re^{n \times q}$, where $q \ll p$.

$$\mathbf{X} = \mathbf{F}\mathbf{W} + \boldsymbol{\epsilon},$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$$

- Integrate out $\mathbf{F}$, optimize with respect to $\mathbf{W}$.
- For Gaussian prior, $\mathbf{F} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  - and $\boldsymbol{\Sigma} = \sigma^2 \mathbf{I}$ we have probabilistic PCA (Tipping and Bishop, 1999; Roweis, 1998).
  - and $\boldsymbol{\Sigma}$ constrained to be diagonal, we have factor analysis.

Dimensionality Reduction: Temporal Data

- Deal with temporal data with a temporal latent prior.
- Independent Gauss-Markov priors over each $f_i(t)$ leads to : Rauch-Tung-Striebel (RTS) smoother (Kalman filter).
- More generally consider a Gaussian process (GP) prior,

$$p(\mathbf{F}|\mathbf{t}) = \prod_{i=1}^{q} \mathcal{N}\left(\mathbf{f}_{:,i}|\mathbf{0}, \mathbf{K}_{f_{:,i},f_{:,i}}\right).$$

- Given the covariance functions for $\{f_i(t)\}$ we have an implied covariance function across all $\{x_i(t)\}$—(ML: semi-parametric latent factor model (Teh et al., 2005), Geostatistics: linear model of coregionalization).
- Rauch-Tung-Striebel smoother has been preferred
  - linear computational complexity in n.
  - Advances in sparse approximations have made the general GP framework practical. (Titsias, 2009; Snelson and Ghahramani, 2006; Quiñonero Candela and Rasmussen, 2005).

Gaussian Distribution

Zero mean Gaussian distribution

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

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

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

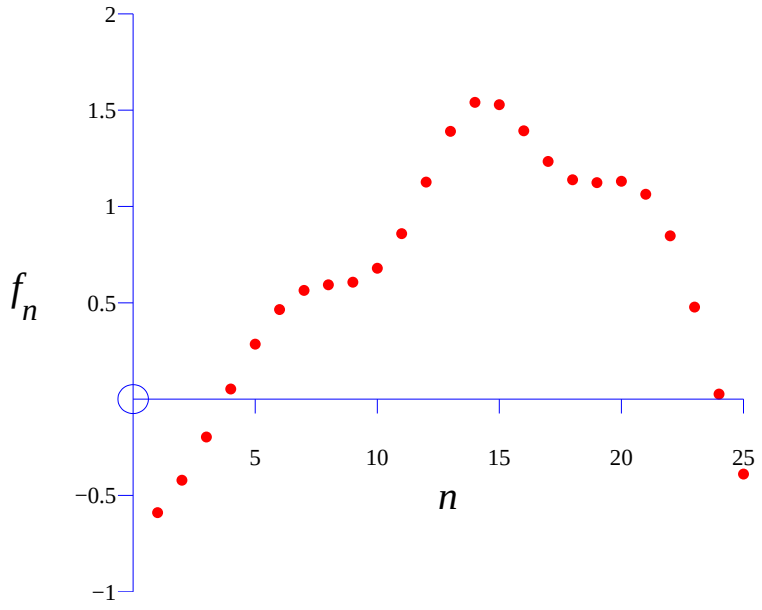
$$\mathcal{N}\left(\mathbf{f}|\mathbf{0},\mathbf{K}\right)=\frac{1}{\left(2\pi\right)^{\frac{n}{2}}\left|\mathbf{K}\right|^{\frac{1}{2}}}\exp\left(-\frac{\mathbf{f}^{\top}\mathbf{K}^{-1}\mathbf{f}}{2}\right).$$

Sampling a Function

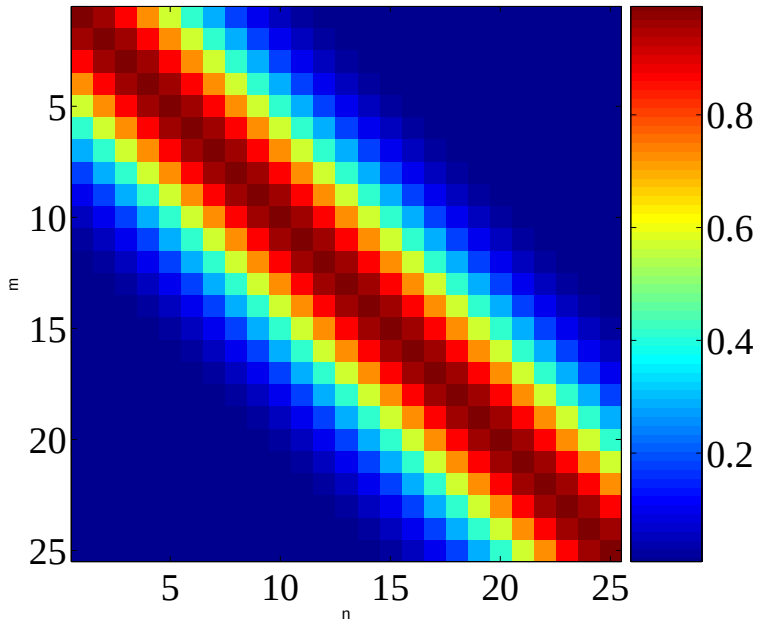
Multi-variate Gaussians

- We will consider a Gaussian with a particular structure of covariance matrix.
- Generate a single sample from this 25 dimensional Gaussian distribution, $\mathbf{f} = [f_1, f_2 \dots f_{25}]$.
- 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])
```

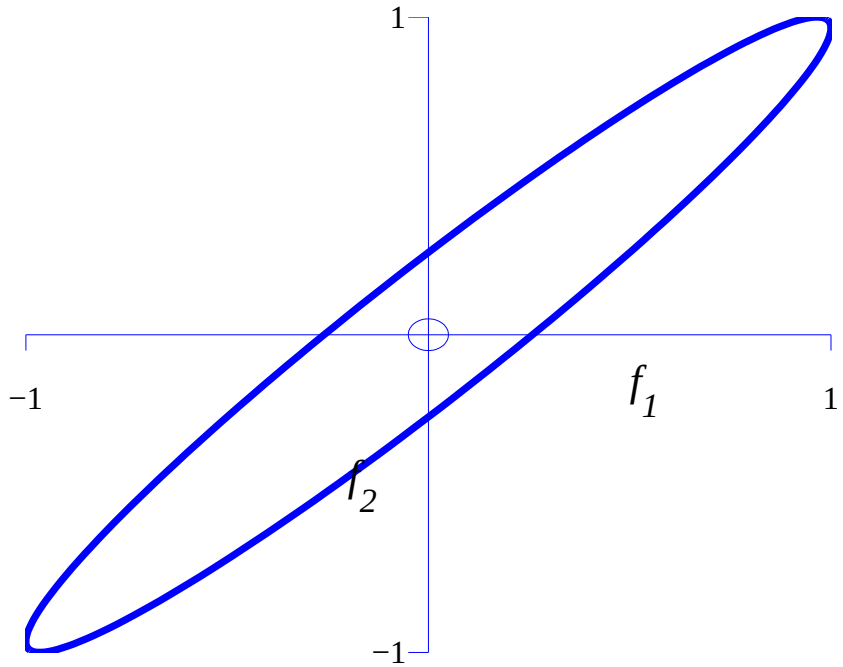


Figure: Covariance for $\begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$ is $\mathbf{K}_{12} = \begin{bmatrix} 1 & 0.966 \\ 0.966 & 1 \end{bmatrix}$.

Prediction of f_2 from f_1

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

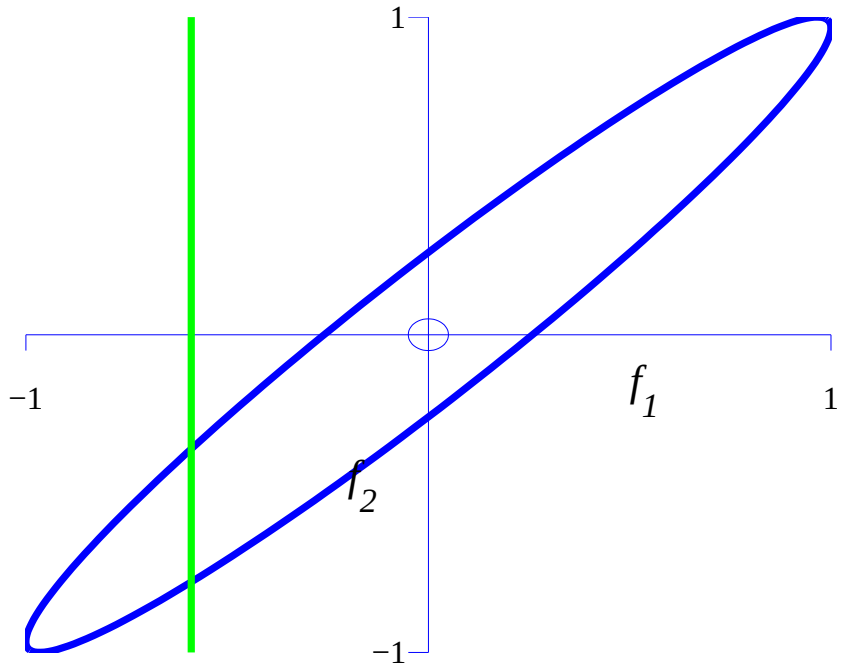


Figure: Covariance for $\begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$ is $\mathbf{K}_{12} = \begin{bmatrix} 1 & 0.966 \\ 0.966 & 1 \end{bmatrix}$.

Prediction of f_2 from f_1

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

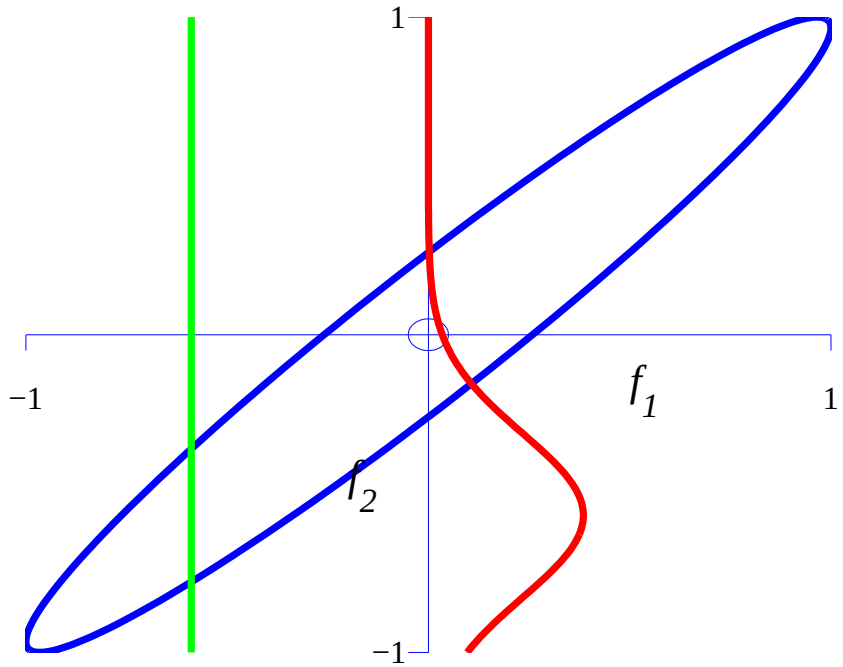


Figure: Covariance for $\begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$ is $\mathbf{K}_{12} = \begin{bmatrix} 1 & 0.966 \\ 0.966 & 1 \end{bmatrix}$.

Prediction of f_5 from f_1

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

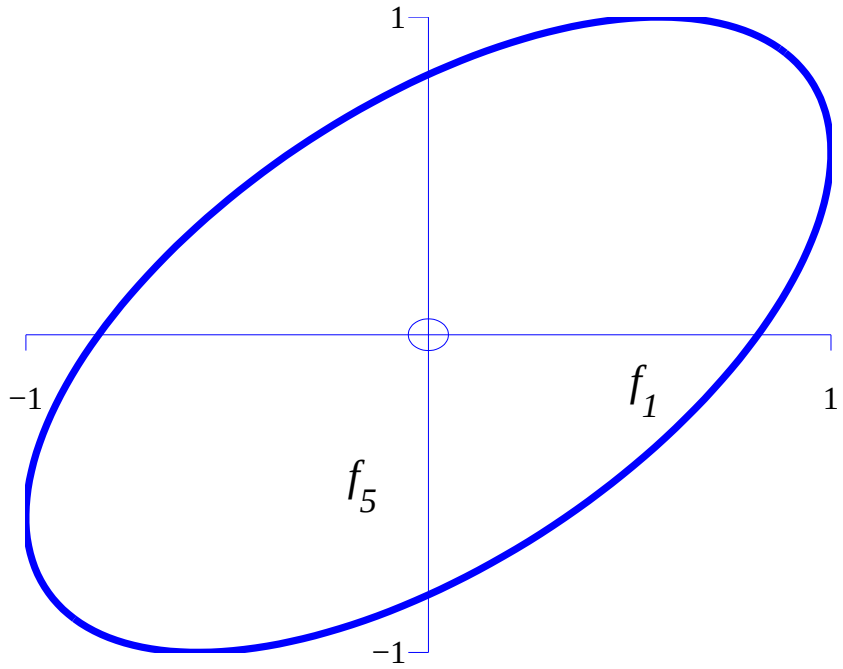


Figure: Covariance for $\begin{bmatrix} f_1 \\ f_5 \end{bmatrix}$ is $\mathbf{K}_{15} = \begin{bmatrix} 1 & 0.574 \\ 0.574 & 1 \end{bmatrix}$.

Prediction of f_5 from f_1

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

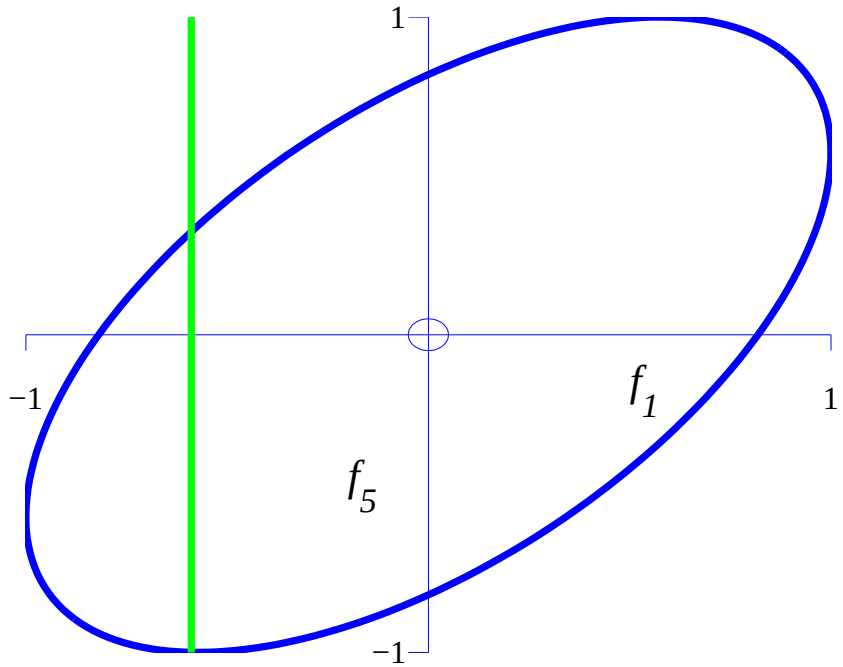


Figure: Covariance for $\begin{bmatrix} f_1 \\ f_5 \end{bmatrix}$ is $\mathbf{K}_{15} = \begin{bmatrix} 1 & 0.574 \\ 0.574 & 1 \end{bmatrix}$.

Prediction of f_5 from f_1

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

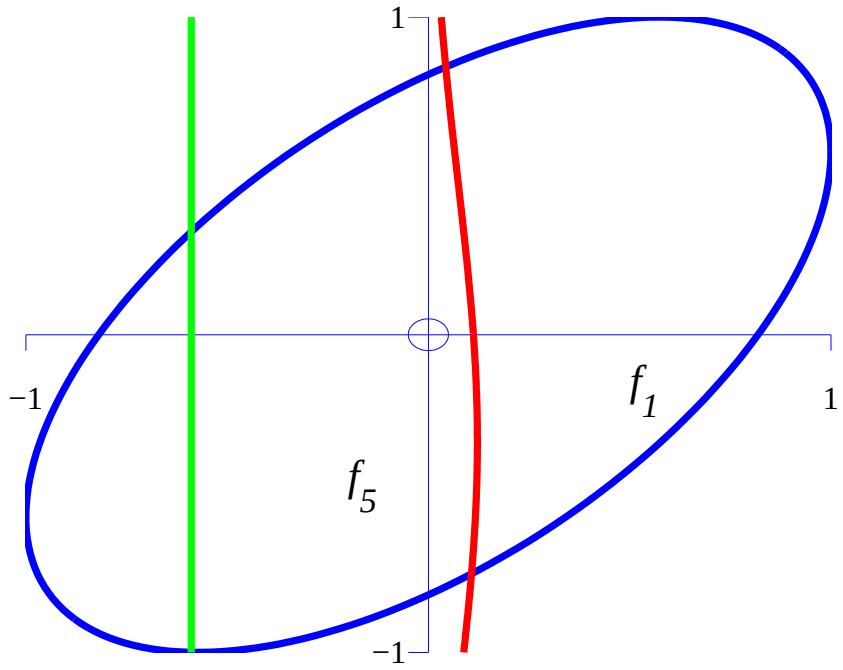


Figure: Covariance for $\begin{bmatrix} f_1 \\ f_5 \end{bmatrix}$ is $\mathbf{K}_{15} = \begin{bmatrix} 1 & 0.574 \\ 0.574 & 1 \end{bmatrix}$.

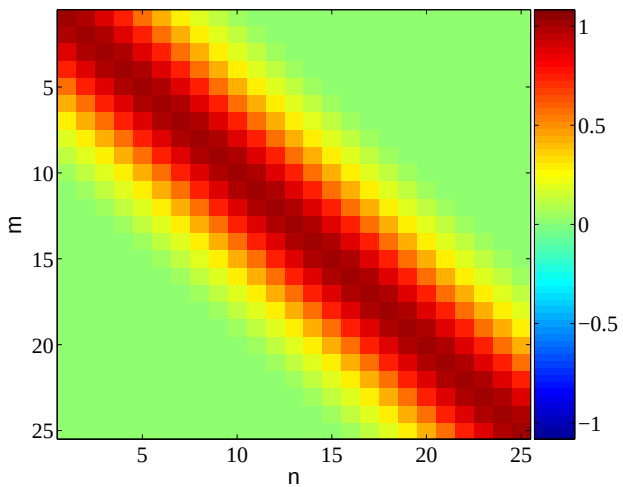
Covariance Functions

Where did this covariance matrix come from?

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

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

- 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`

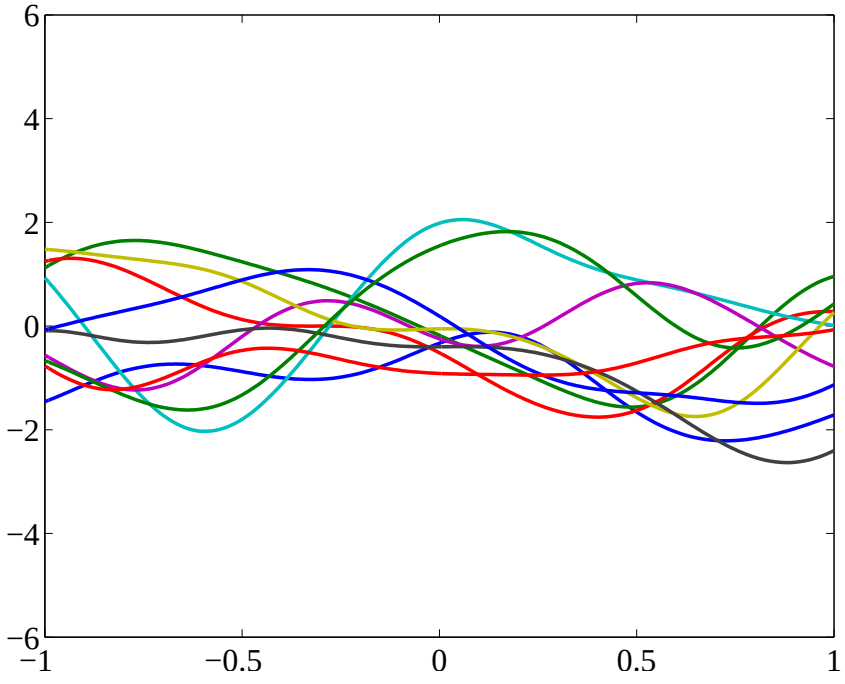


Figure: Exponentiated quadratic kernel with $\ell = 0.3$, $\alpha = 1$

Covariance Samples

`demCovFuncSample`

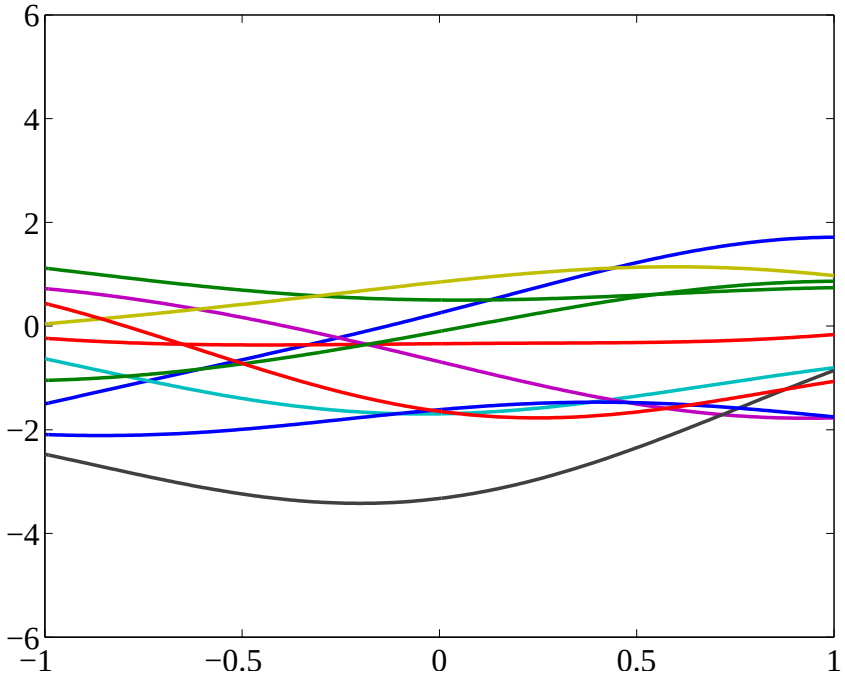


Figure: Exponentiated quadratic kernel with $\ell = 1$, $\alpha = 1$

Covariance Samples

`demCovFuncSample`

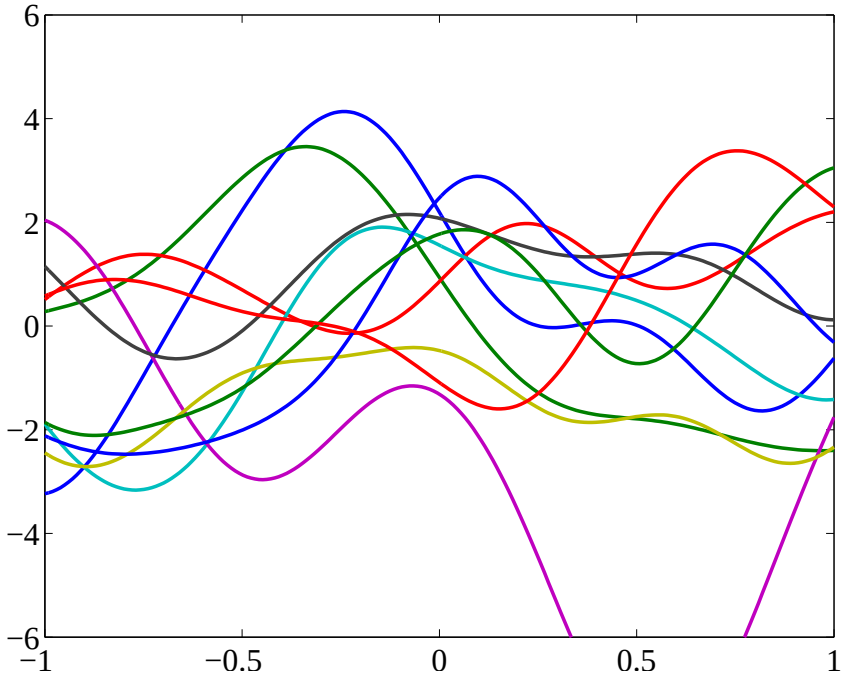


Figure: Exponentiated quadratic kernel with $\ell = 0.3$, $\alpha = 4$

Covariance Samples

`demCovFuncSample`

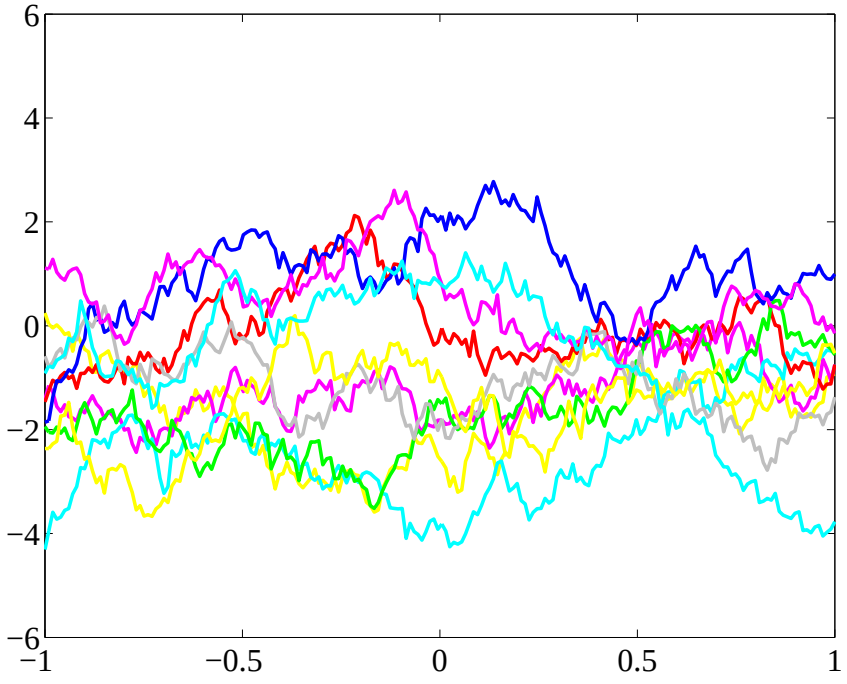


Figure: Ornstein-Uhlenbeck (stationary Gauss-Markov) covariance function $\ell = 1$, $\alpha = 4$

Outline

Mechanical Analogy

Back to Mechanistic Models!

- These models rely on the latent variables to provide the dynamic information.
- We now introduce a further dynamical system with a *mechanistic* inspiration.
- Physical Interpretation:
  - the latent functions, $f_i(t)$ are q forces.
  - We observe the displacement of p springs to the forces.,
  - Interpret system as the force balance equation, $\mathbf{XD} = \mathbf{FS} + \epsilon$.
  - Forces act, e.g. through levers — a matrix of sensitivities, $\mathbf{S} \in \Re^{q \times p}$.
  - Diagonal matrix of spring constants, $\mathbf{D} \in \Re^{p \times p}$.
  - Original System: $\mathbf{W} = \mathbf{S}\mathbf{D}^{-1}$.

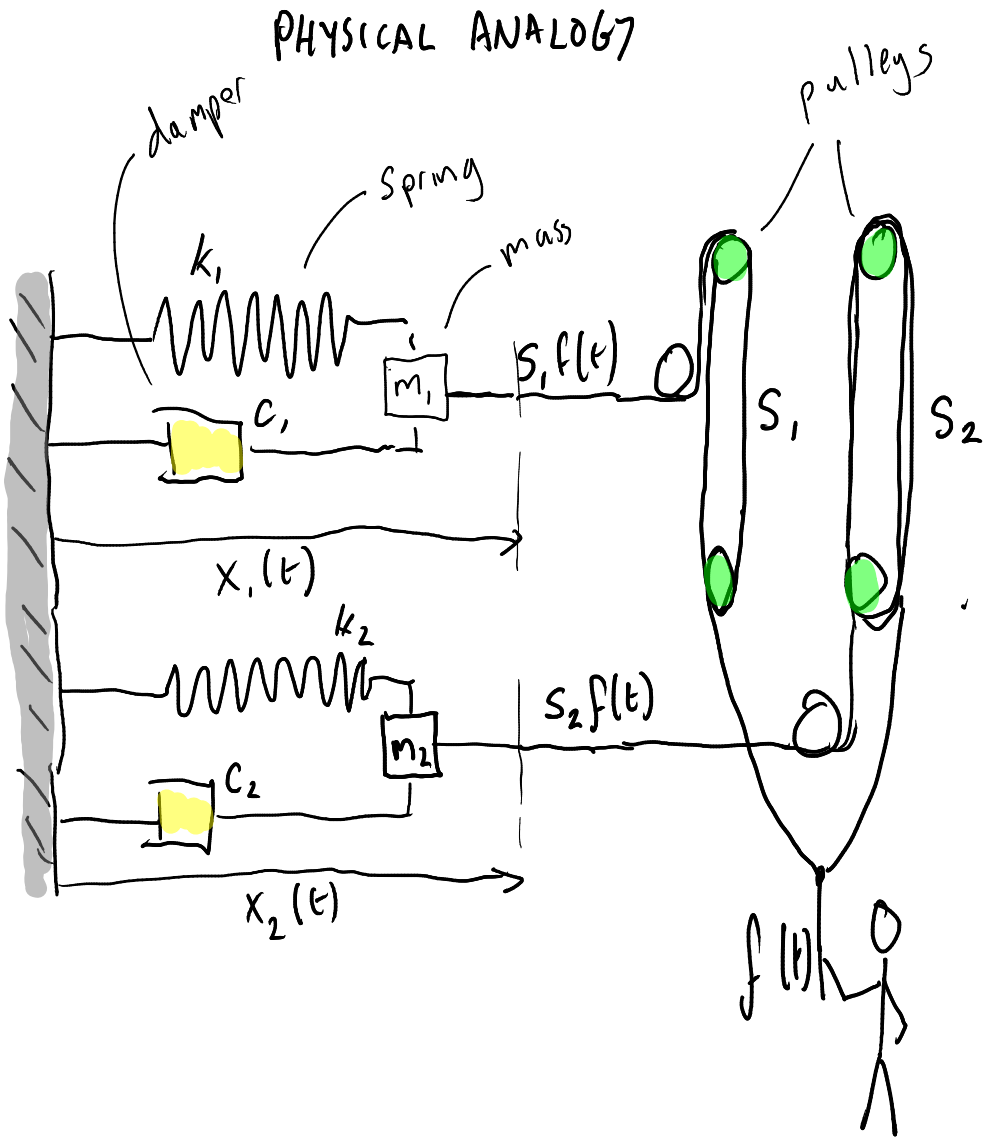
Extend Model

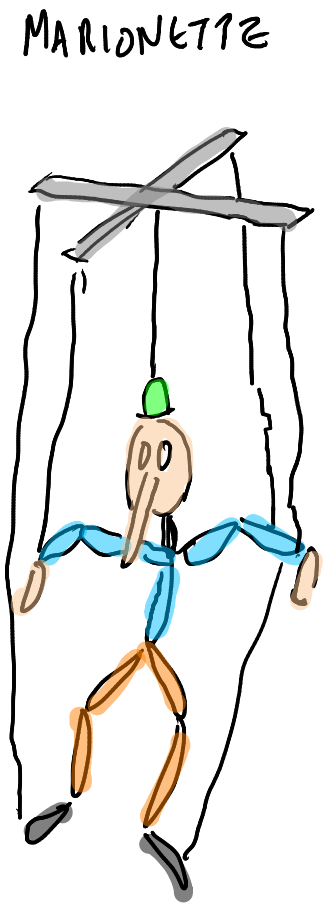
- Add a damper and give the system mass.

$$\mathbf{FS} = \ddot{\mathbf{X}}\mathbf{M} + \dot{\mathbf{X}}\mathbf{C} + \mathbf{X}\mathbf{D} + \epsilon.$$

- Now have a second order mechanical system.
- It will exhibit inertia and resonance.
- There are many systems that can also be represented by differential equations.
  - When being forced by latent function(s), $\{f_i(t)\}_{i=1}^{q}$, we call this a *latent force model*.

Physical Analogy





Gaussian Process priors and Latent Force Models

Driven Harmonic Oscillator

- For Gaussian process we can compute the covariance matrices for the output displacements.
- For one displacement the model is

$$m_k \ddot{x}_k(t) + c_k \dot{x}_k(t) + d_k x_k(t) = b_k + \sum_{i=0}^{q} s_{ik} f_i(t), \qquad (1)$$

where, m_k is the kth diagonal element from $\mathbf{M}$ and similarly for c_k and d_k. s_{ik} is the i, kth element of $\mathbf{S}$.

- Model the latent forces as q independent, GPs with exponentiated quadratic covariances

$$k_{f_i f_l}(t, t') = \exp\left(-\frac{(t - t')^2}{2\ell_i^2}\right) \delta_{il}.$$

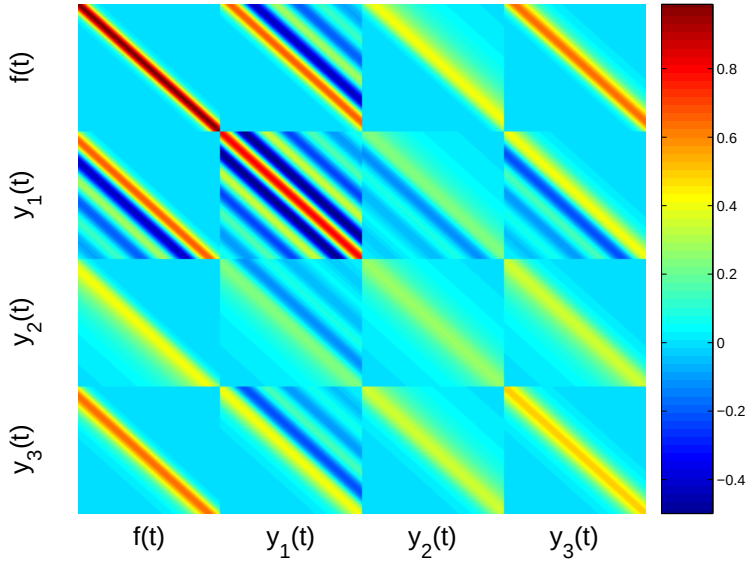
Covariance for ODE Model

- Exponentiated Quadratic Covariance function for $f(t)$

$$x_j(t) = \frac{1}{m_j \omega_j} \sum_{i=1}^{q} s_{ji} \exp(-\alpha_j t) \int_0^t f_i(\tau) \exp(\alpha_j \tau) \sin(\omega_j (t-\tau)) \mathrm{d}\tau$$

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

ζ_1	ζ_2	ζ_3
0.125	2	1



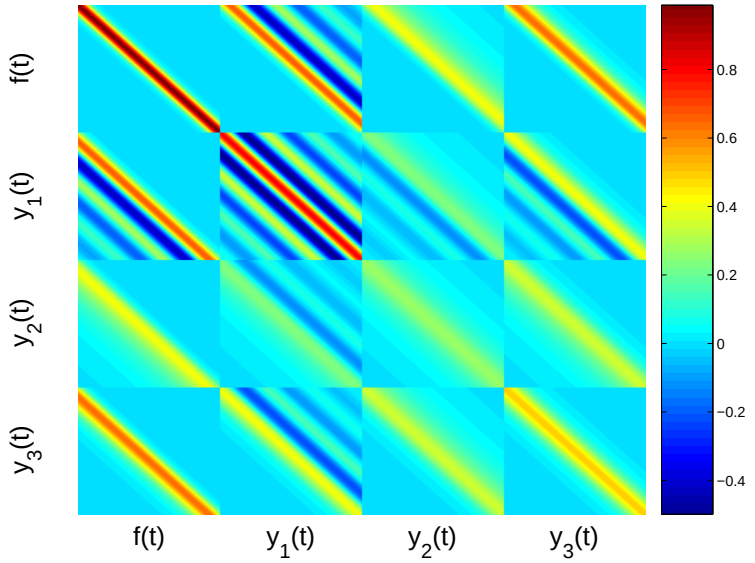
Covariance for ODE Model

- Analogy

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

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

ζ_1	ζ_2	ζ_3
0.125	2	1



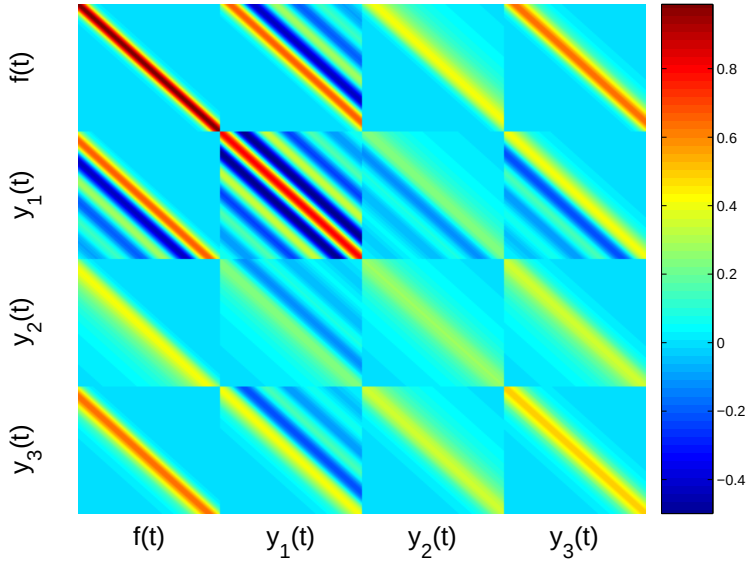
Covariance for ODE Model

- Exponentiated Quadratic Covariance function for $f(t)$

$$x_j(t) = \frac{1}{m_j \omega_j} \sum_{i=1}^{q} s_{ji} \exp(-\alpha_j t) \int_0^t f_i(\tau) \exp(\alpha_j \tau) \sin(\omega_j (t-\tau)) \mathrm{d}\tau$$

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

ζ_1	ζ_2	ζ_3
0.125	2	1



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

- `lfmSample`

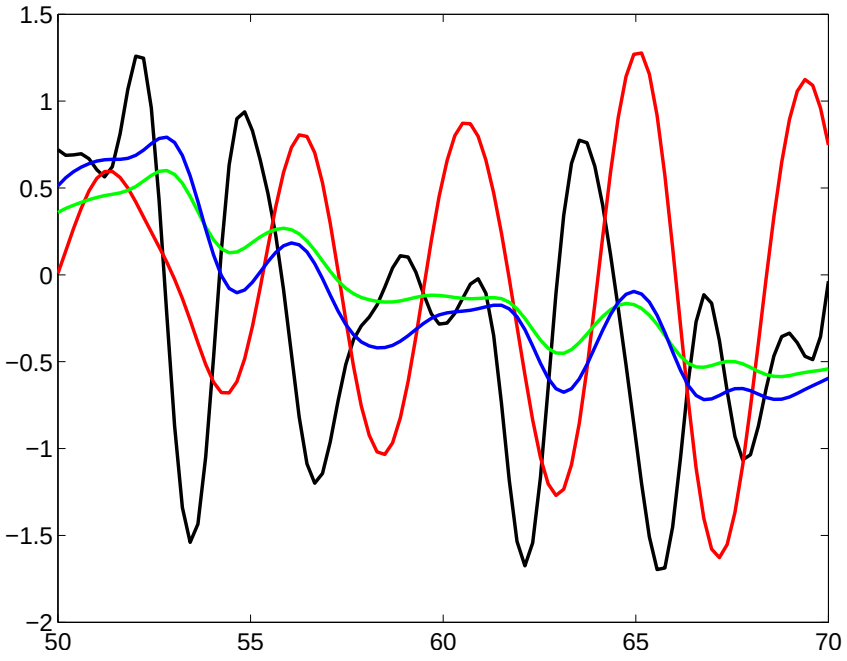


Figure: Joint samples from the ODE covariance, *black*: $f(t)$, *red*: $x_1(t)$ (underdamped), *green*: $x_2(t)$ (overdamped), and *blue*: $x_3(t)$ (critically damped).

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

- `lfmSample`

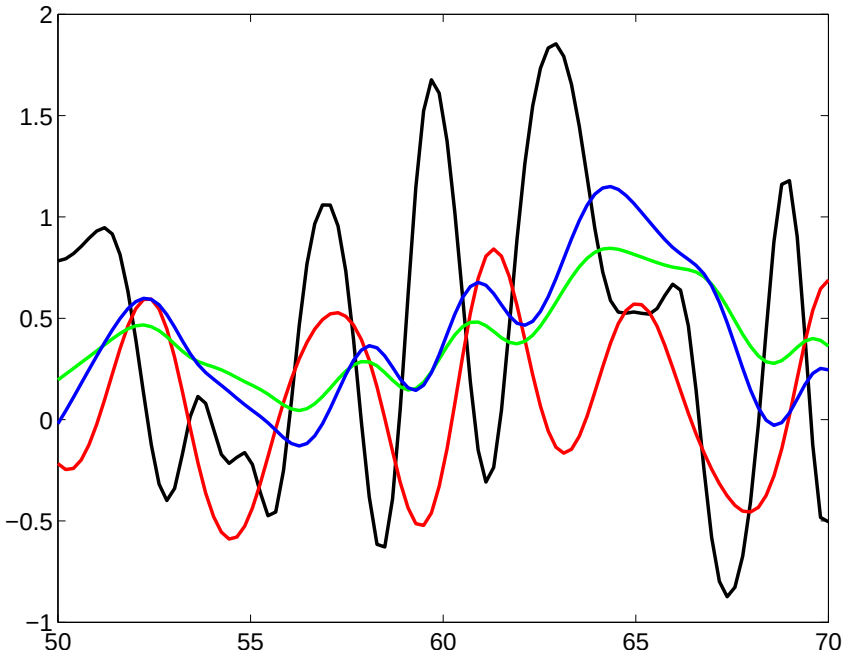


Figure: Joint samples from the ODE covariance, *black*: $f(t)$, *red*: $x_1(t)$ (underdamped), *green*: $x_2(t)$ (overdamped), and *blue*: $x_3(t)$ (critically damped).

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

- `lfmSample`

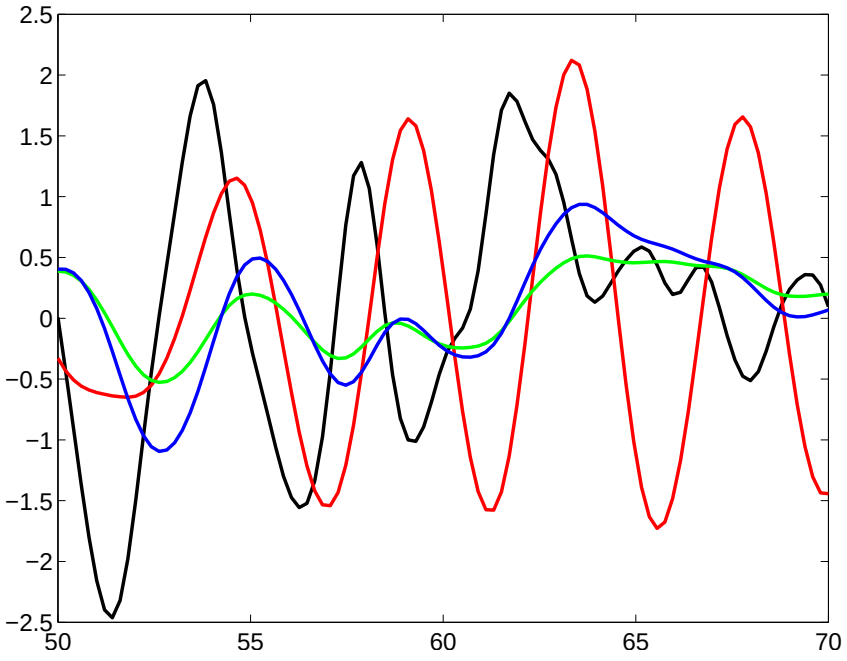


Figure: Joint samples from the ODE covariance, *black*: $f(t)$, *red*: $x_1(t)$ (underdamped), *green*: $x_2(t)$ (overdamped), and *blue*: $x_3(t)$ (critically damped).

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

- `lfmSample`

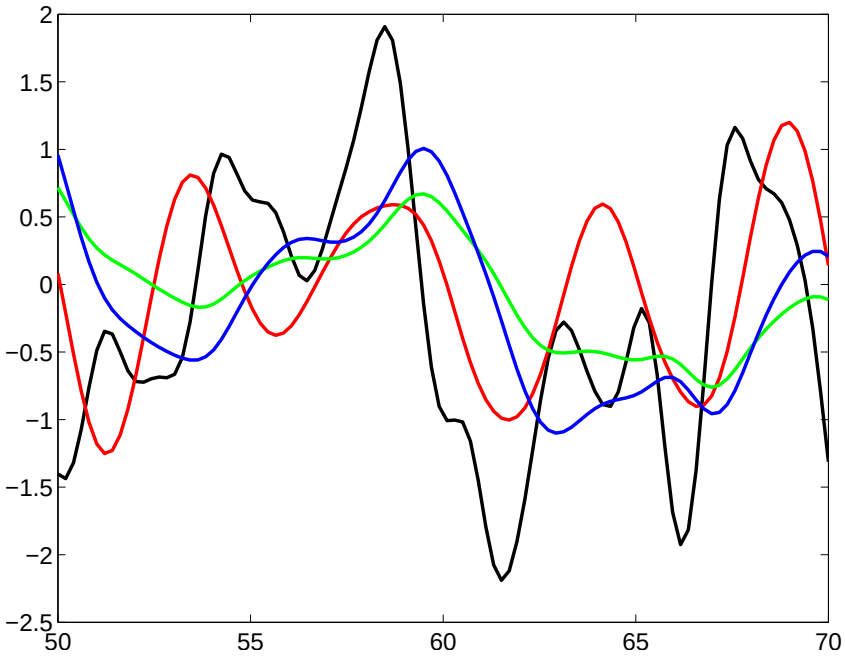


Figure: Joint samples from the ODE covariance, *black*: $f(t)$, *red*: $x_1(t)$ (underdamped), *green*: $x_2(t)$ (overdamped), and *blue*: $x_3(t)$ (critically damped).

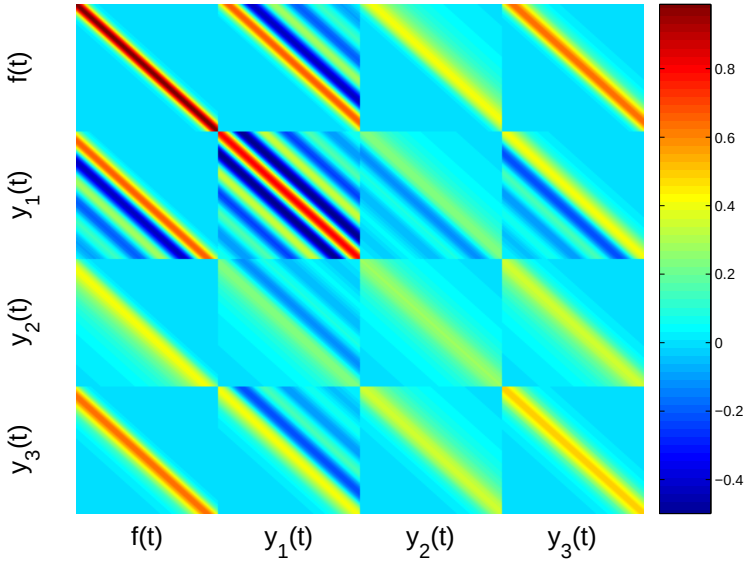
Covariance for ODE

- Exponentiated Quadratic Covariance function for $f(t)$

$$x_j(t) = \frac{1}{m_j \omega_j} \sum_{i=1}^{q} s_{ji} \exp(-\alpha_j t) \int_0^t f_i(\tau) \exp(\alpha_j \tau) \sin(\omega_j (t-\tau)) \mathrm{d}\tau$$

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

ζ_1	ζ_2	ζ_3
0.125	2	1



Example: Motion Capture

Mauricio Alvarez and David Luengo (Álvarez et al., 2009)

- Motion capture data: used for animating human motion.
- Multivariate time series of angles representing joint positions.
- Objective: generalize from training data to realistic motions.
- Use 2nd Order Latent Force Model with mass/spring/damper (resistor inductor capacitor) at each joint.

Example: Motion Capture

Mauricio Alvarez and David Luengo (Álvarez et al., 2009)

- Motion capture data: used for animating human motion.
- Multivariate time series of angles representing joint positions.
- Objective: generalize from training data to realistic motions.
- Use 2nd Order Latent Force Model with mass/spring/damper (resistor inductor capacitor) at each joint.

Example: Motion Capture

Mauricio Alvarez and David Luengo (Álvarez et al., 2009)

- Motion capture data: used for animating human motion.
- Multivariate time series of angles representing joint positions.
- Objective: generalize from training data to realistic motions.
- Use 2nd Order Latent Force Model with mass/spring/damper (resistor inductor capacitor) at each joint.

Example: Motion Capture

Mauricio Alvarez and David Luengo (Álvarez et al., 2009)

- Motion capture data: used for animating human motion.
- Multivariate time series of angles representing joint positions.
- Objective: generalize from training data to realistic motions.
- Use 2nd Order Latent Force Model with mass/spring/damper (resistor inductor capacitor) at each joint.

Prediction of Test Motion

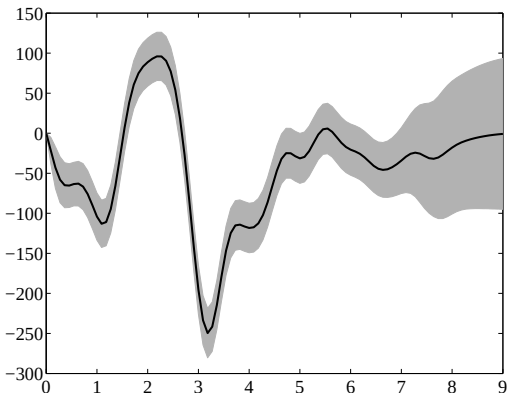
- Model left arm only.
- 3 balancing motions (18, 19, 20) from subject 49.
- 18 and 19 are similar, 20 contains more dramatic movements.
- Train on 18 and 19 and testing on 20
- Data was down-sampled by 32 (from 120 fps).
- Reconstruct motion of left arm for 20 given other movements.
- Compare with GP that predicts left arm angles given other body angles.

Mocap Results

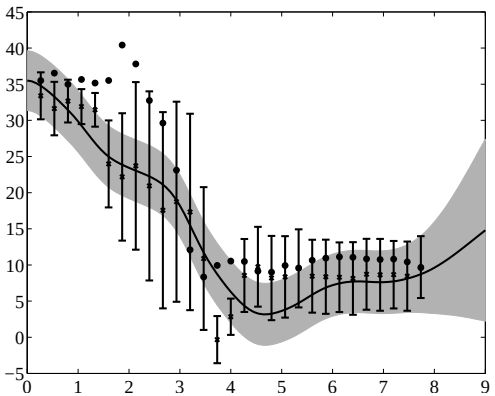
Table: Root mean squared (RMS) angle error for prediction of the left arm's configuration in the motion capture data. Prediction with the latent force model outperforms the prediction with regression for all apart from the radius's angle.

Angle	Latent Force Error	Regression Error
Radius	4.11	4.02
Wrist	6.55	6.65
Hand X rotation	1.82	3.21
Hand Z rotation	2.76	6.14
Thumb X rotation	1.77	3.10
Thumb Z rotation	2.73	6.09

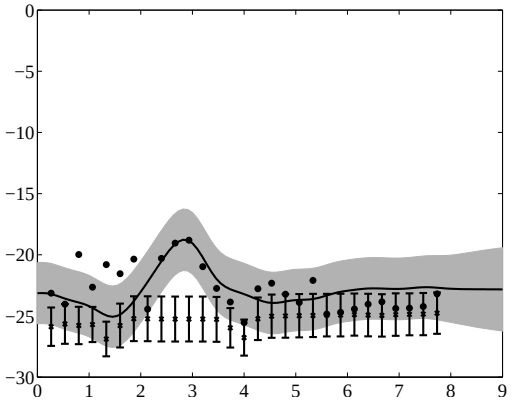
Mocap Results II



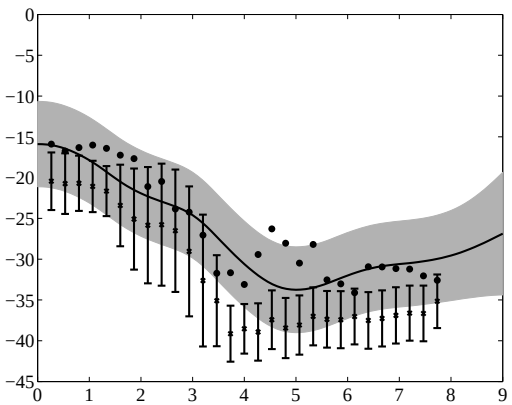
(a) Inferred Latent Force



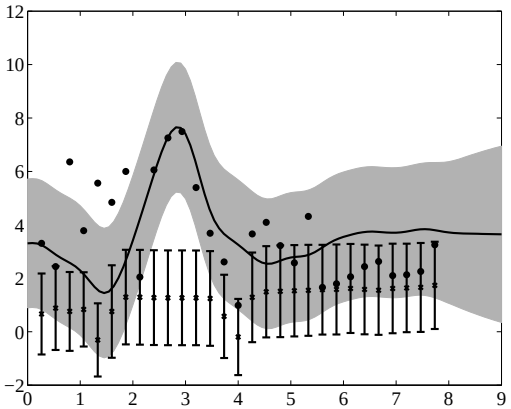
(b) Wrist



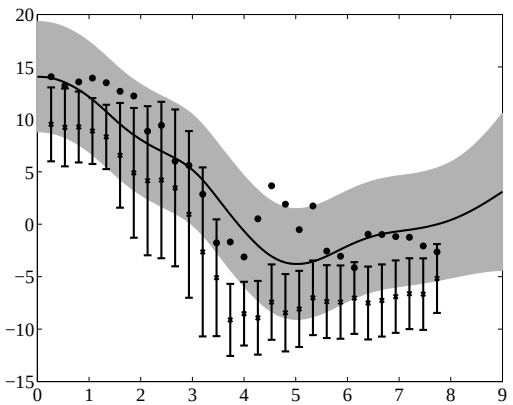
(c) Hand X Rotation



(d) Hand Z Rotation



(e) Thumb X Rotation



(f) Thumb Z Rotation

Figure: Predictions from LFM (solid line, grey error bars) and direct regression (crosses with stick error bars).

Example: Transcriptional Regulation

- First Order Differential Equation

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

- Can be used as a model of gene transcription: Barenco et al., 2006; Gao et al., 2008.
- $x_j(t)$ – concentration of gene j's mRNA
- $f(t)$ – concentration of active transcription factor
- Model parameters: baseline b_j, sensitivity s_j and decay d_j
- Application: identifying co-regulated genes (targets)
- Problem: how do we fit the model when $f(t)$ is not observed?

Example: Transcriptional Regulation

- First Order Differential Equation

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

- Can be used as a model of gene transcription: Barenco et al., 2006; Gao et al., 2008.
- $x_j(t)$ – concentration of gene j's mRNA
- $f(t)$ – concentration of active transcription factor
- Model parameters: baseline b_j, sensitivity s_j and decay d_j
- Application: identifying co-regulated genes (targets)
- Problem: how do we fit the model when $f(t)$ is not observed?

Example: Transcriptional Regulation

- First Order Differential Equation

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

- Can be used as a model of gene transcription: Barenco et al., 2006; Gao et al., 2008.
- $x_j(t)$ – concentration of gene j's mRNA
- $f(t)$ – concentration of active transcription factor
- Model parameters: baseline b_j, sensitivity s_j and decay d_j
- Application: identifying co-regulated genes (targets)
- Problem: how do we fit the model when $f(t)$ is not observed?

Example: Transcriptional Regulation

- First Order Differential Equation

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

- Can be used as a model of gene transcription: Barenco et al., 2006; Gao et al., 2008.
- $x_j(t)$ – concentration of gene j's mRNA
- $f(t)$ – concentration of active transcription factor
- Model parameters: baseline b_j, sensitivity s_j and decay d_j
- Application: identifying co-regulated genes (targets)
- Problem: how do we fit the model when $f(t)$ is not observed?

Example: Transcriptional Regulation

- First Order Differential Equation

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

- Can be used as a model of gene transcription: Barenco et al., 2006; Gao et al., 2008.
- $x_j(t)$ – concentration of gene j's mRNA
- $f(t)$ – concentration of active transcription factor
- Model parameters: baseline b_j, sensitivity s_j and decay d_j
- Application: identifying co-regulated genes (targets)
- Problem: how do we fit the model when $f(t)$ is not observed?

Example: Transcriptional Regulation

- First Order Differential Equation

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

- Can be used as a model of gene transcription: Barenco et al., 2006; Gao et al., 2008.
- $x_j(t)$ – concentration of gene j's mRNA
- $f(t)$ – concentration of active transcription factor
- Model parameters: baseline b_j, sensitivity s_j and decay d_j
- Application: identifying co-regulated genes (targets)
- Problem: how do we fit the model when $f(t)$ is not observed?

Example: Transcriptional Regulation

- First Order Differential Equation

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

- Can be used as a model of gene transcription: Barenco et al., 2006; Gao et al., 2008.
- $x_j(t)$ – concentration of gene j's mRNA
- $f(t)$ – concentration of active transcription factor
- Model parameters: baseline b_j, sensitivity s_j and decay d_j
- Application: identifying co-regulated genes (targets)
- Problem: how do we fit the model when $f(t)$ is not observed?

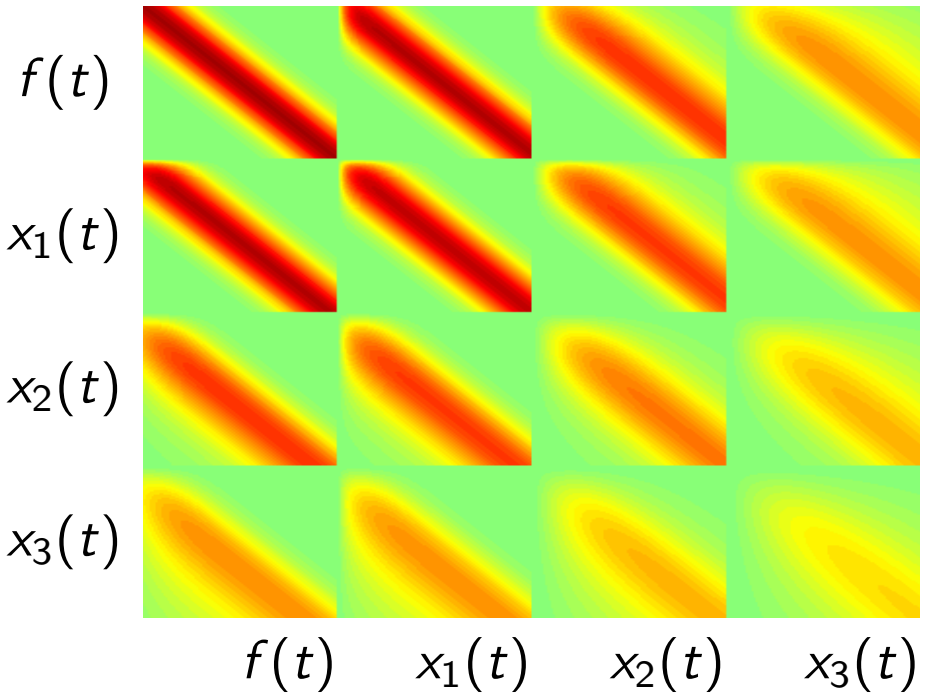
Covariance for Transcription Model

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



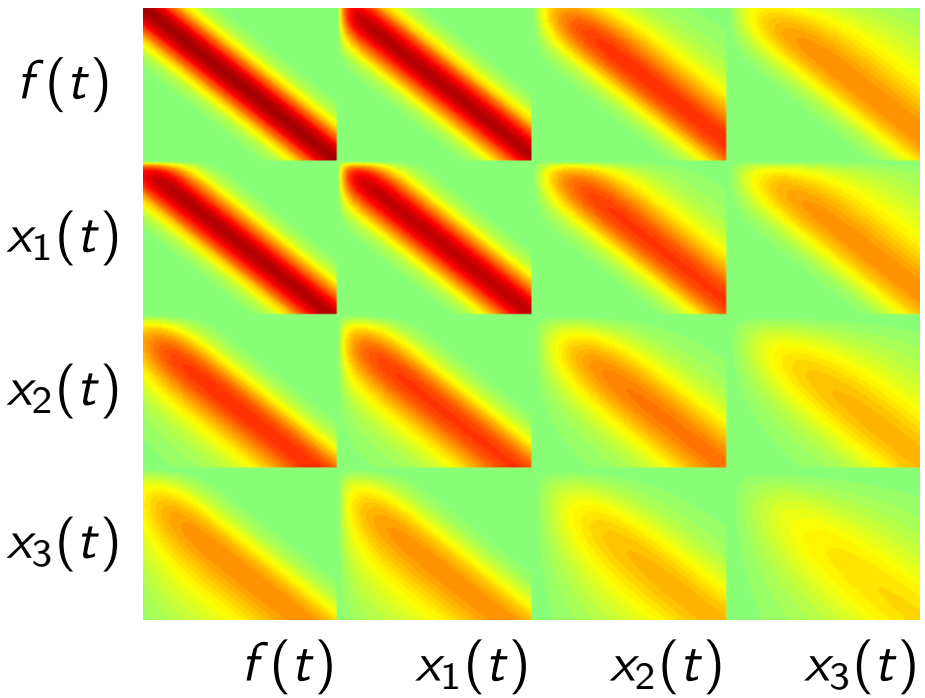
Covariance for Transcription Model

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



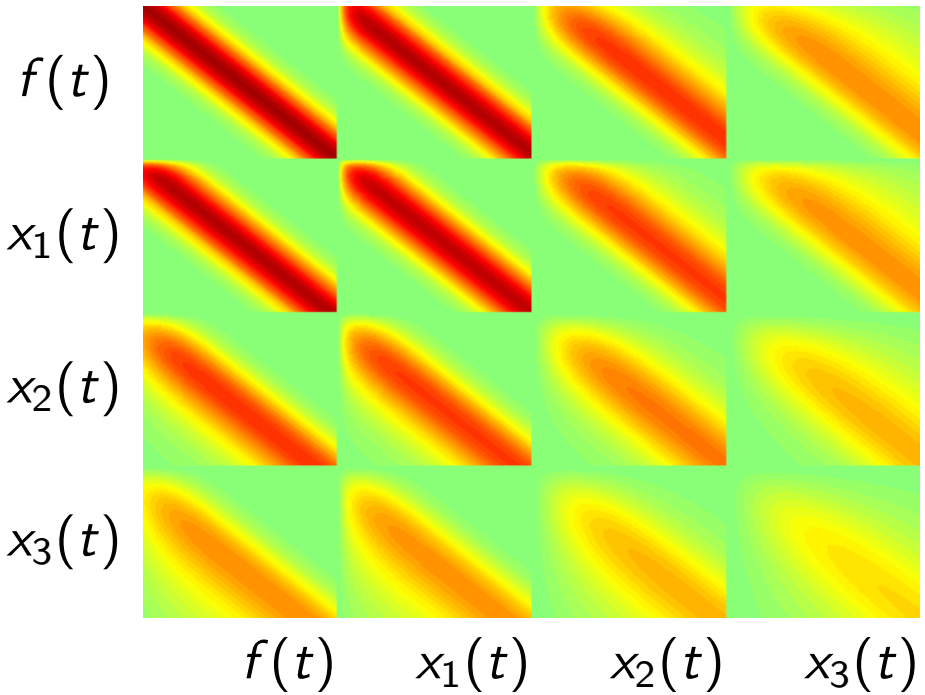
Covariance for Transcription Model

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`

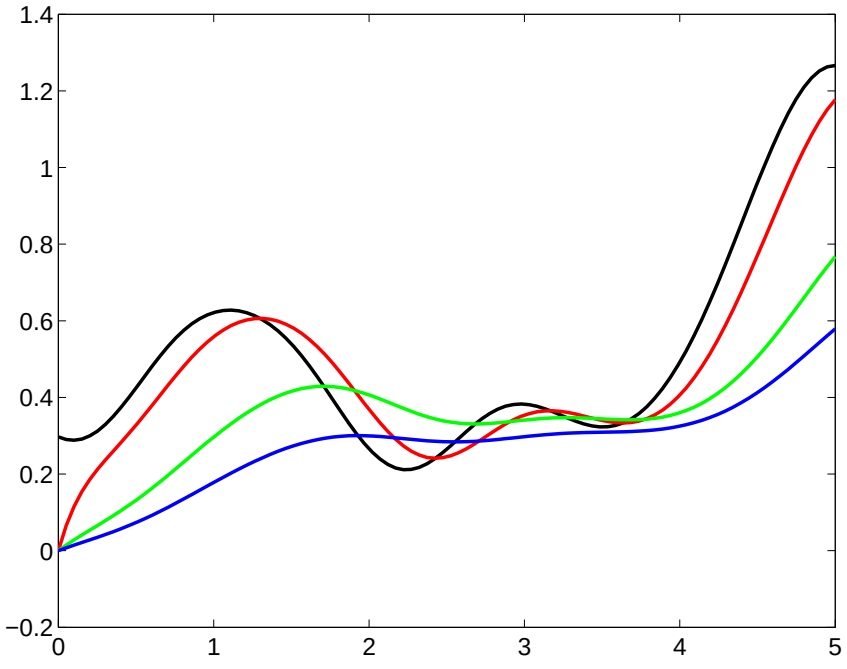


Figure: Joint samples from the ODE covariance, *black*: $f(t)$, *red*: $x_1(t)$ (high decay/sensitivity), *green*: $x_2(t)$ (medium decay/sensitivity) and *blue*: $x_3(t)$ (low decay/sensitivity).

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

- `simSample`

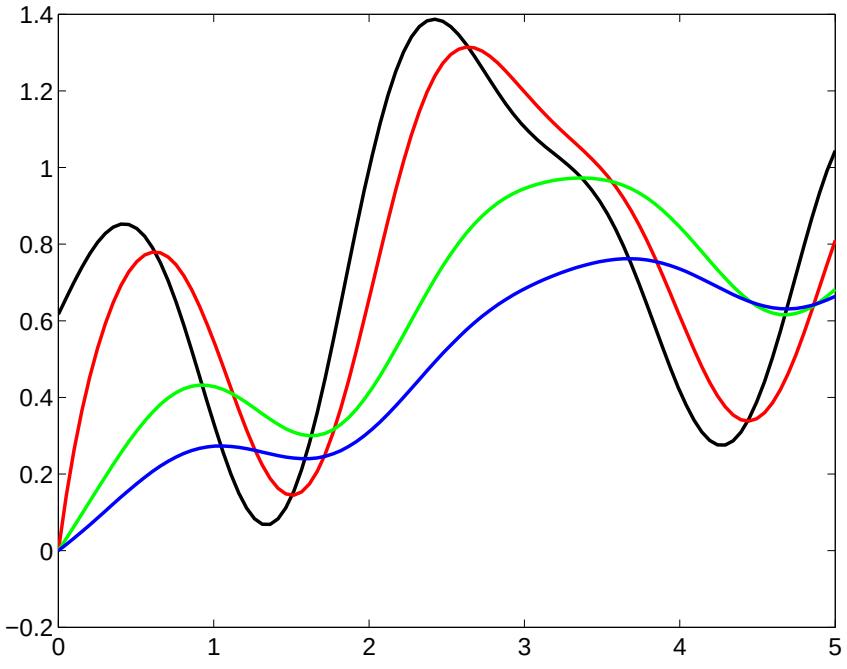


Figure: Joint samples from the ODE covariance, *black*: $f(t)$, *red*: $x_1(t)$ (high decay/sensitivity), *green*: $x_2(t)$ (medium decay/sensitivity) and *blue*: $x_3(t)$ (low decay/sensitivity).

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

- `simSample`

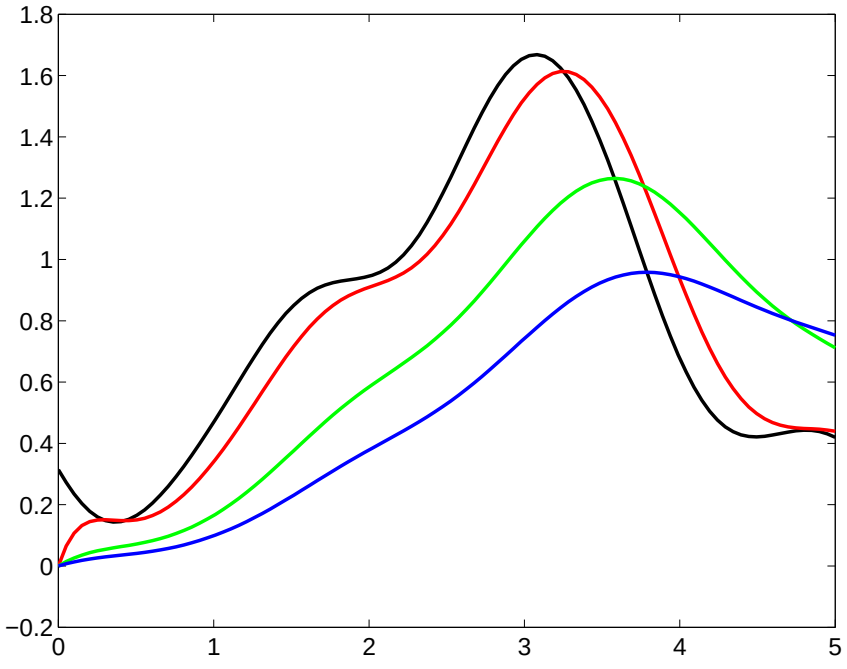


Figure: Joint samples from the ODE covariance, *black*: $f(t)$, *red*: $x_1(t)$ (high decay/sensitivity), *green*: $x_2(t)$ (medium decay/sensitivity) and *blue*: $x_3(t)$ (low decay/sensitivity).

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

- `simSample`

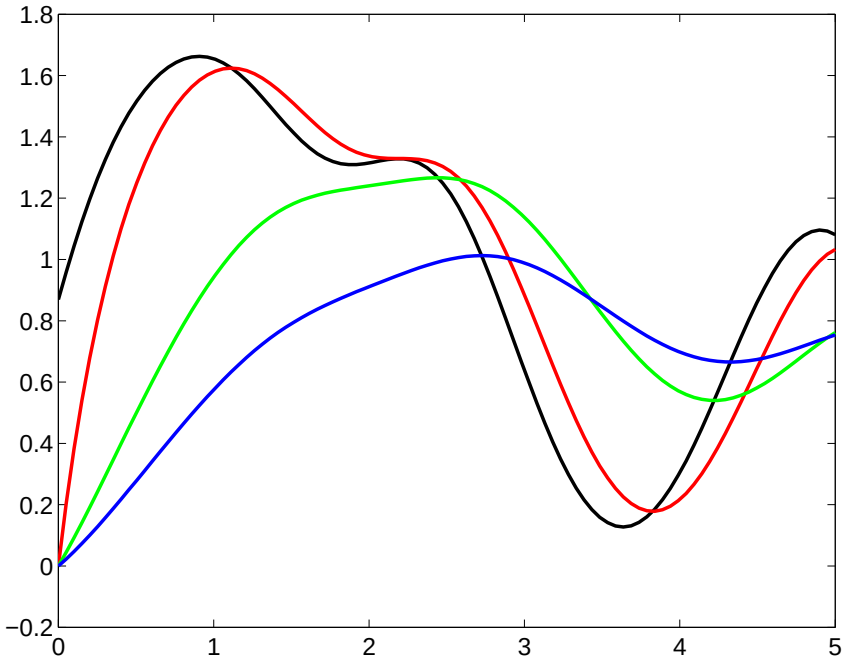
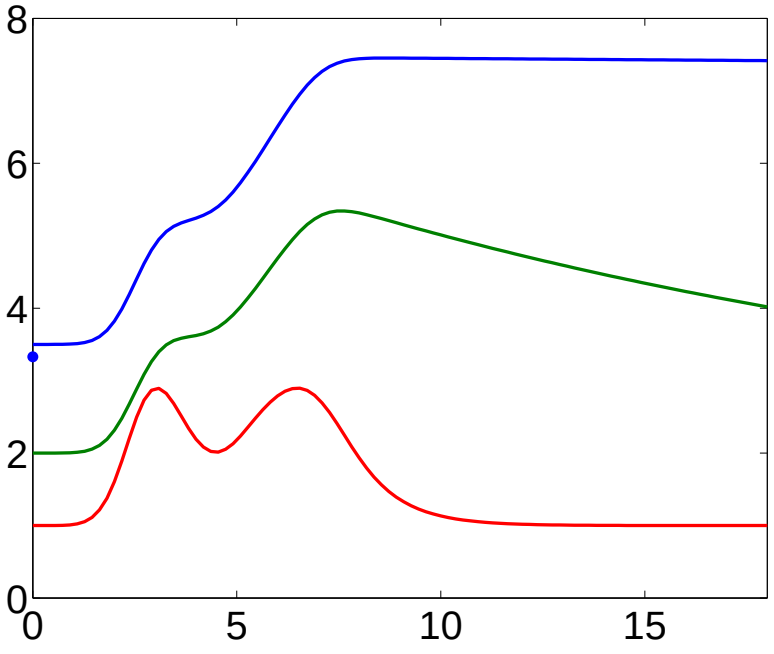


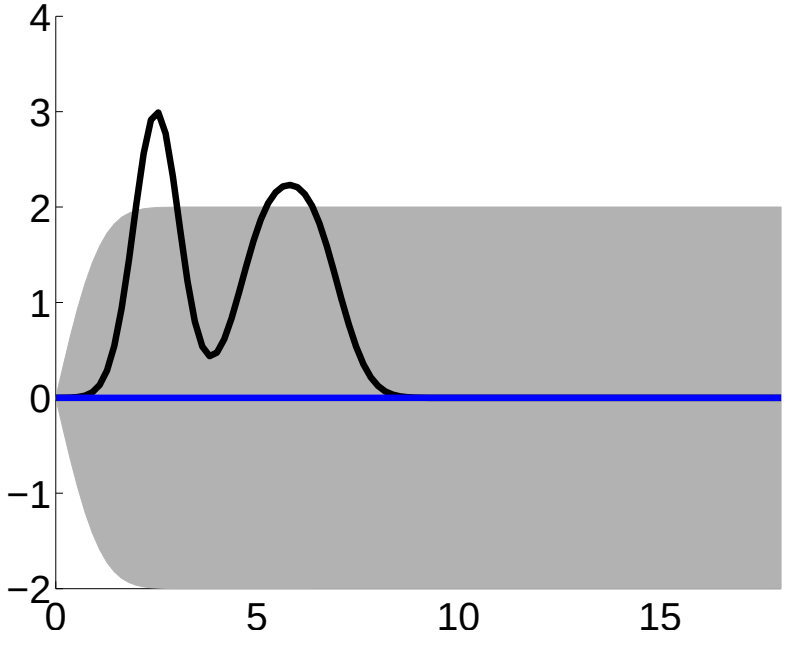
Figure: Joint samples from the ODE covariance, *black*: $f(t)$, *red*: $x_1(t)$ (high decay/sensitivity), *green*: $x_2(t)$ (medium decay/sensitivity) and *blue*: $x_3(t)$ (low decay/sensitivity).

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



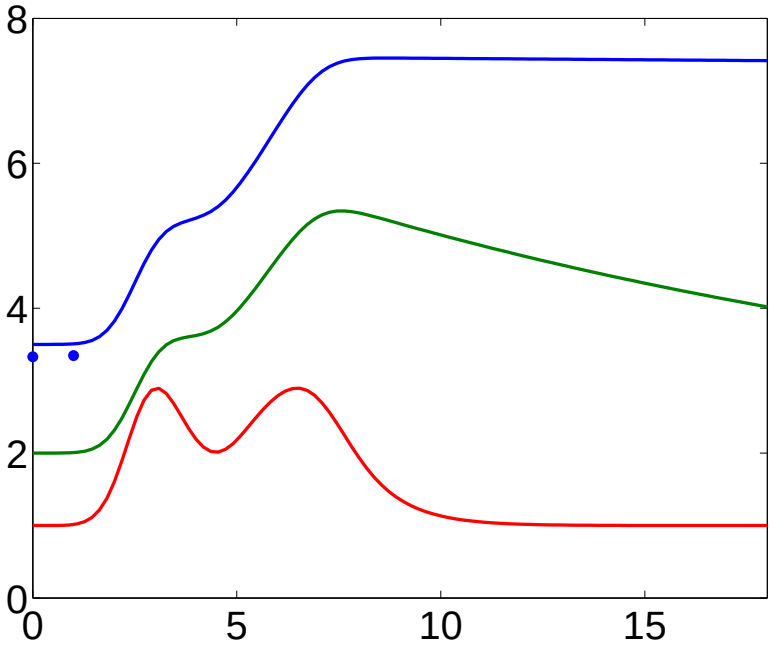
True "gene profiles" and noisy observations.



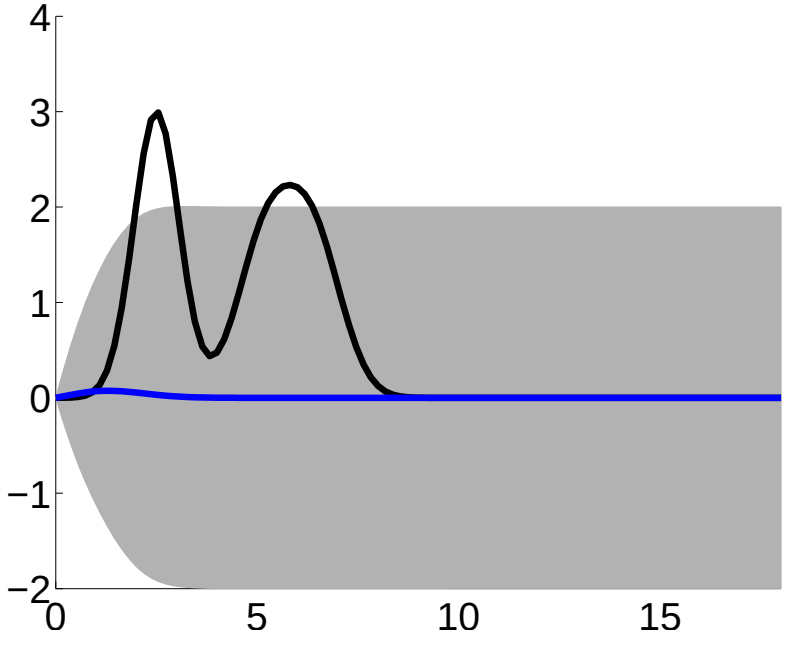
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



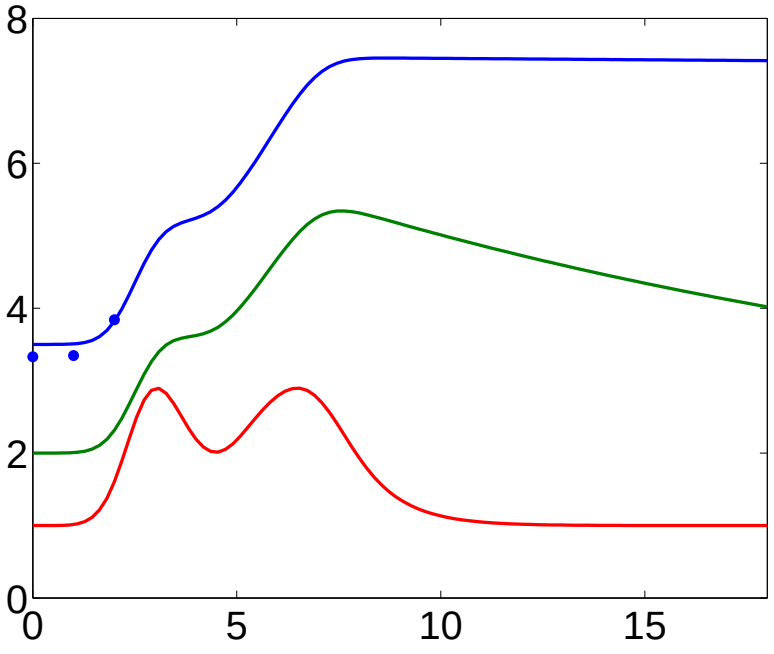
True "gene profiles" and noisy observations.



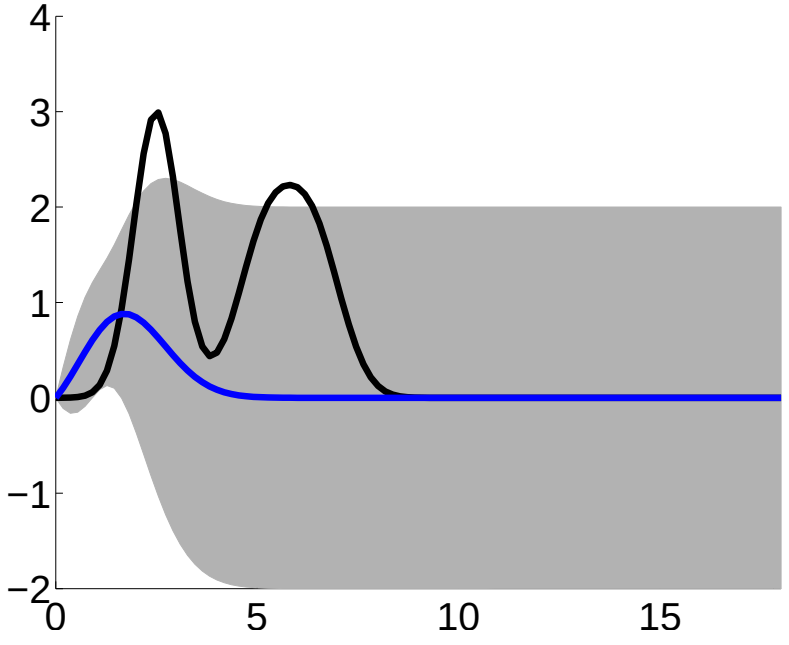
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



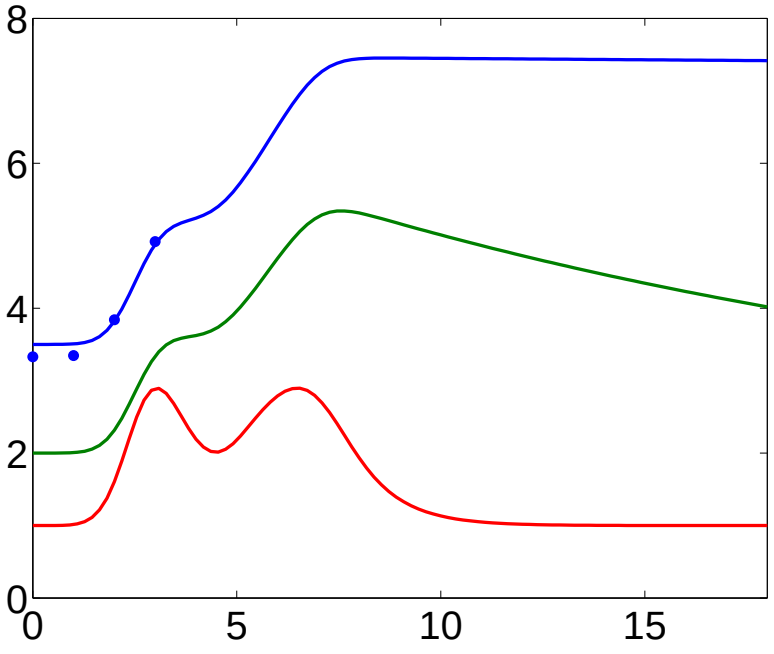
True "gene profiles" and noisy observations.



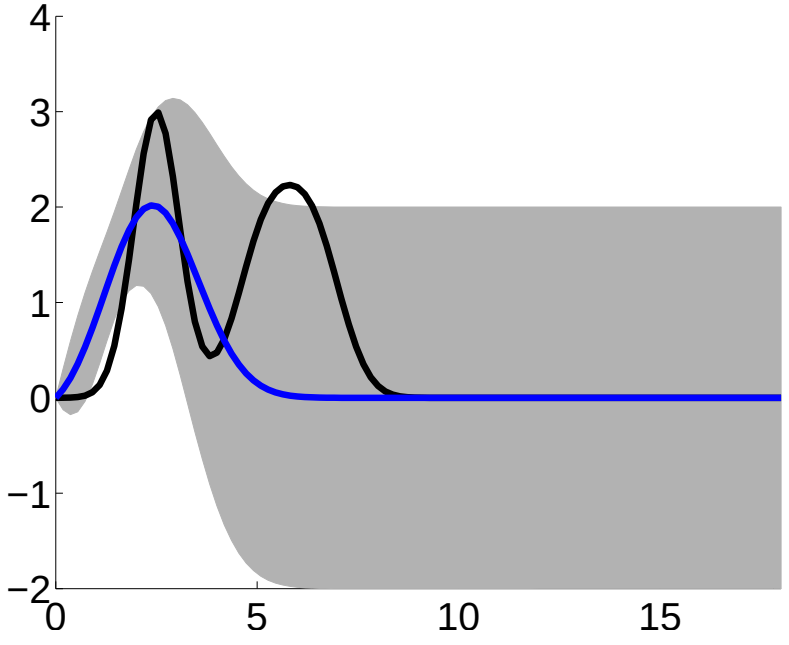
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



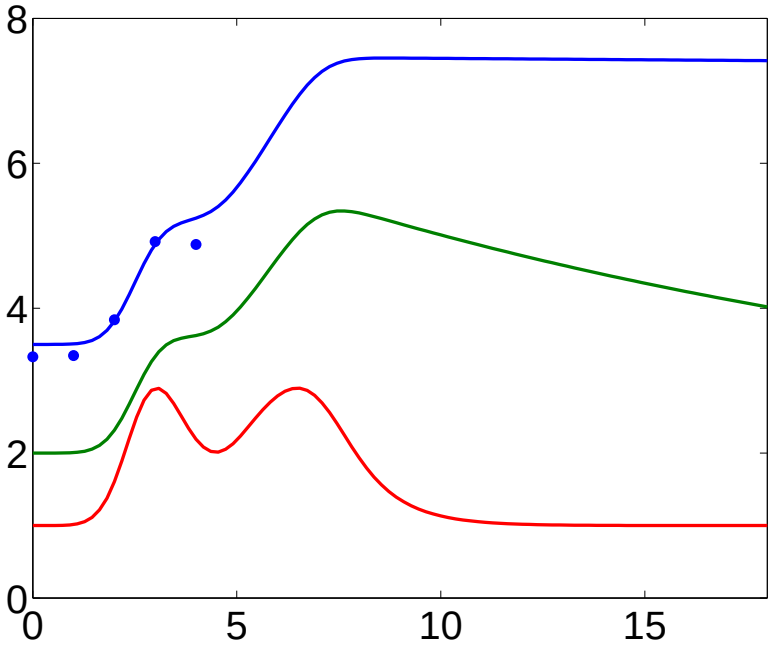
True "gene profiles" and noisy observations.



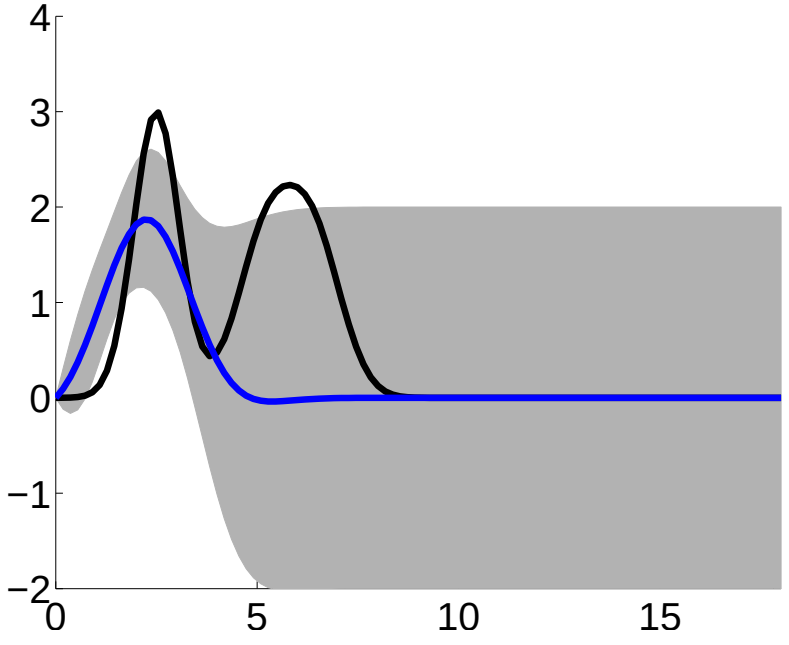
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



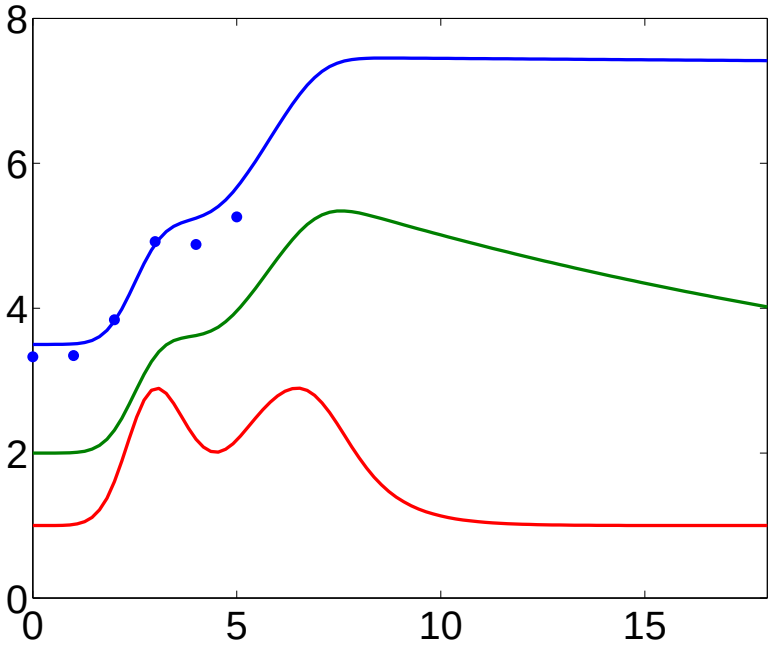
True "gene profiles" and noisy observations.



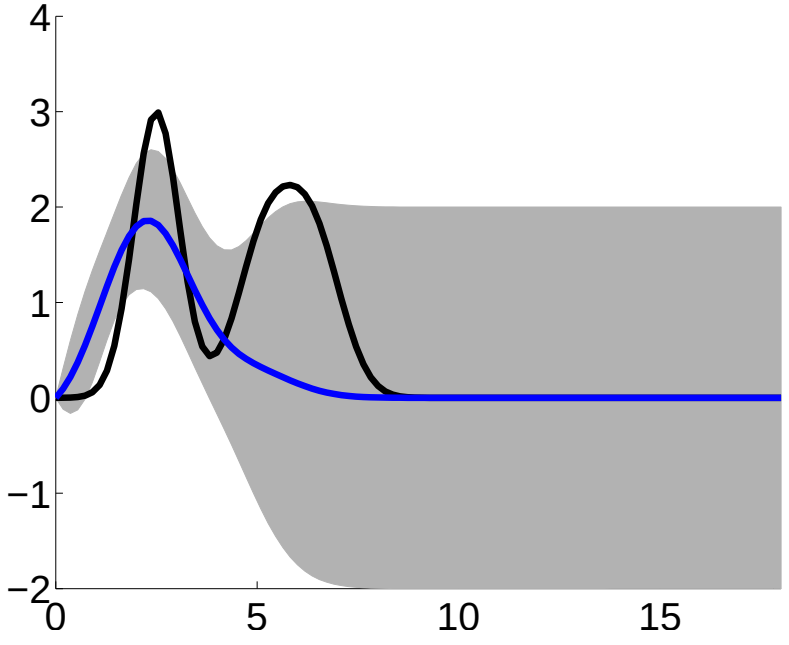
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



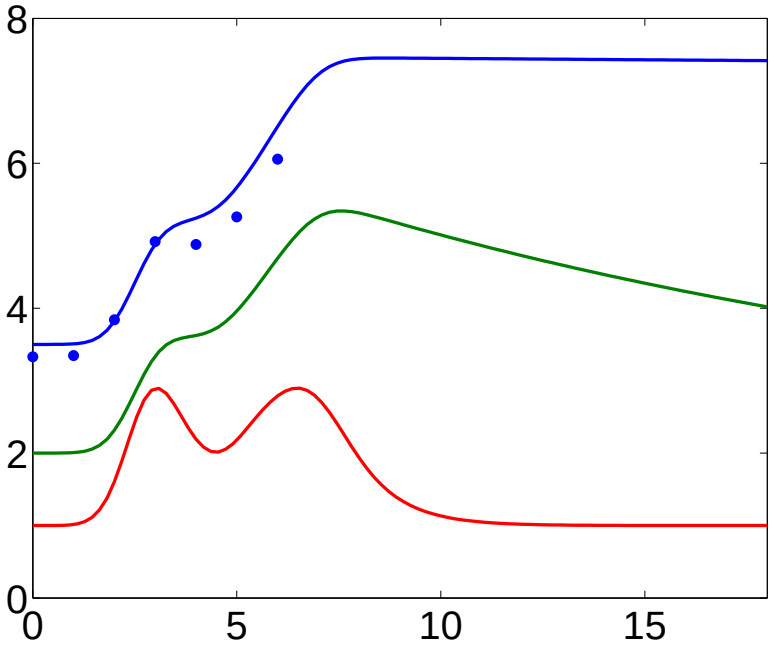
True "gene profiles" and noisy observations.



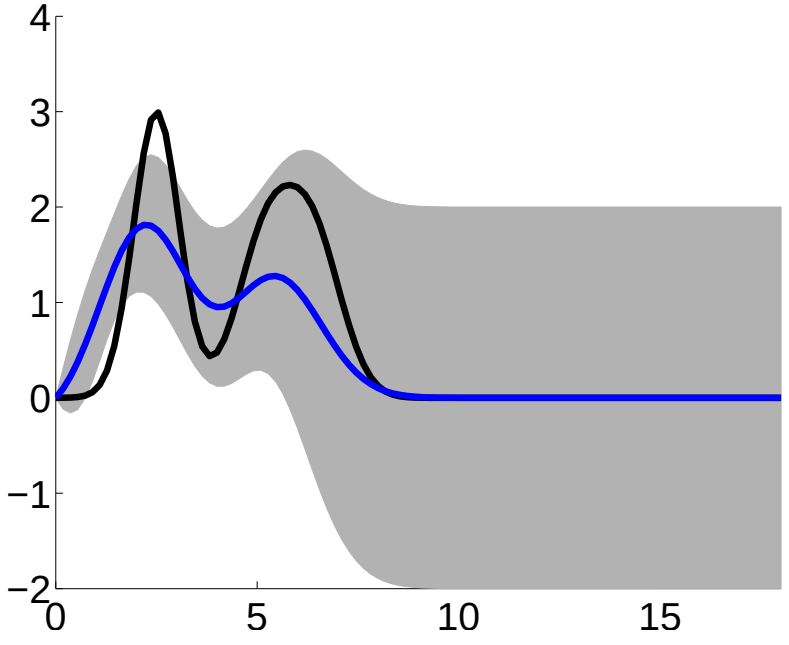
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



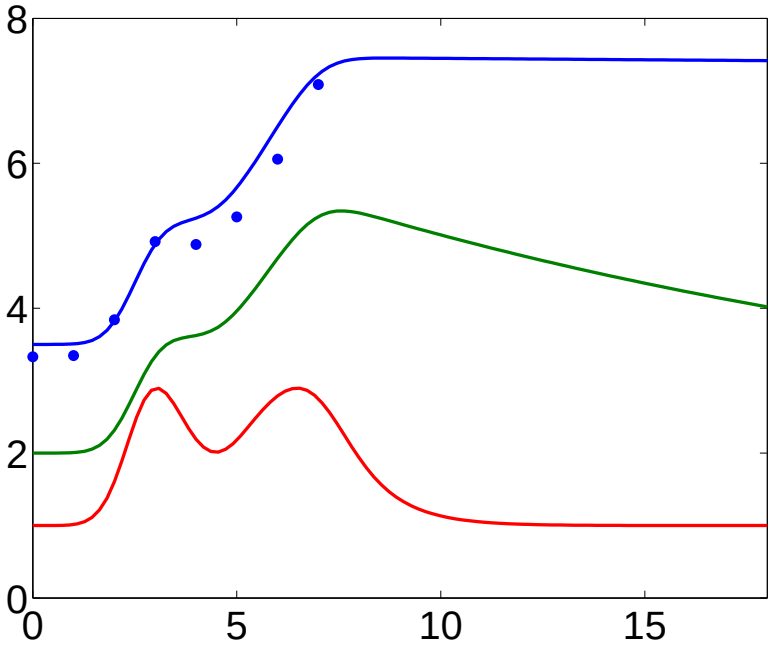
True "gene profiles" and noisy observations.



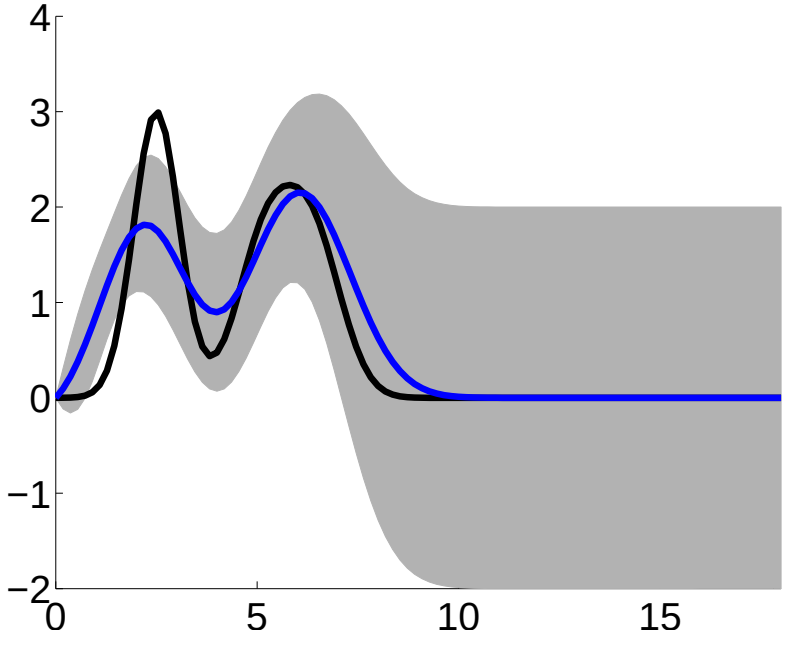
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



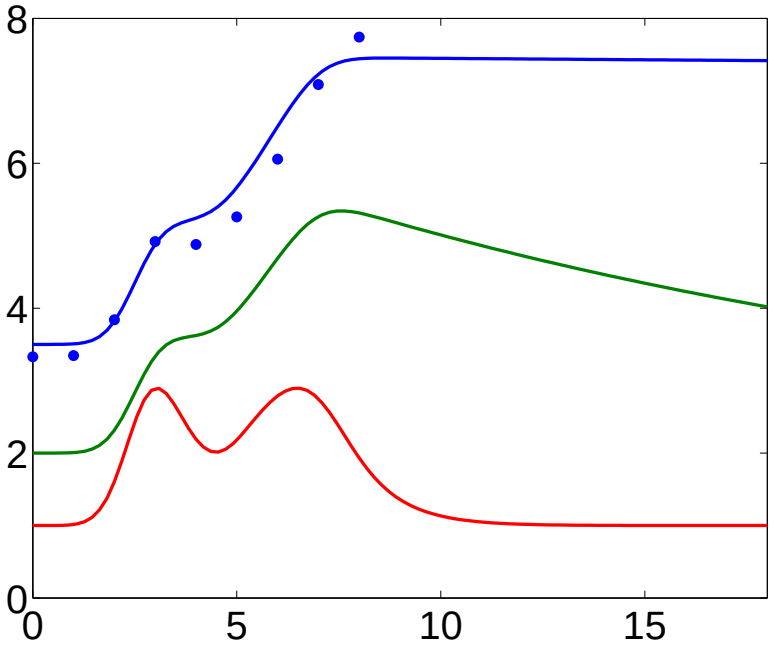
True "gene profiles" and noisy observations.



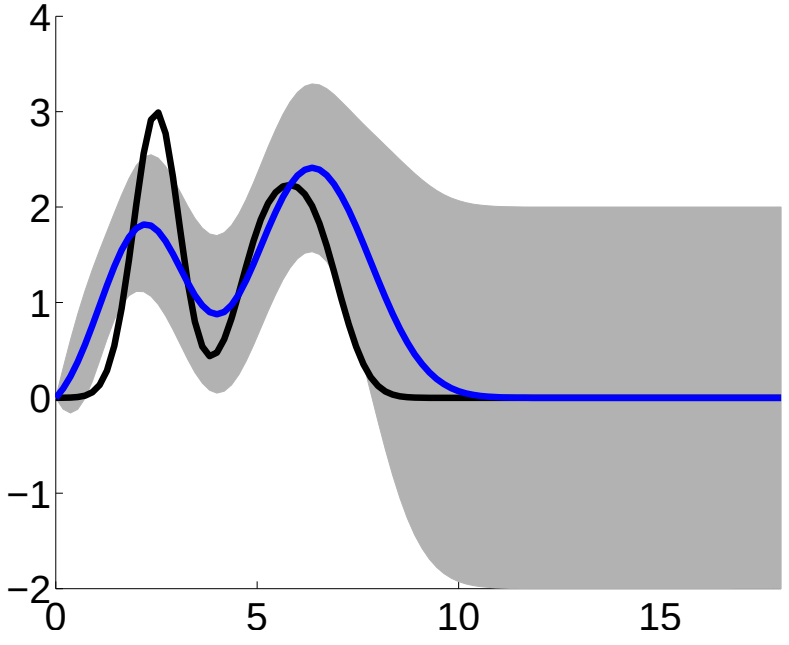
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



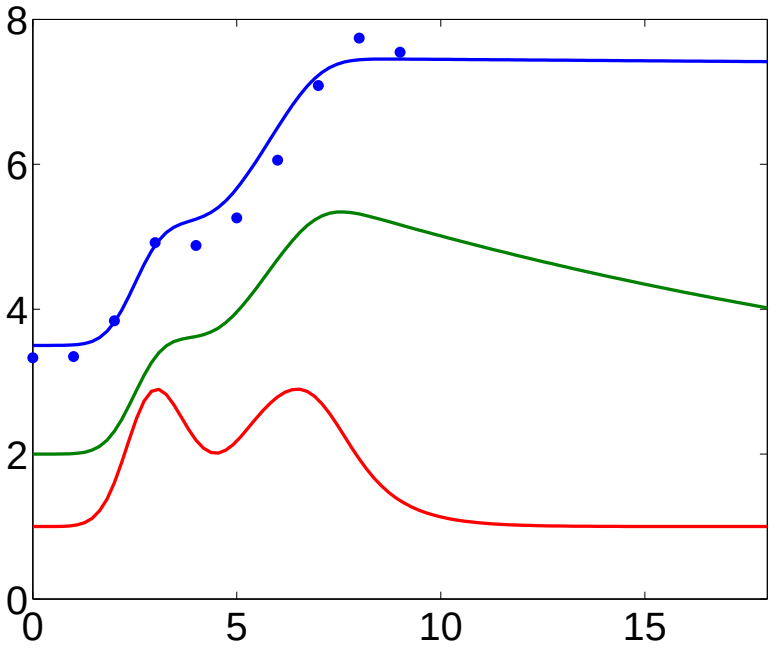
True "gene profiles" and noisy observations.



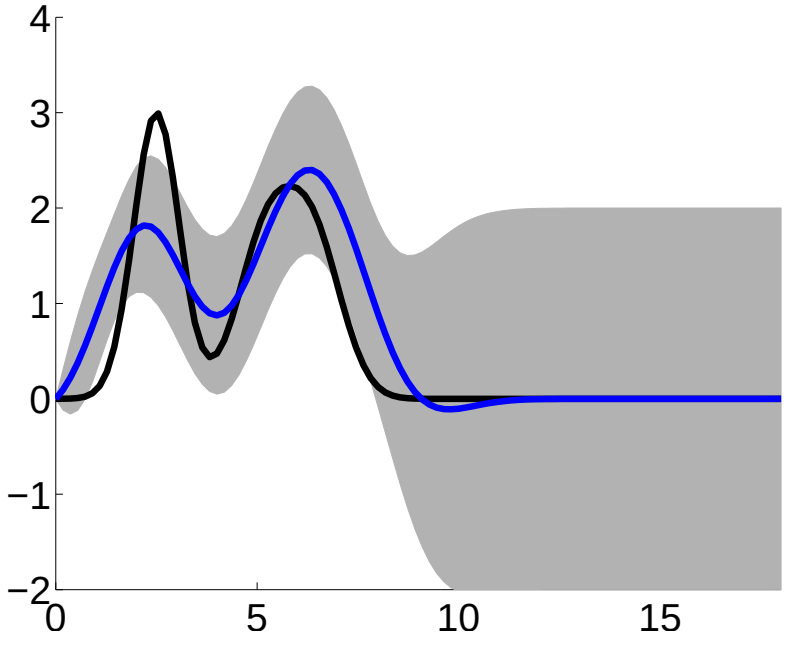
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



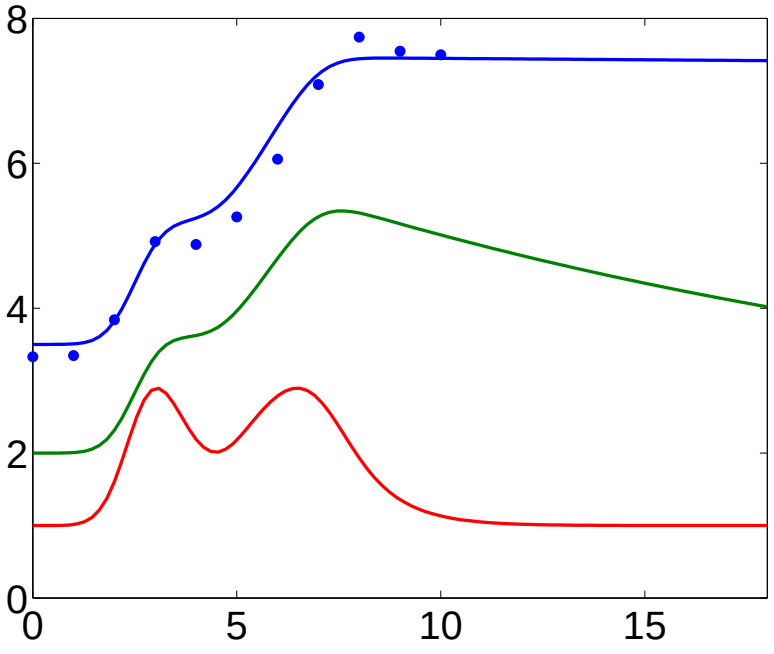
True "gene profiles" and noisy observations.



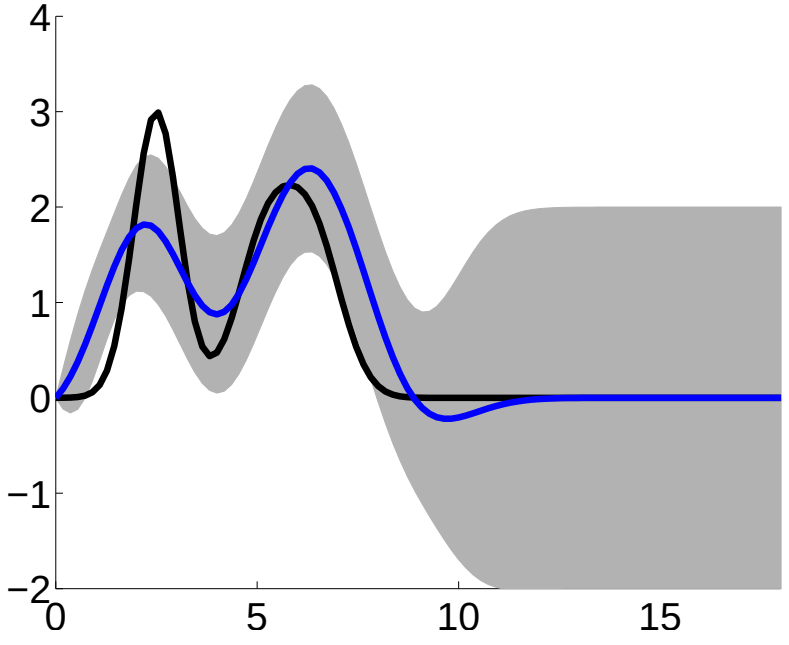
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



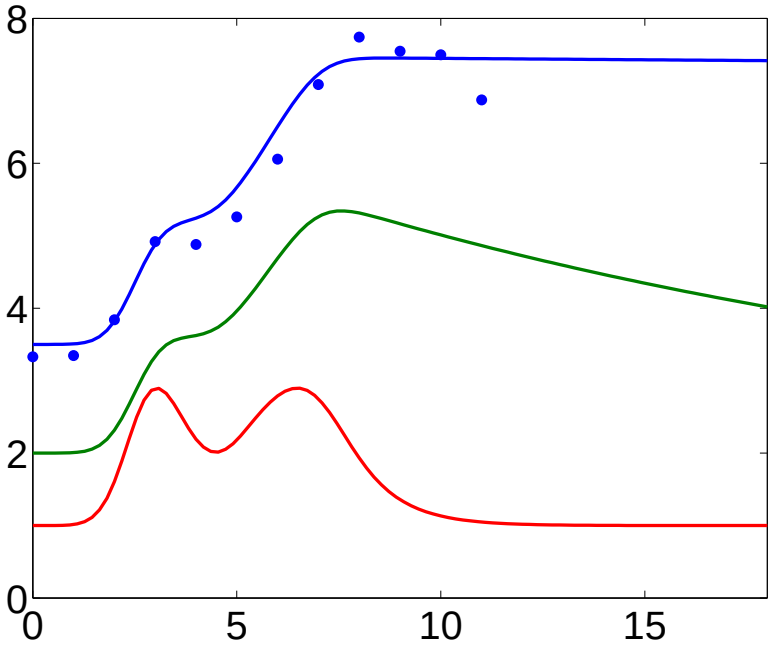
True "gene profiles" and noisy observations.



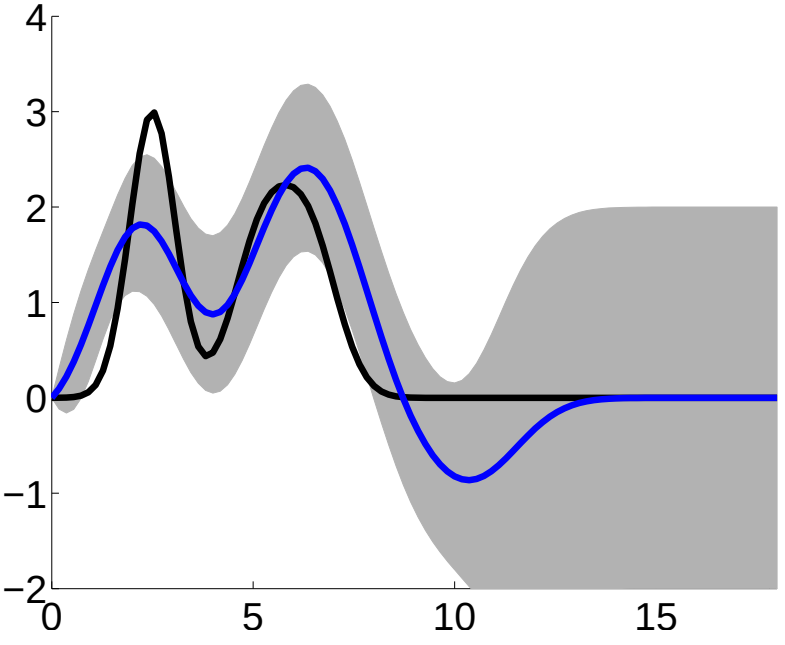
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



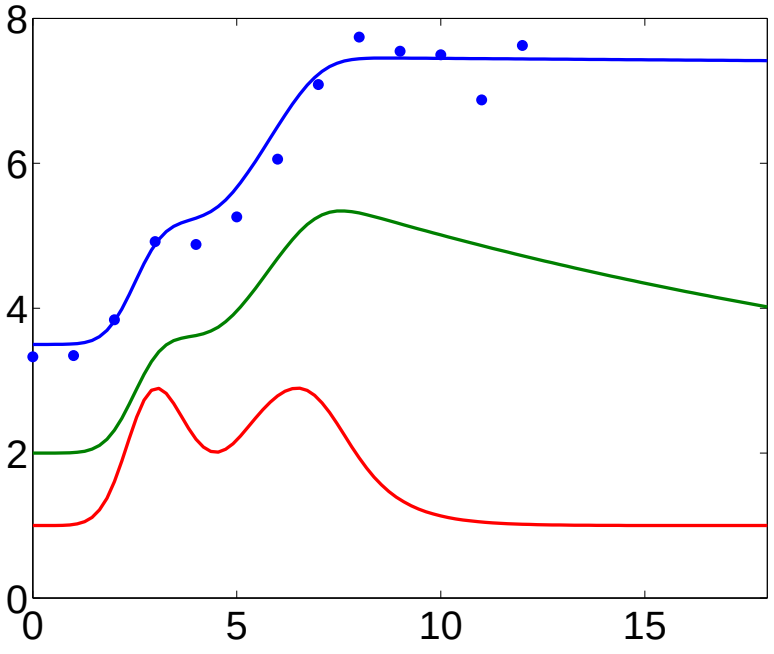
True "gene profiles" and noisy observations.



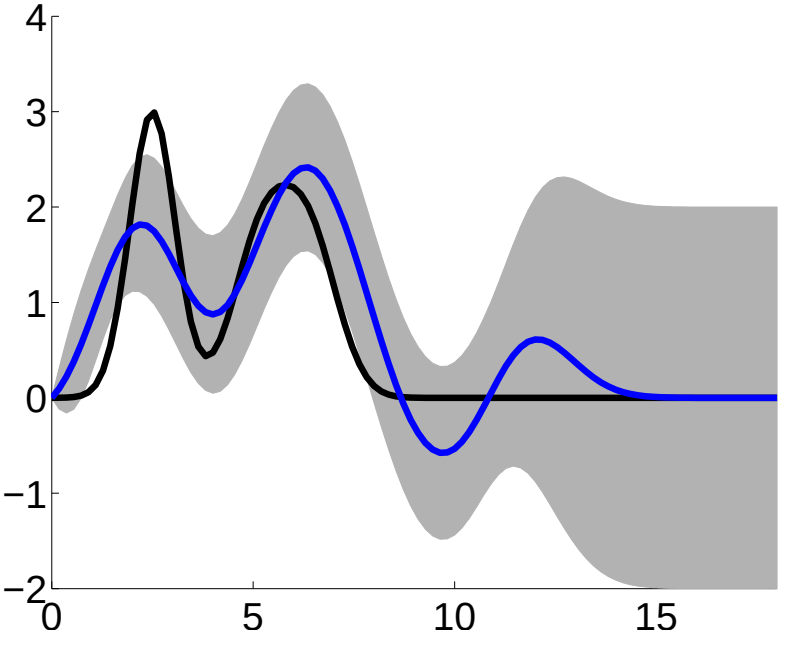
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



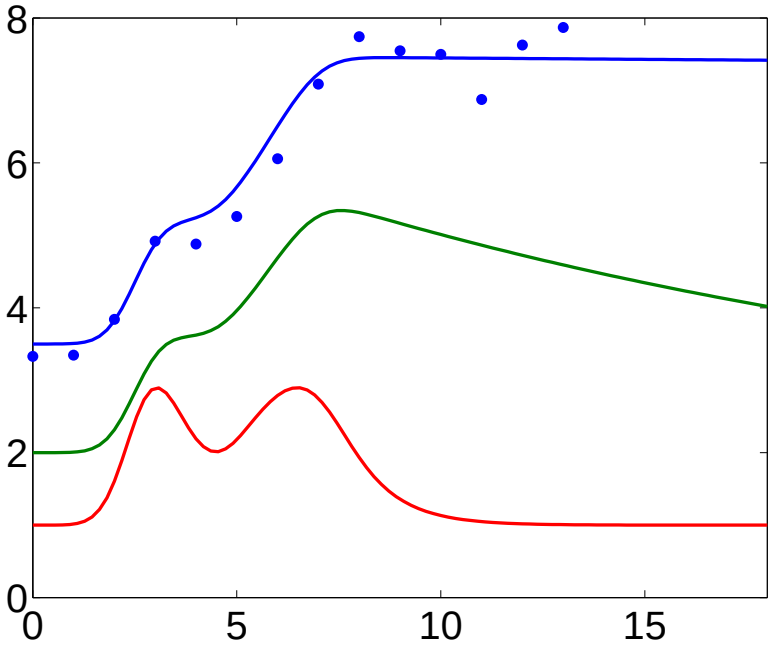
True "gene profiles" and noisy observations.



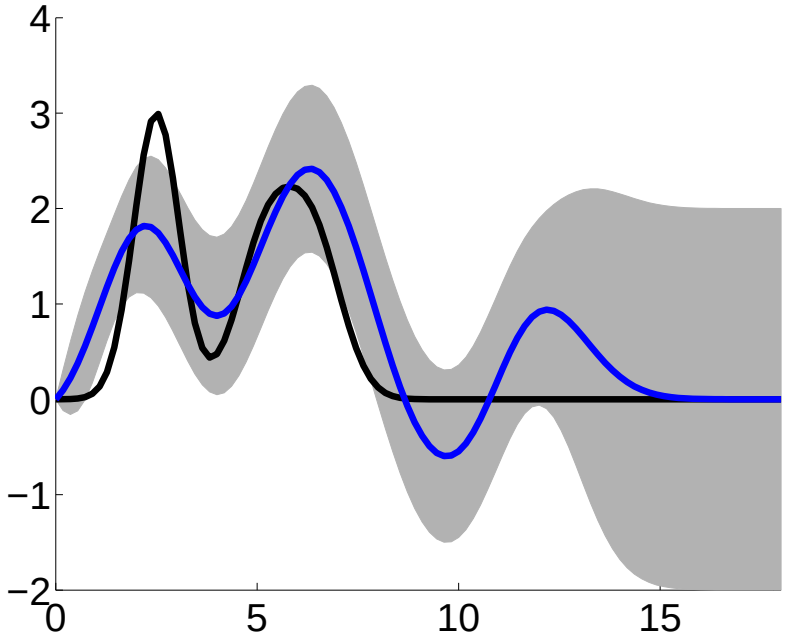
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



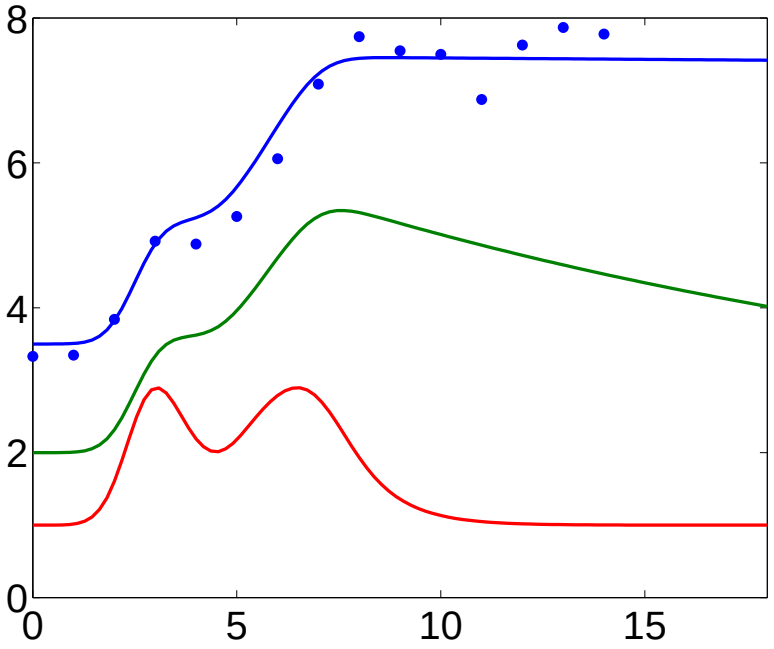
True "gene profiles" and noisy observations.



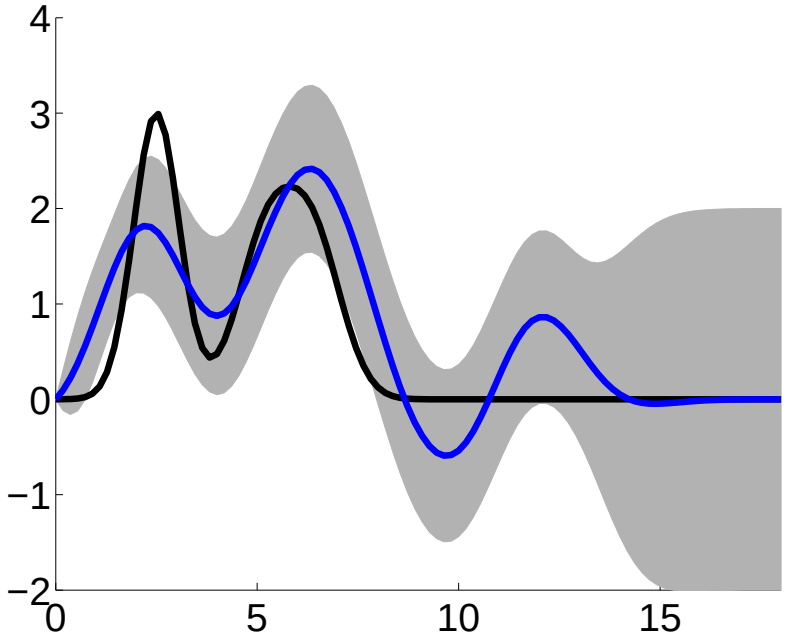
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



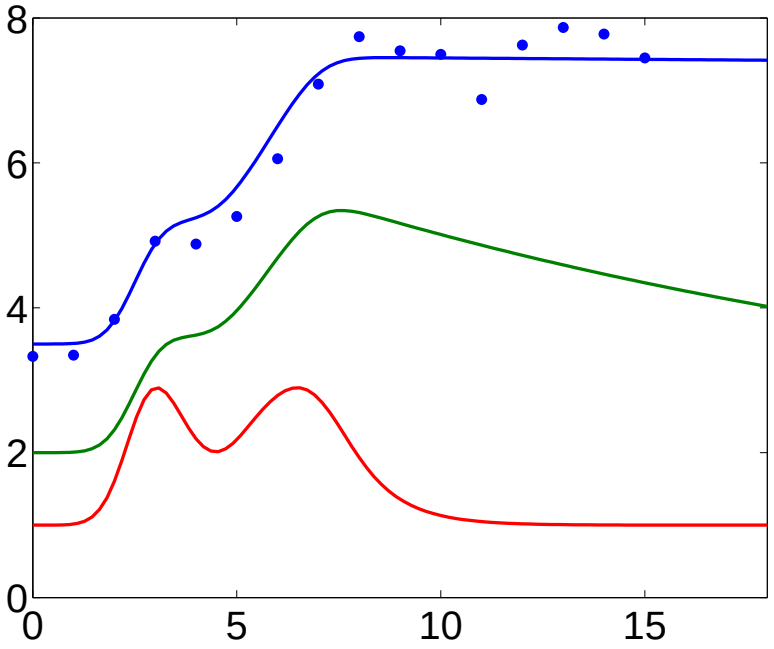
True "gene profiles" and noisy observations.



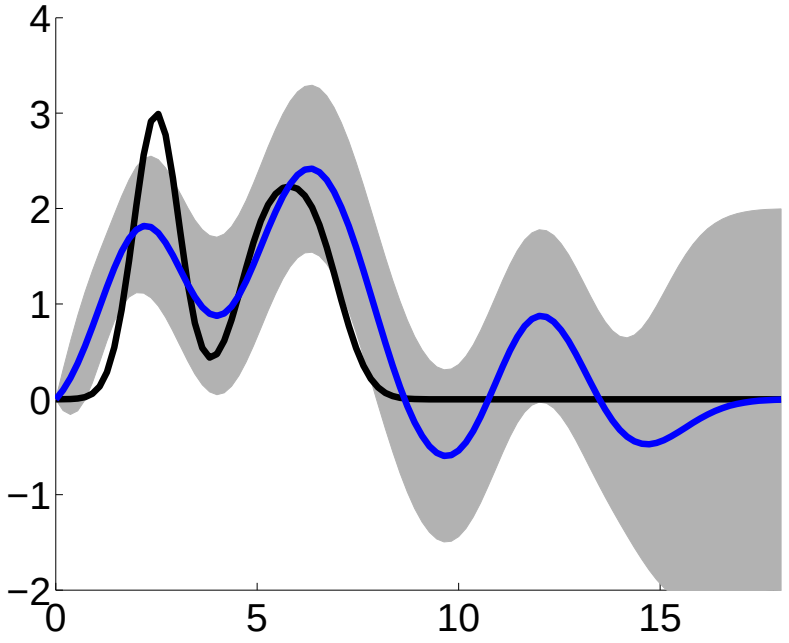
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



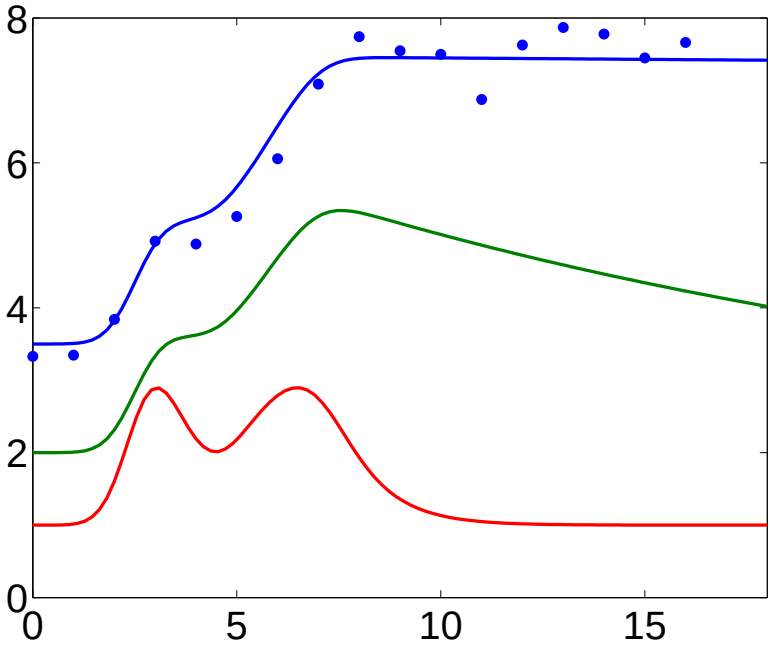
True "gene profiles" and noisy observations.



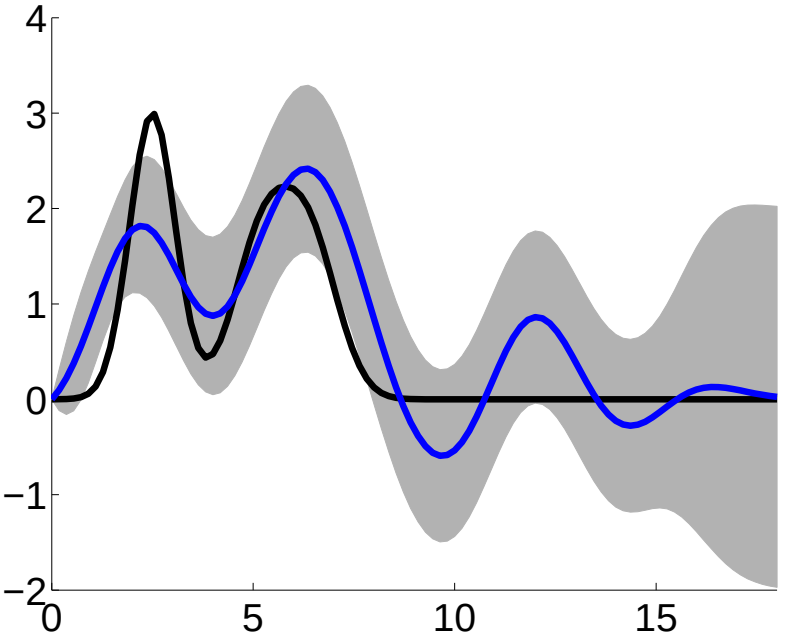
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



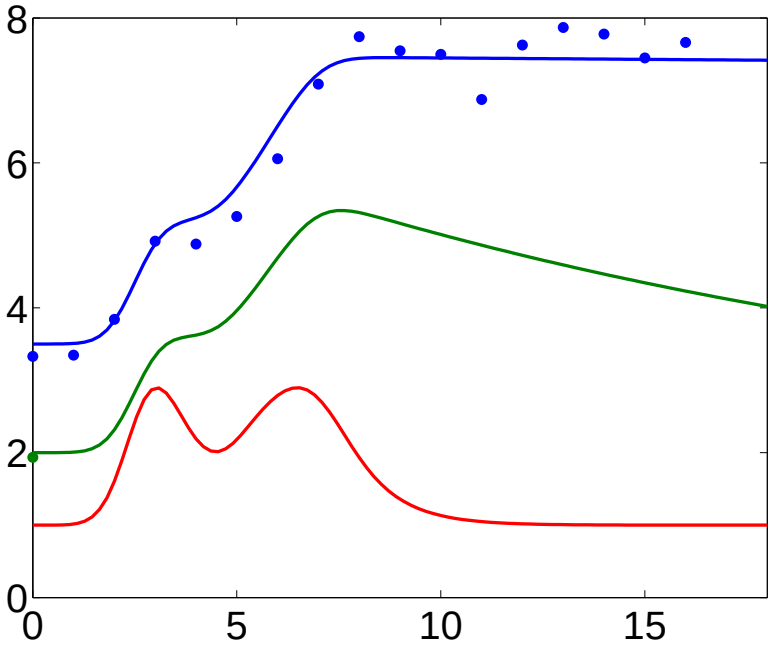
True "gene profiles" and noisy observations.



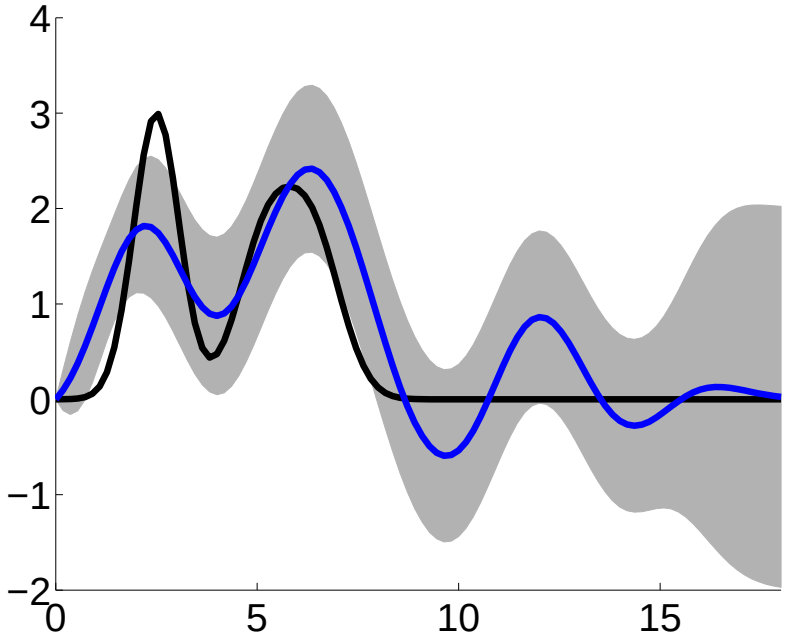
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



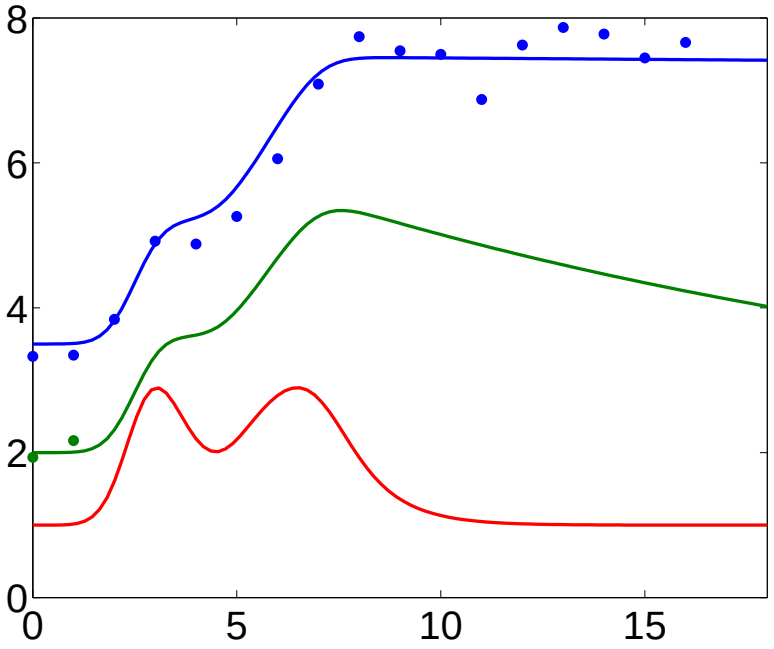
True "gene profiles" and noisy observations.



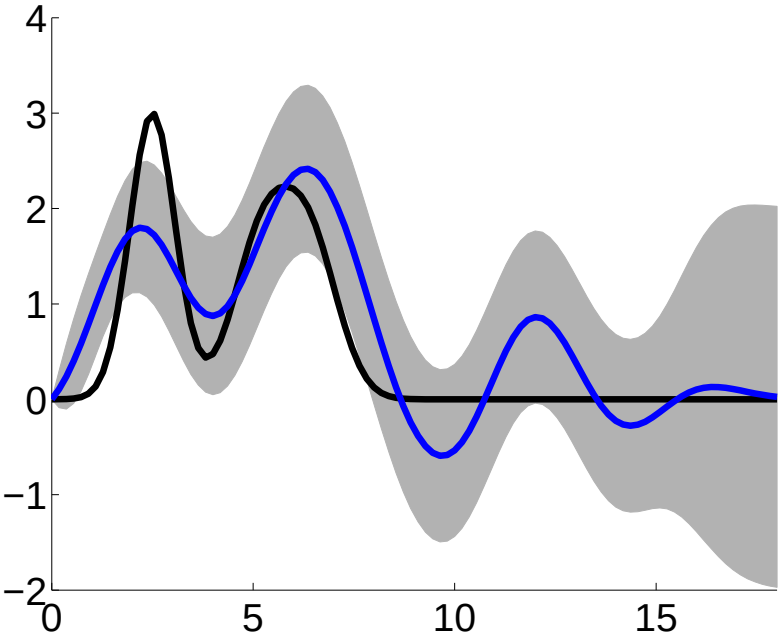
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



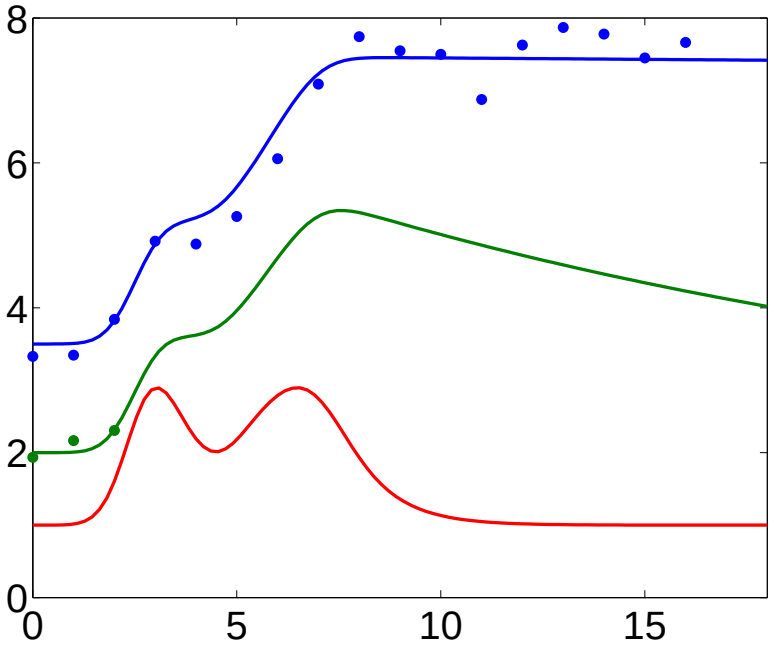
True "gene profiles" and noisy observations.



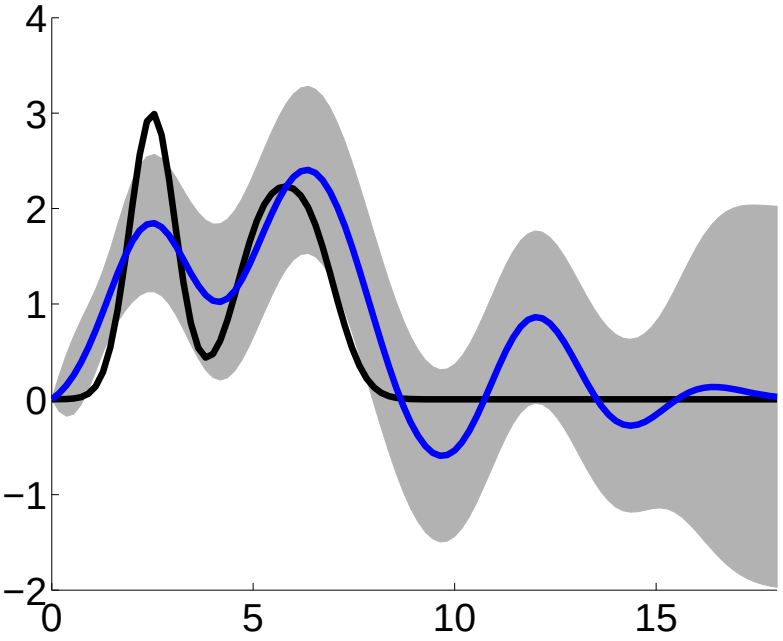
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



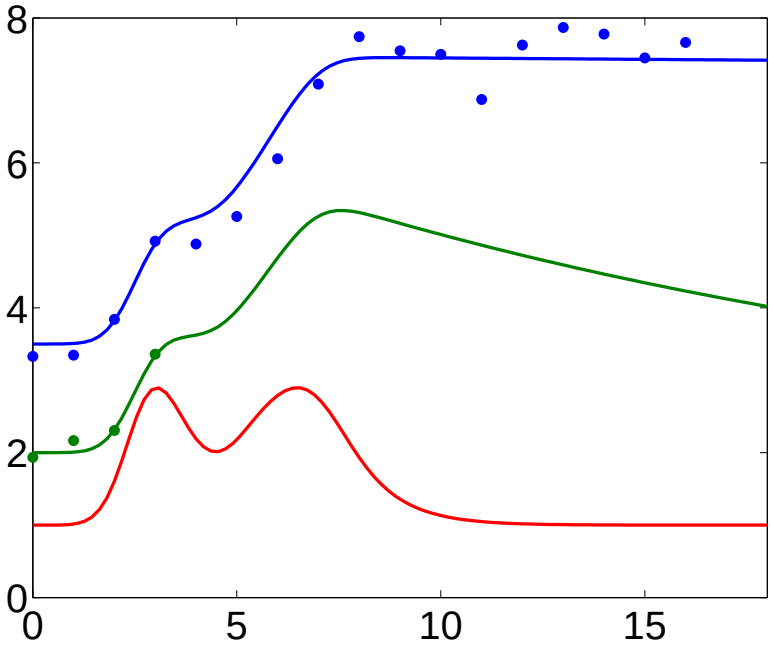
True "gene profiles" and noisy observations.



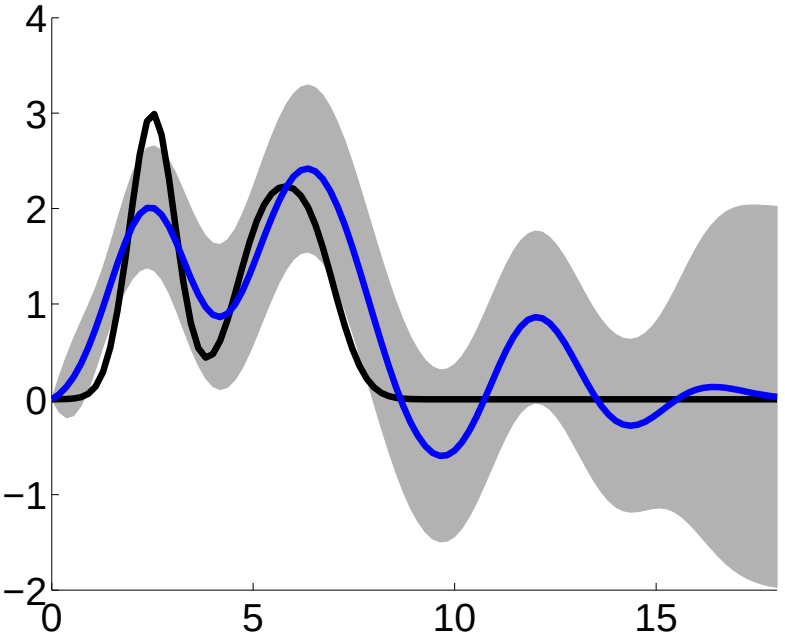
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



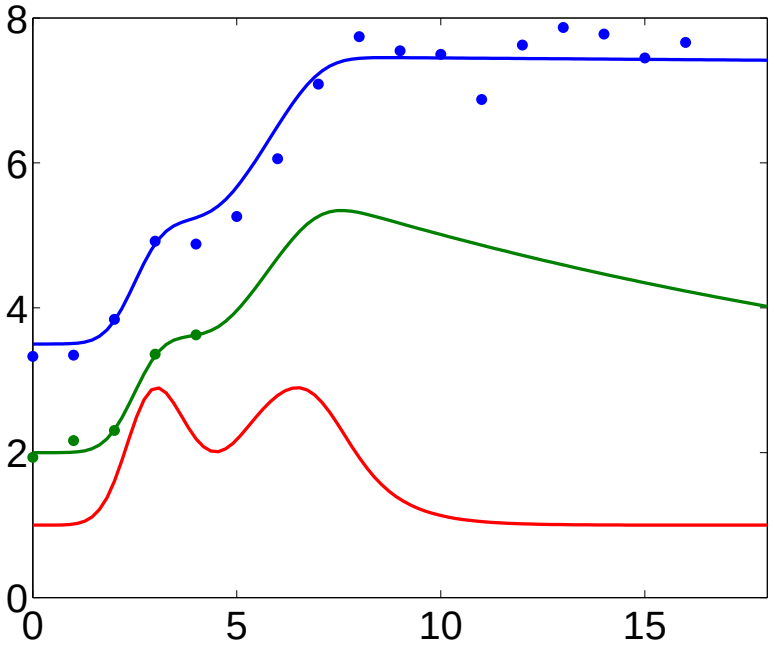
True "gene profiles" and noisy observations.



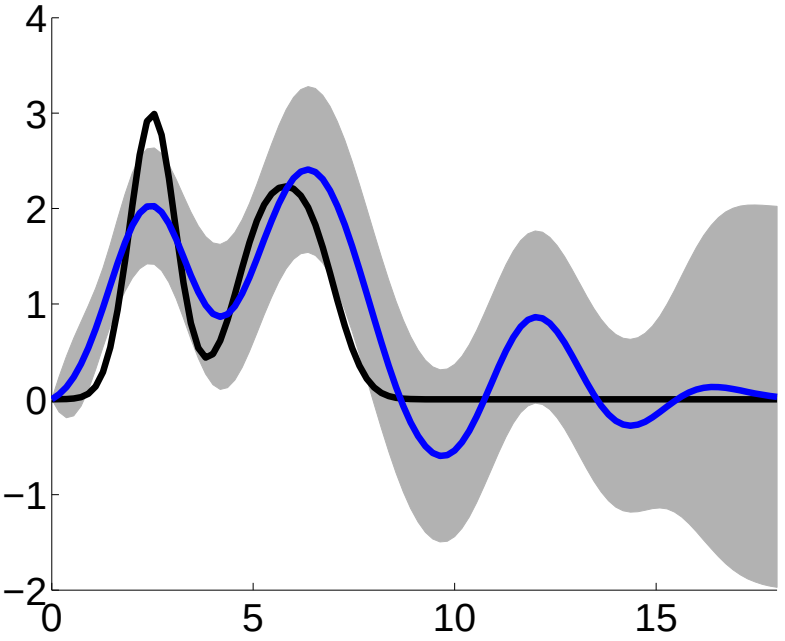
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



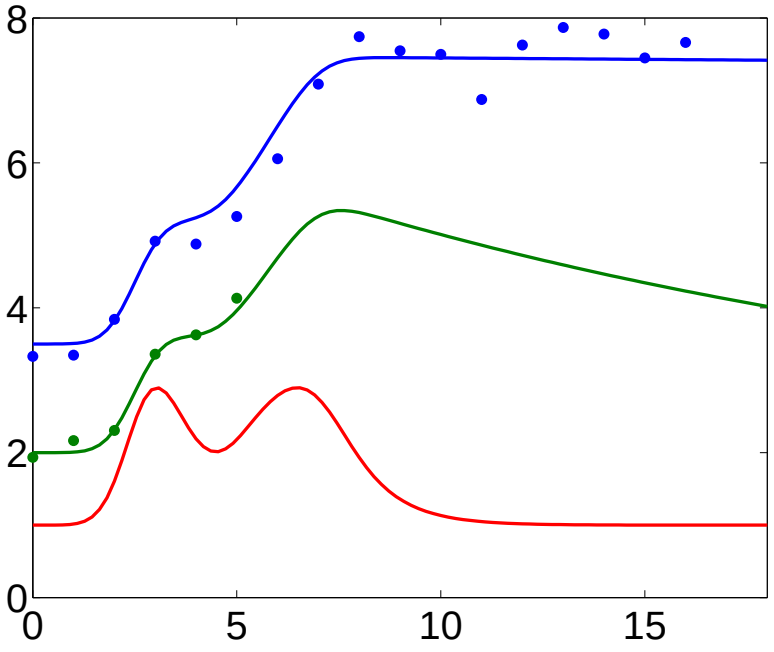
True "gene profiles" and noisy observations.



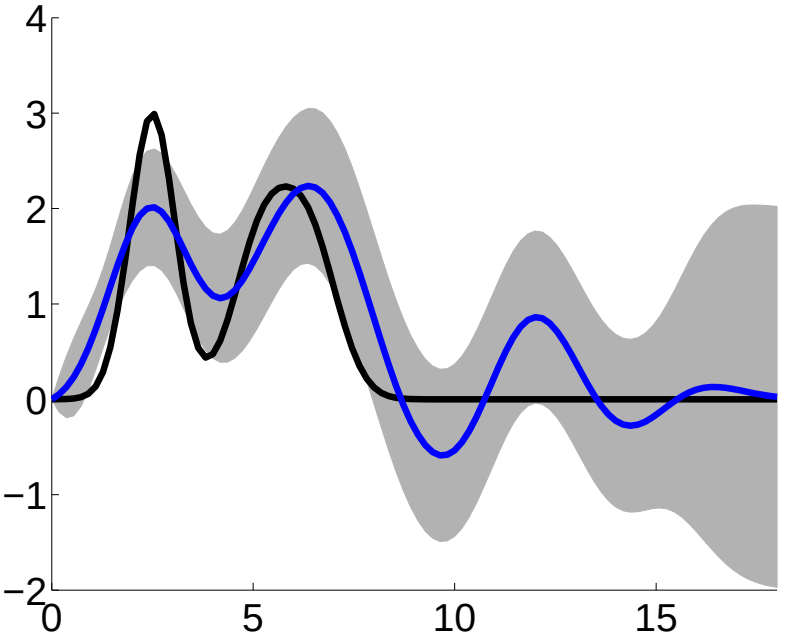
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



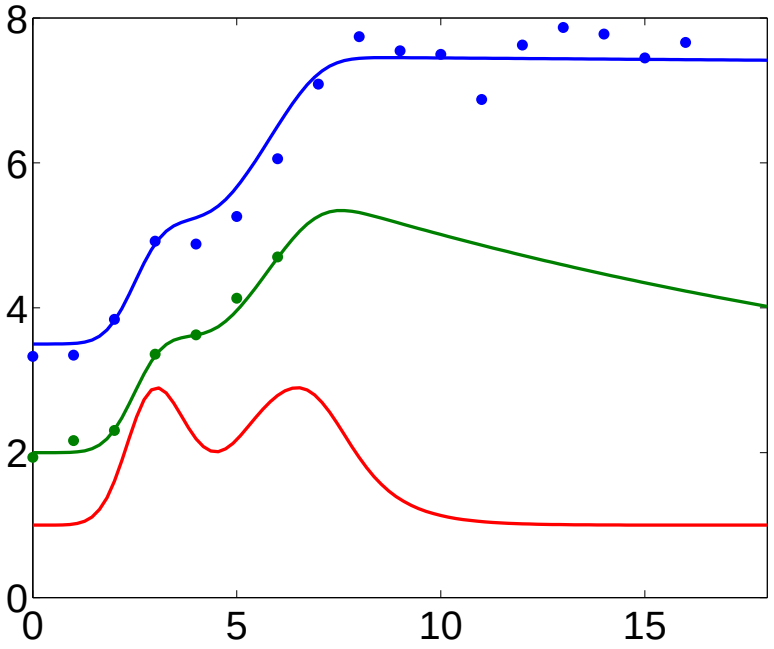
True "gene profiles" and noisy observations.



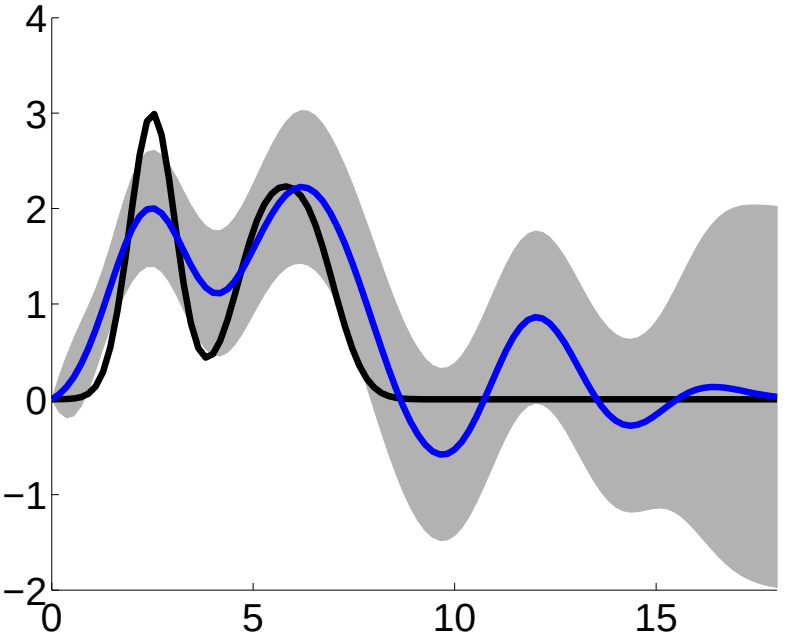
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



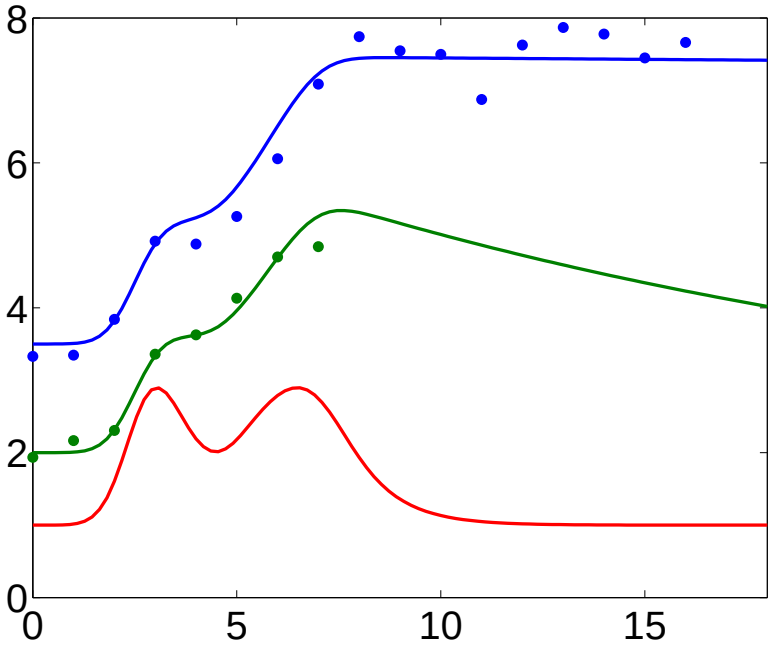
True "gene profiles" and noisy observations.



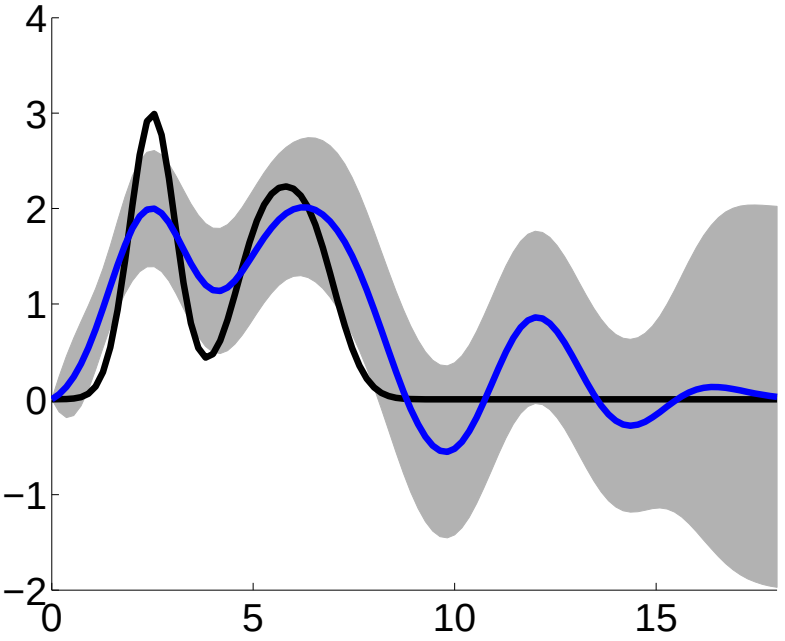
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



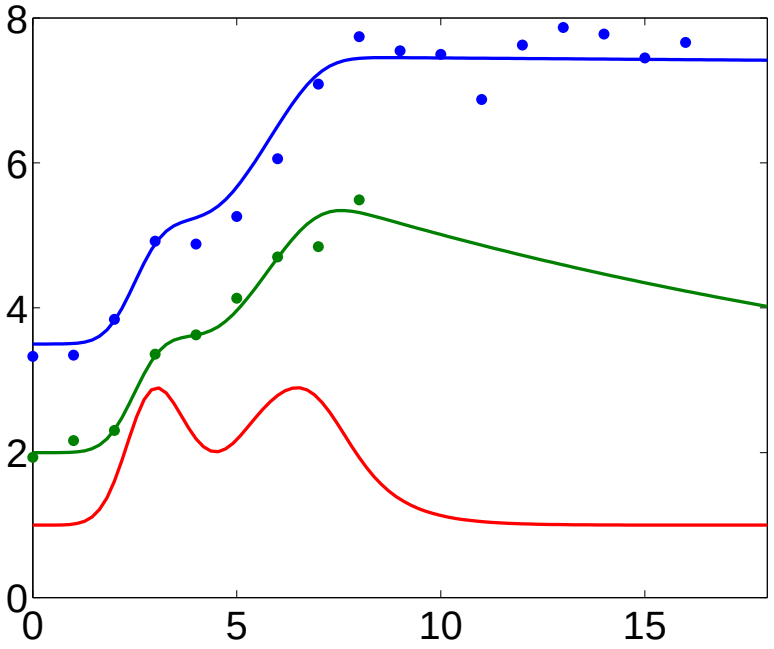
True "gene profiles" and noisy observations.



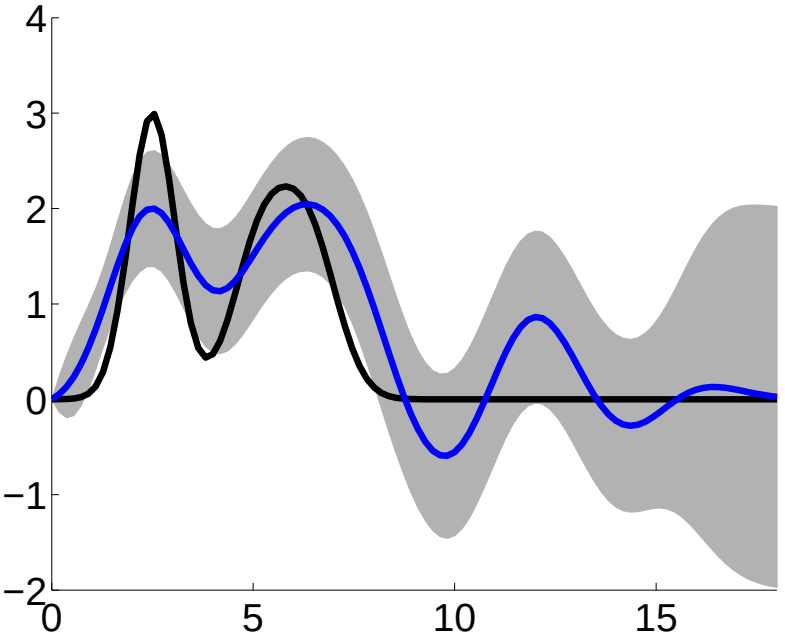
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



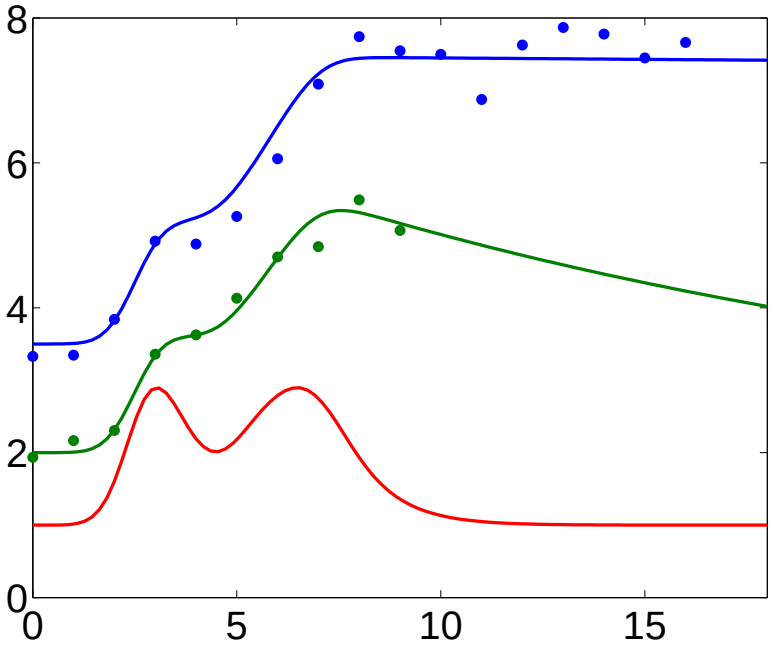
True "gene profiles" and noisy observations.



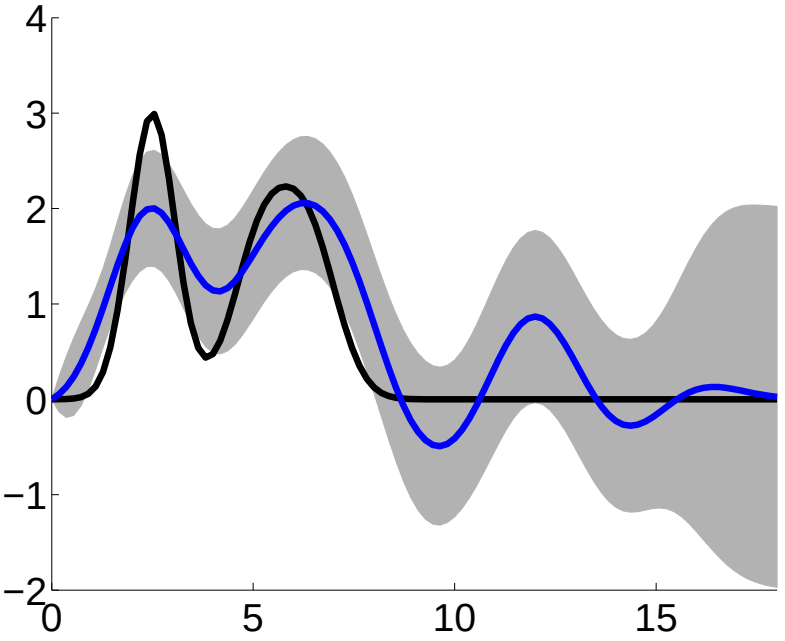
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



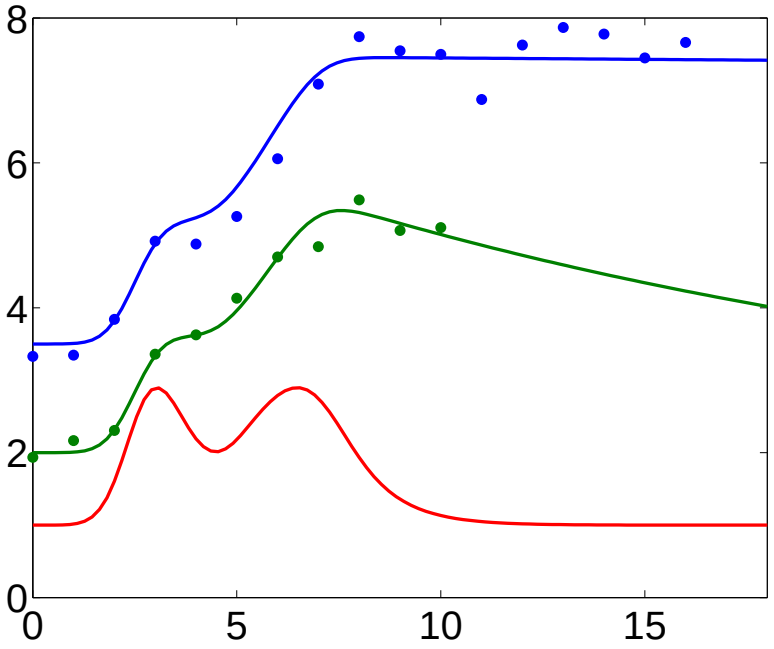
True "gene profiles" and noisy observations.



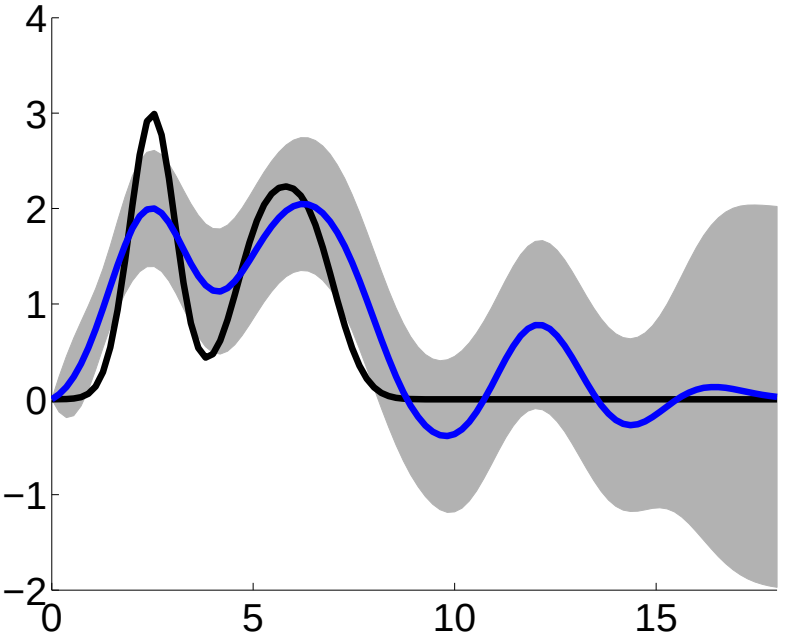
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



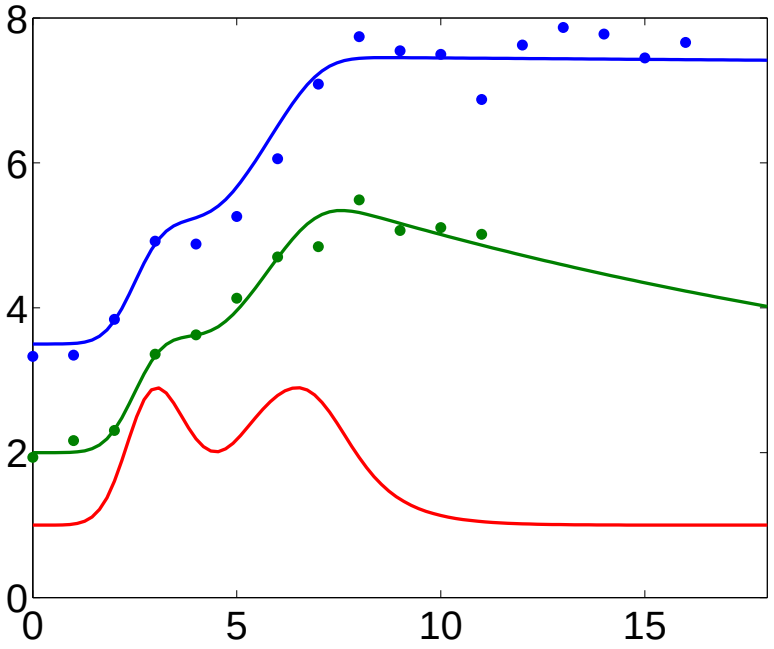
True "gene profiles" and noisy observations.



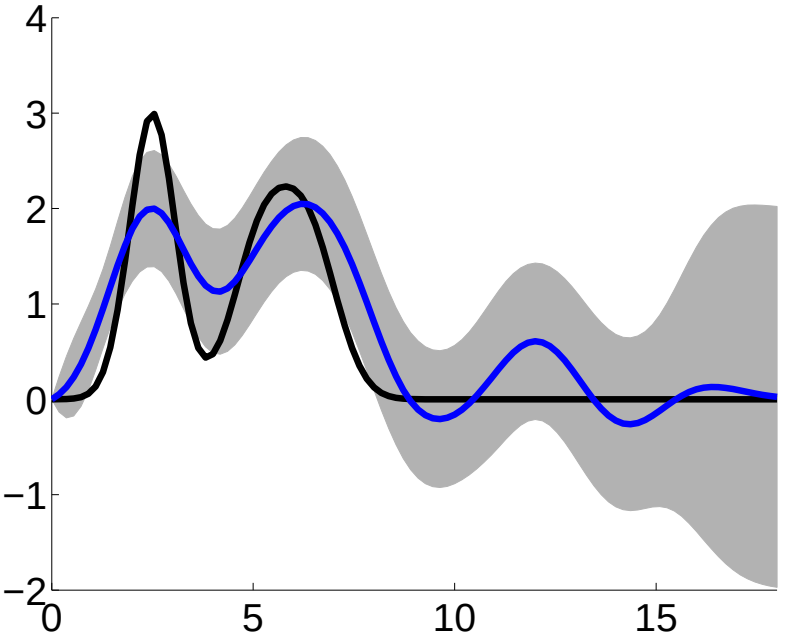
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



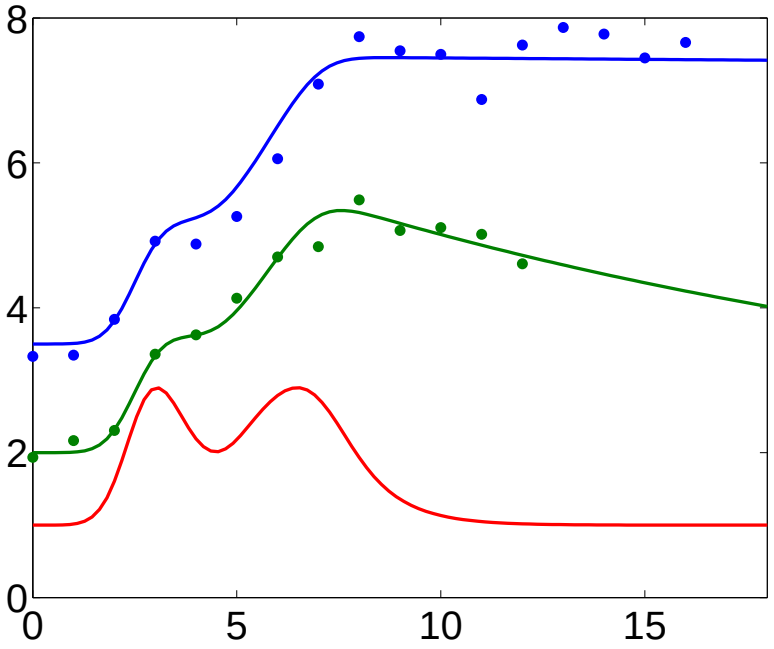
True "gene profiles" and noisy observations.



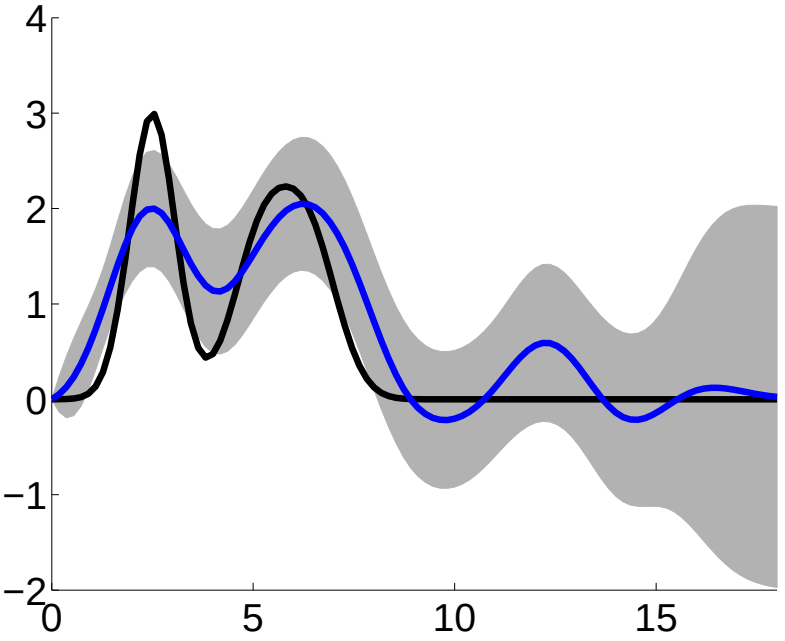
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



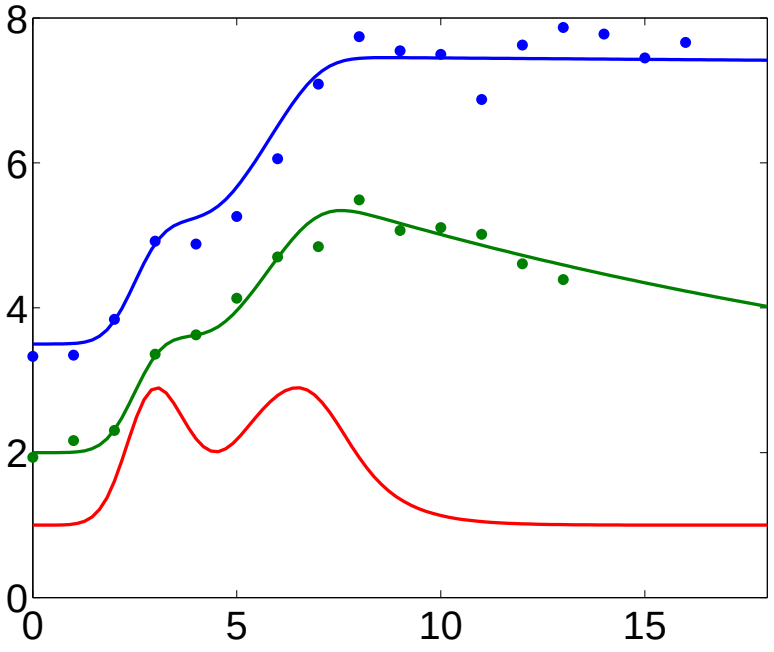
True "gene profiles" and noisy observations.



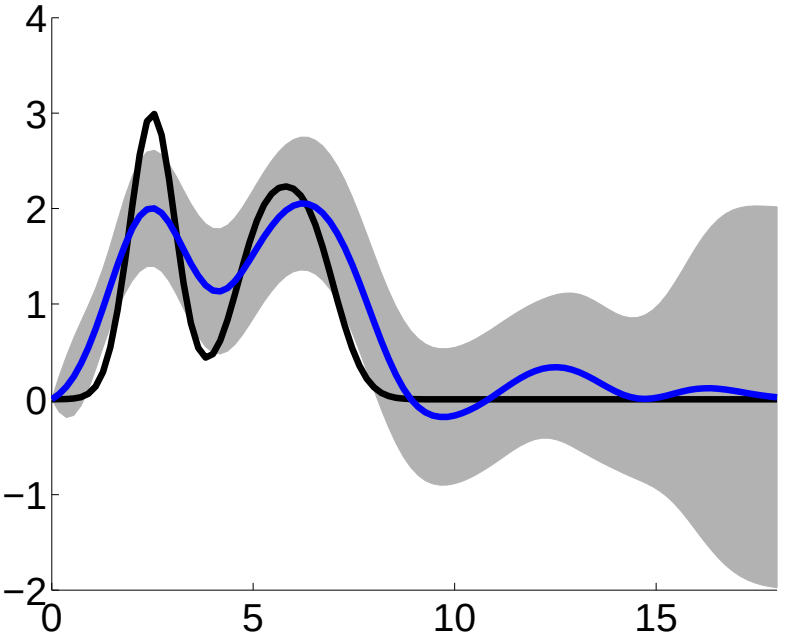
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



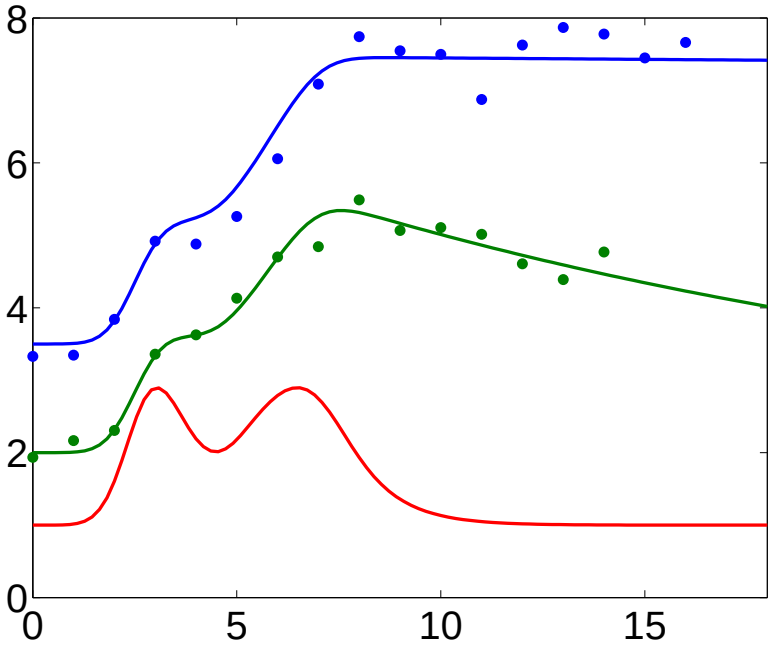
True "gene profiles" and noisy observations.



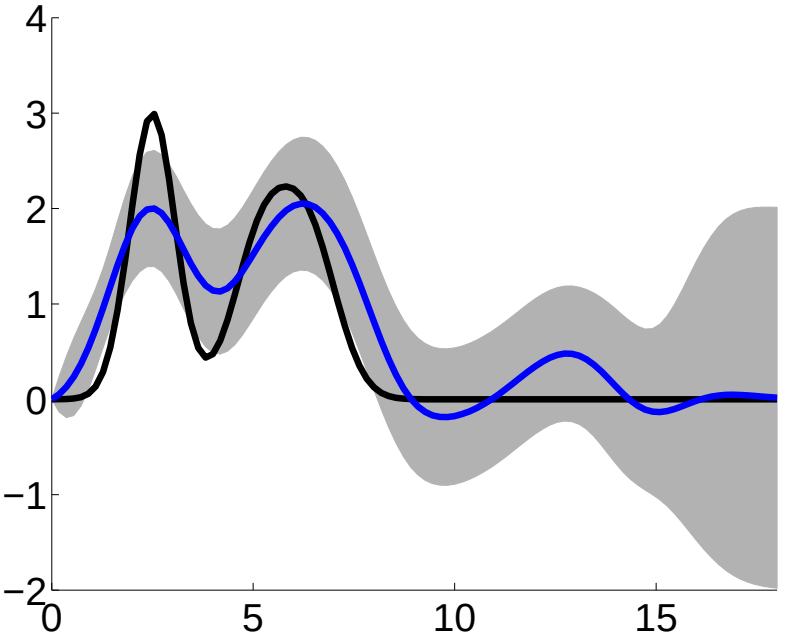
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



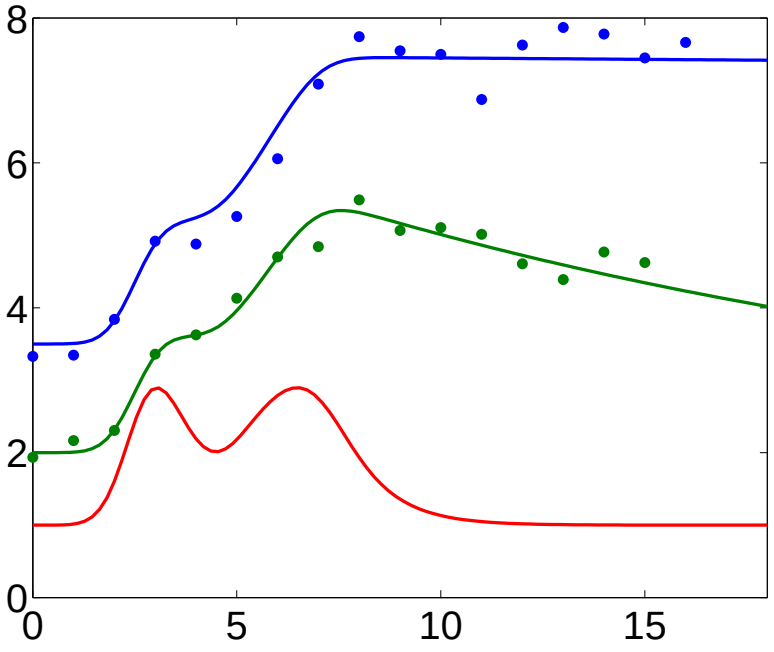
True "gene profiles" and noisy observations.



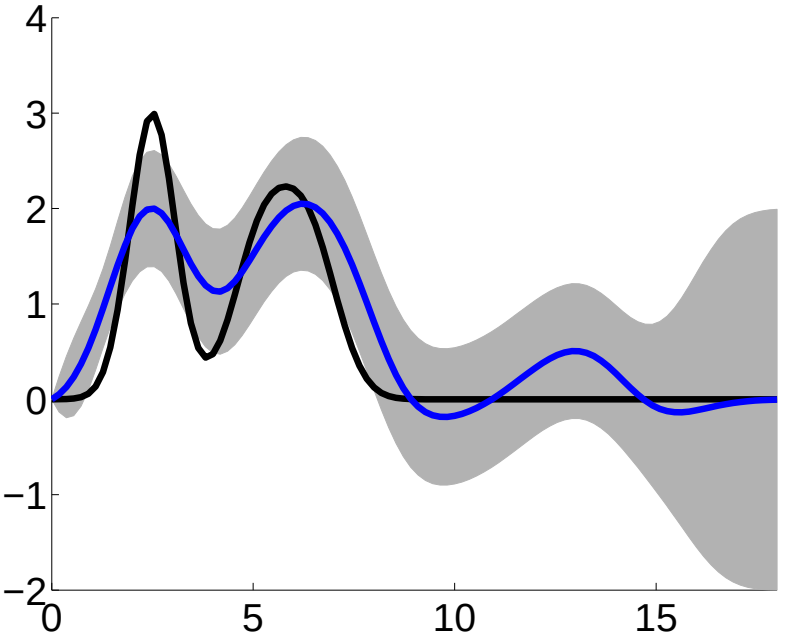
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



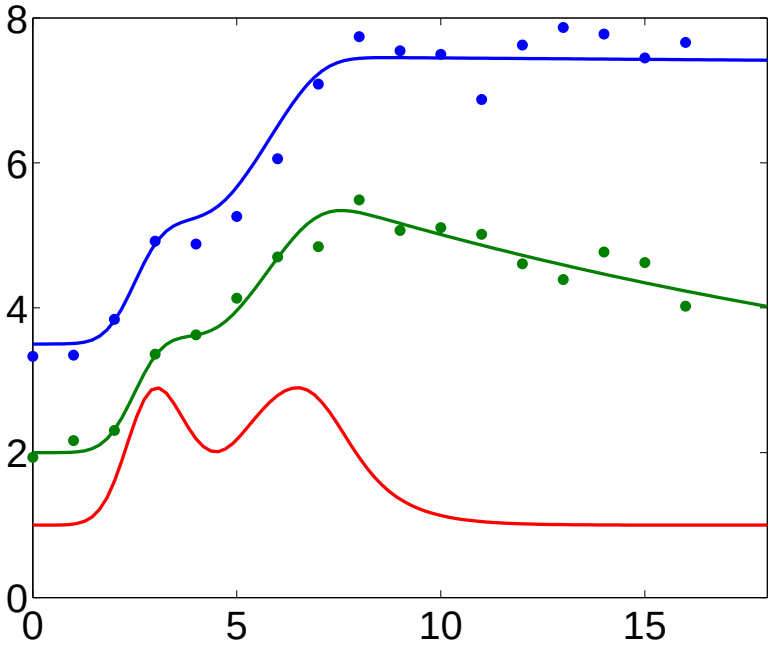
True "gene profiles" and noisy observations.



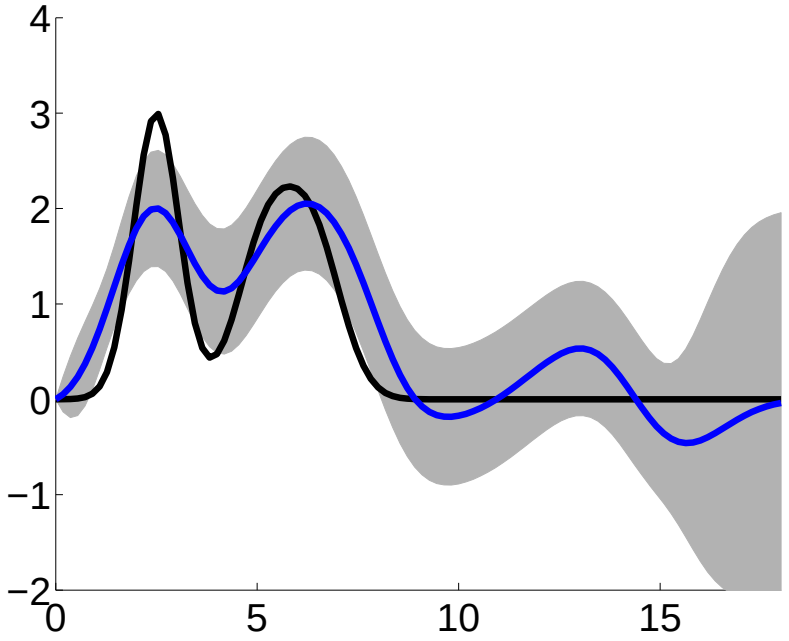
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



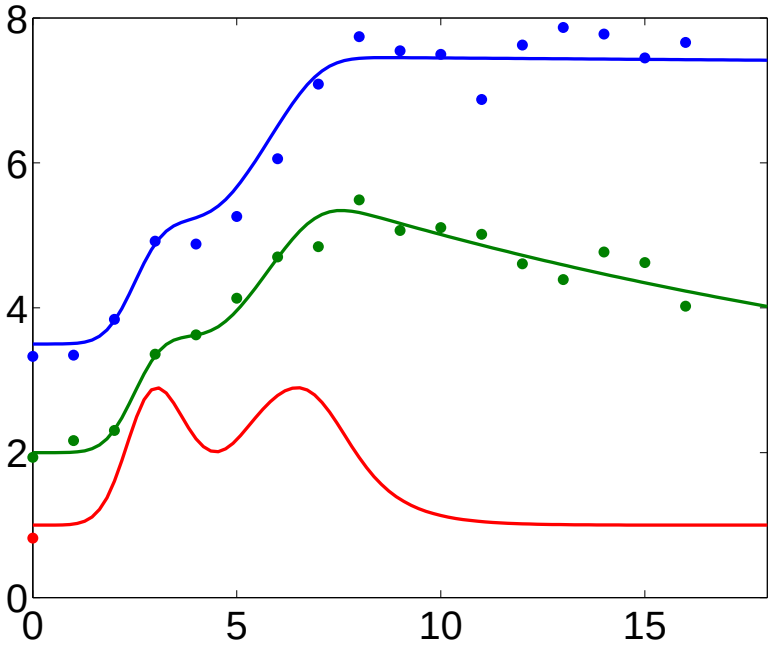
True "gene profiles" and noisy observations.



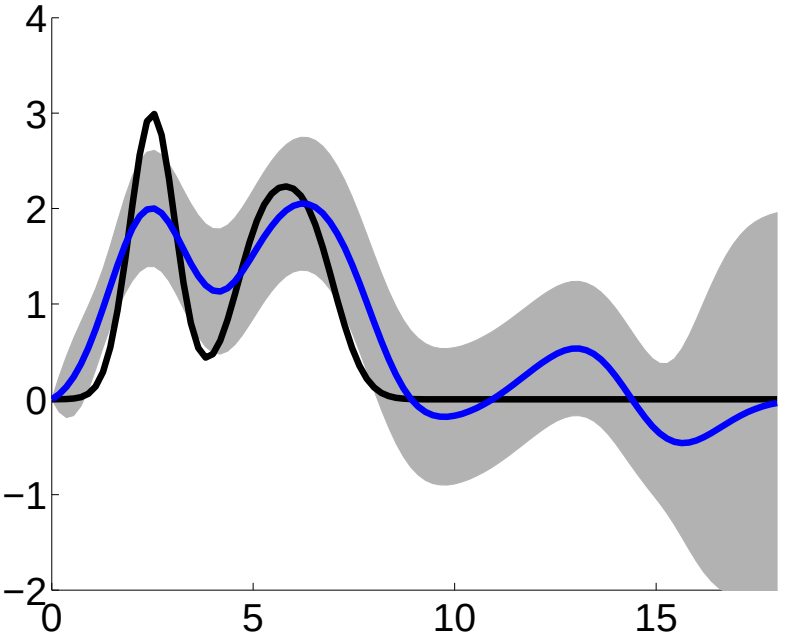
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



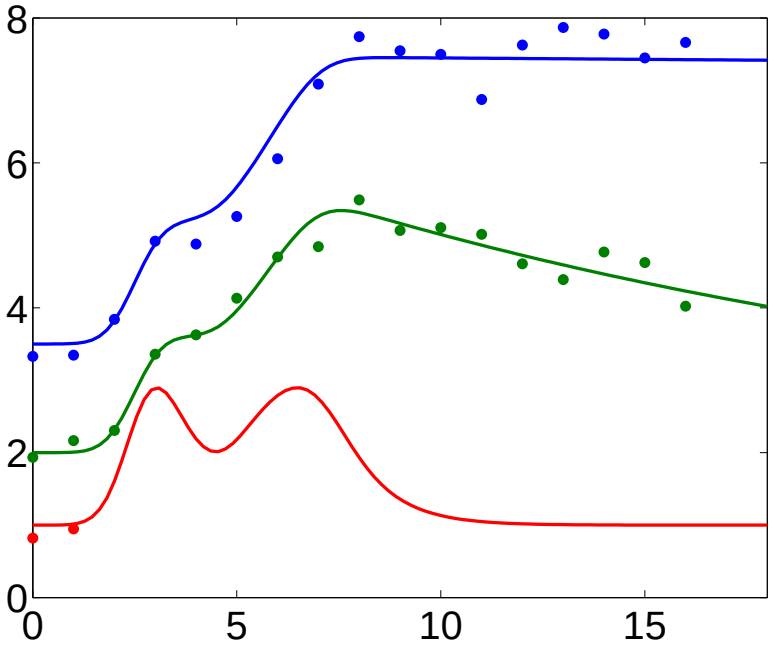
True "gene profiles" and noisy observations.



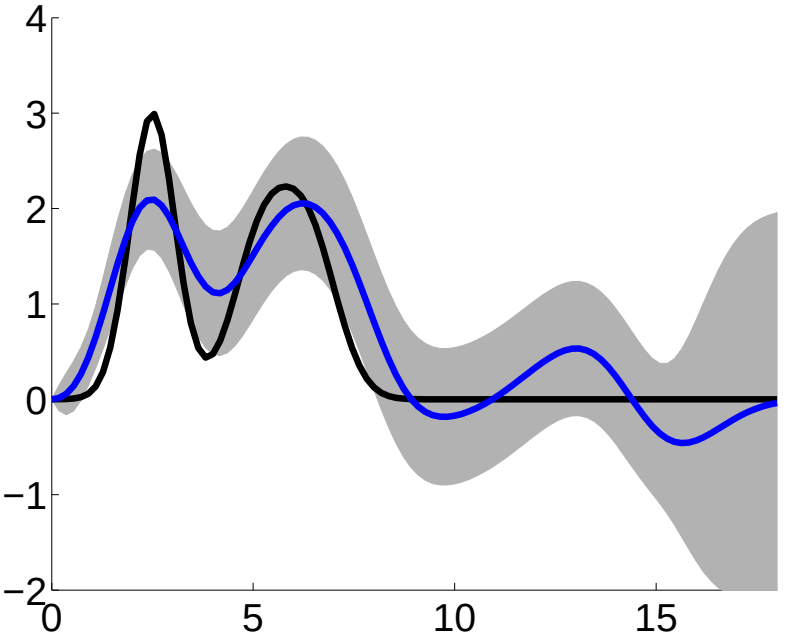
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



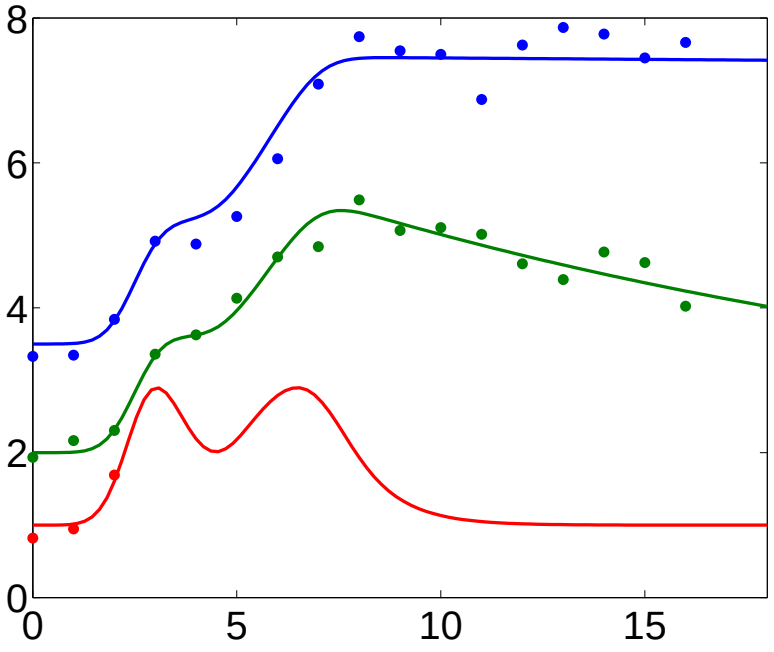
True "gene profiles" and noisy observations.



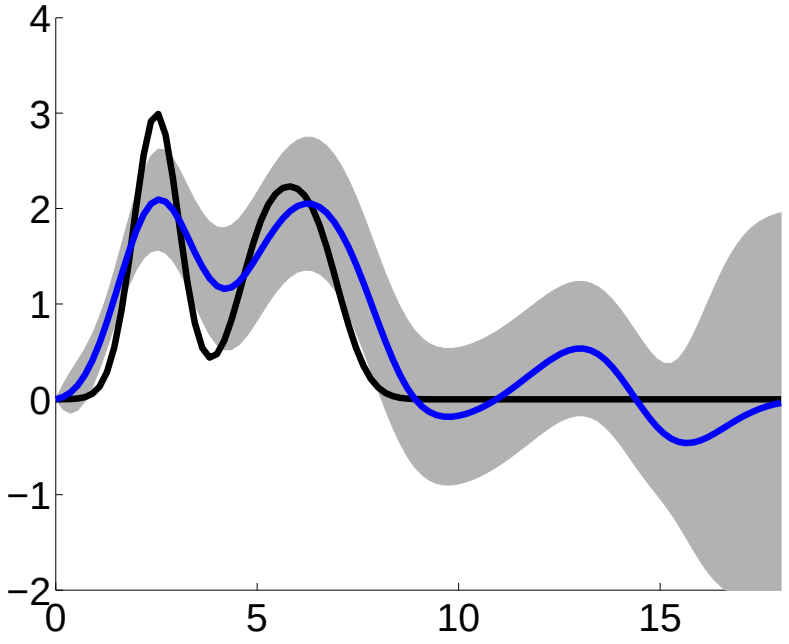
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



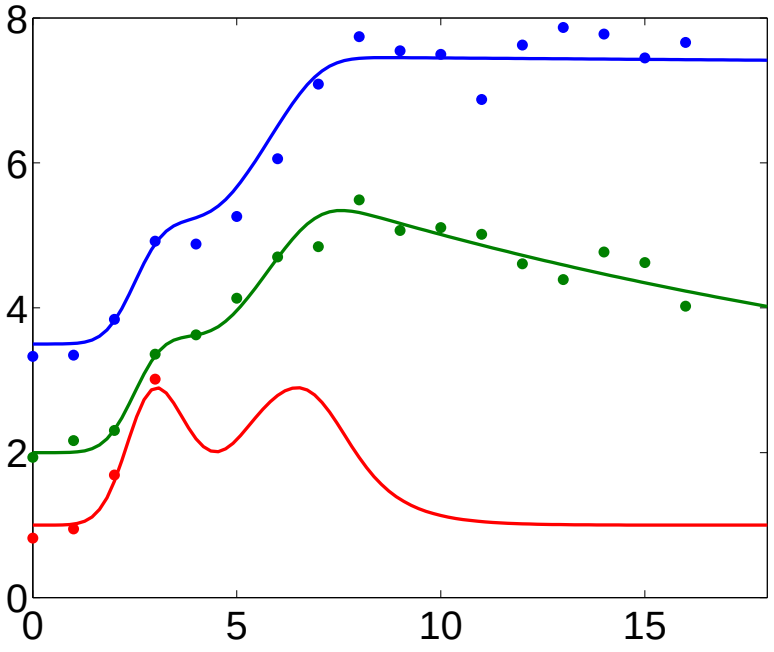
True "gene profiles" and noisy observations.



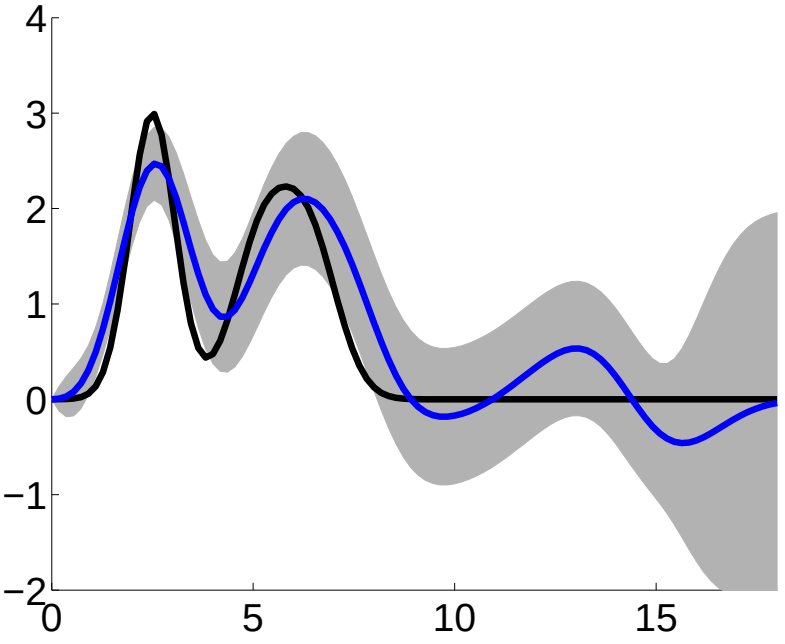
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



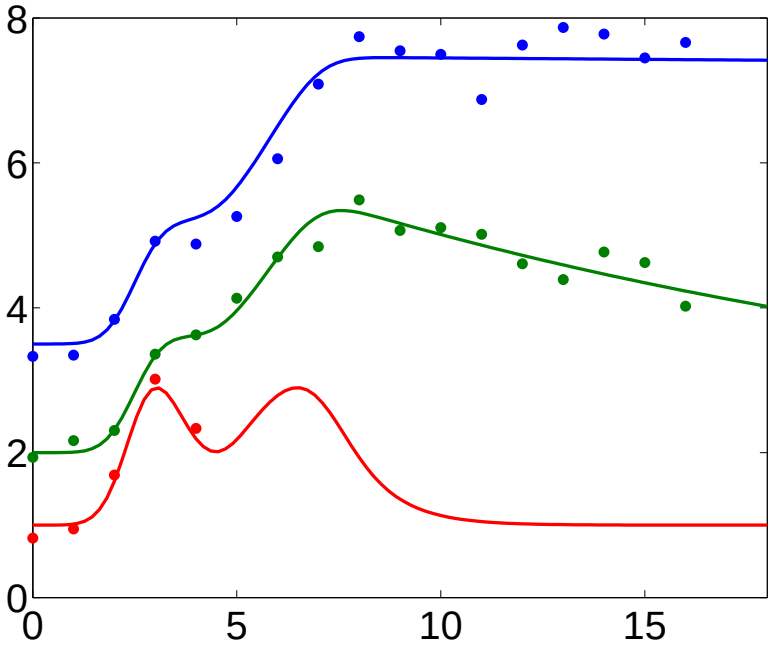
True "gene profiles" and noisy observations.



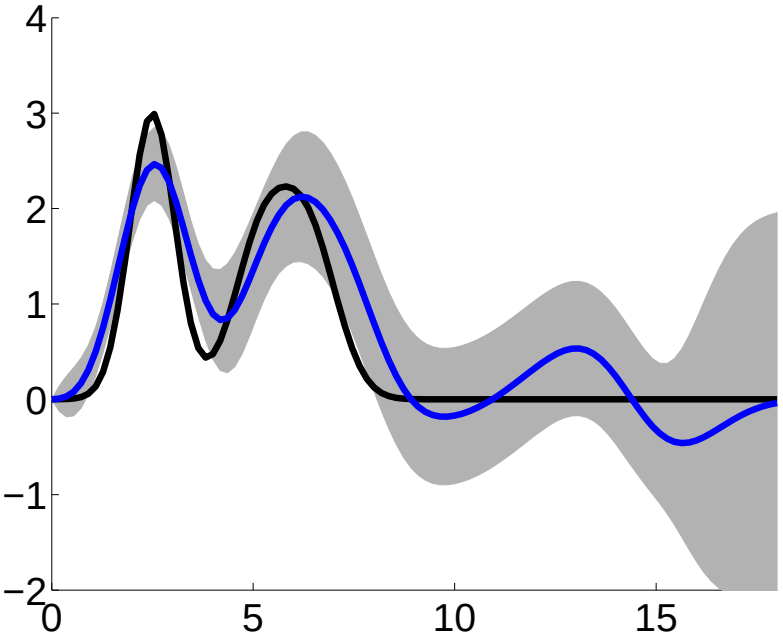
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



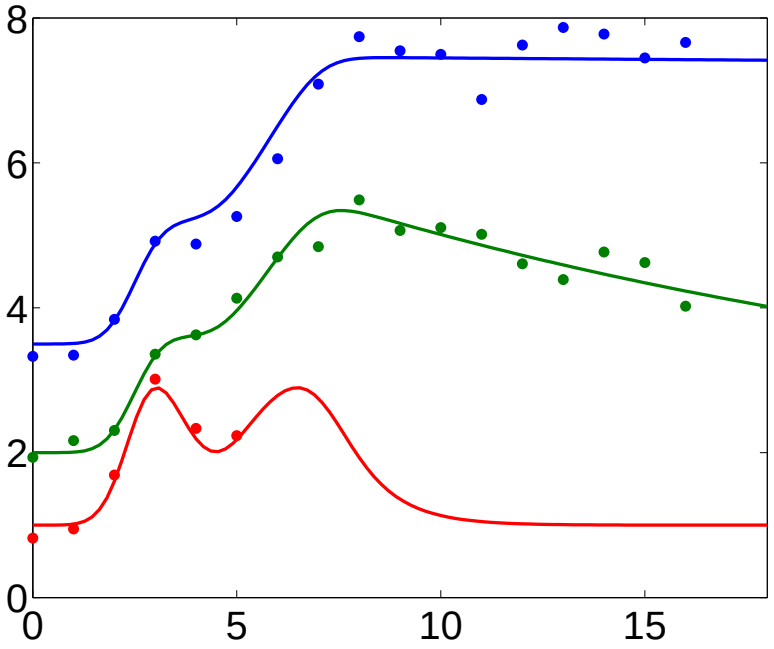
True "gene profiles" and noisy observations.



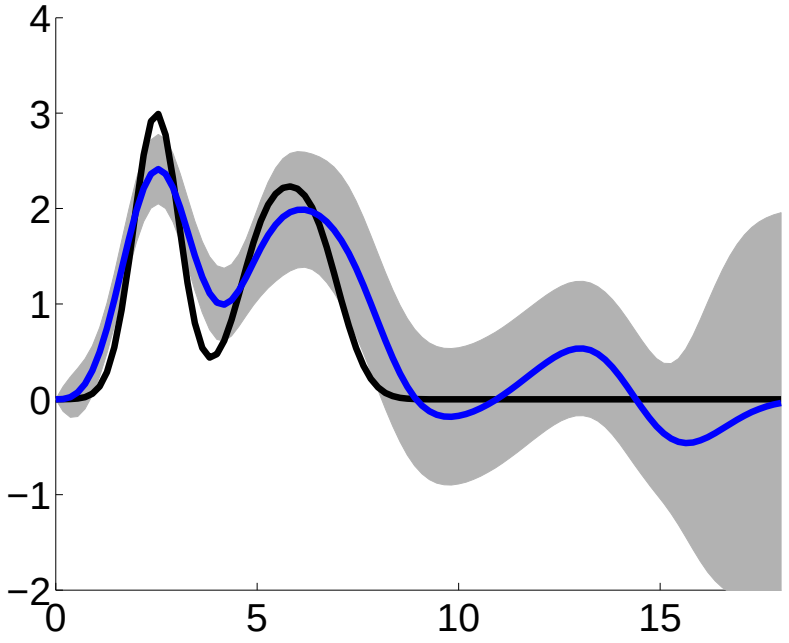
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



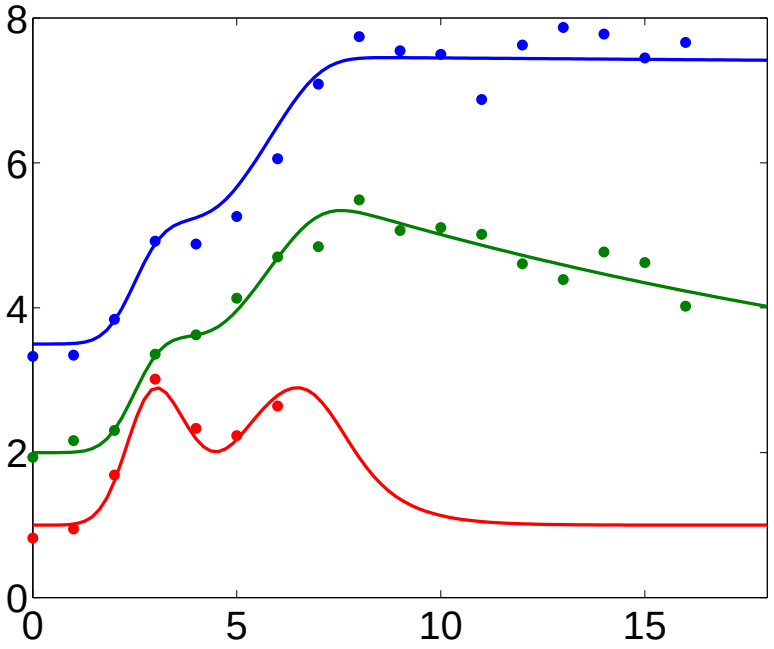
True "gene profiles" and noisy observations.



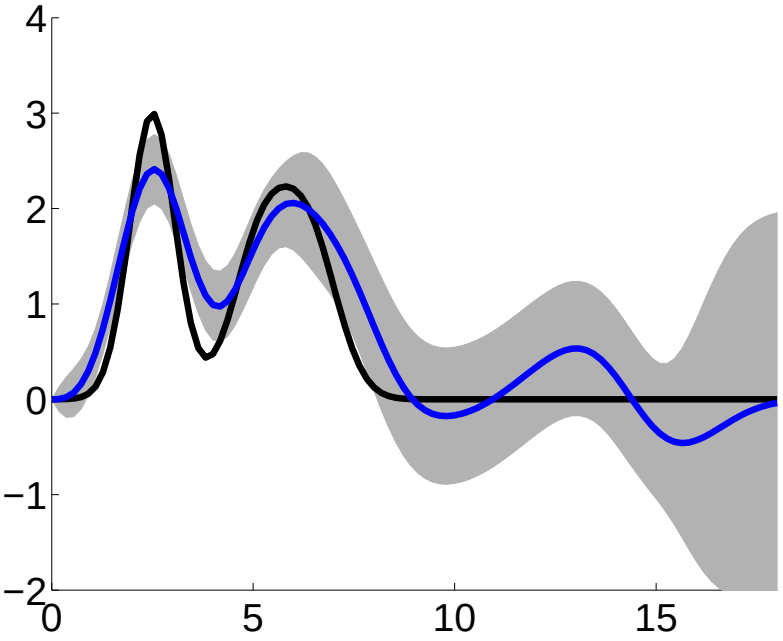
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



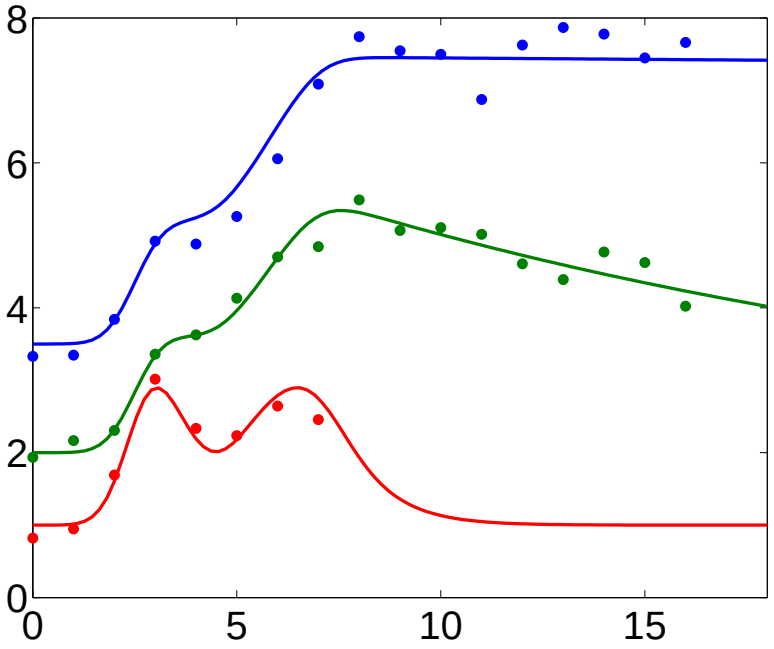
True "gene profiles" and noisy observations.



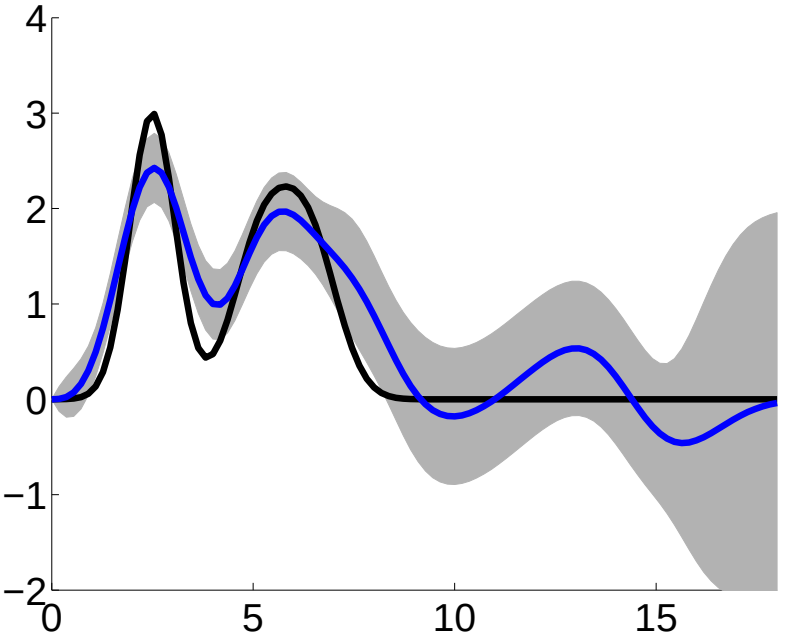
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



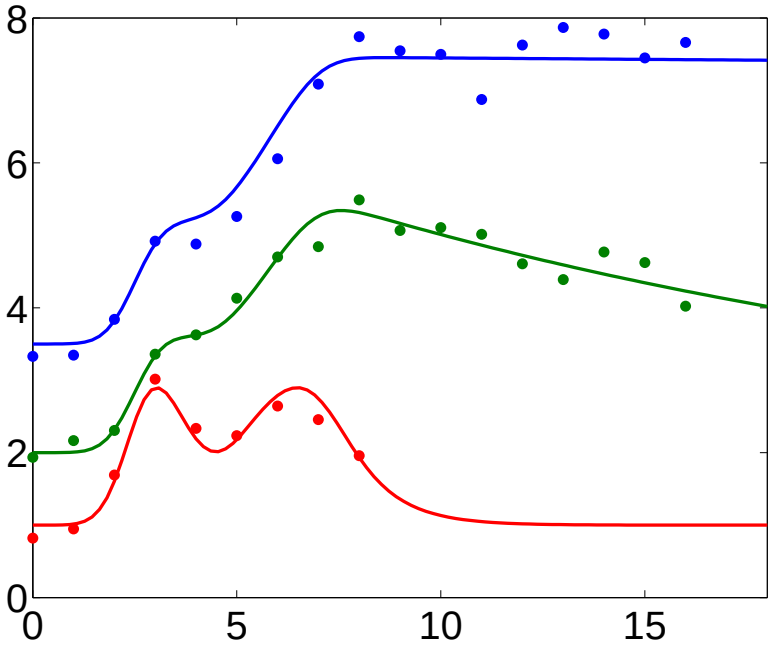
True "gene profiles" and noisy observations.



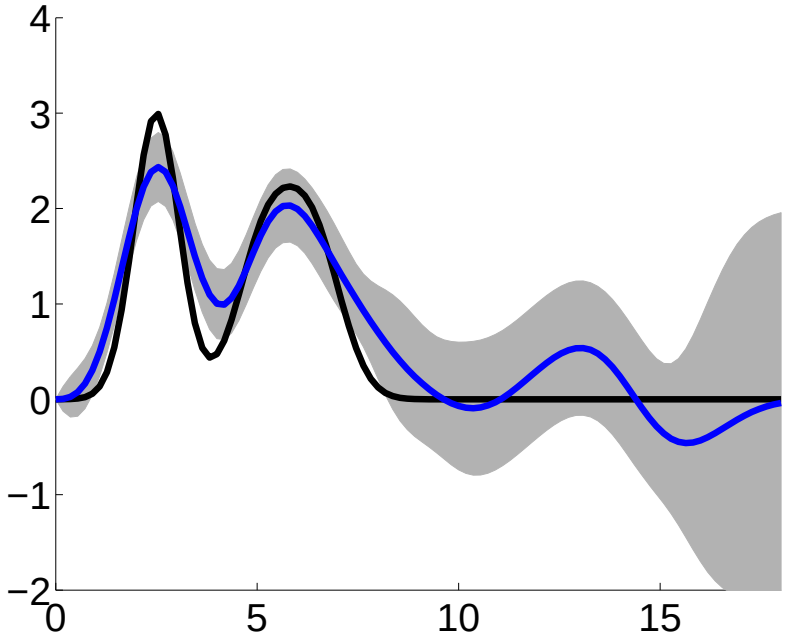
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



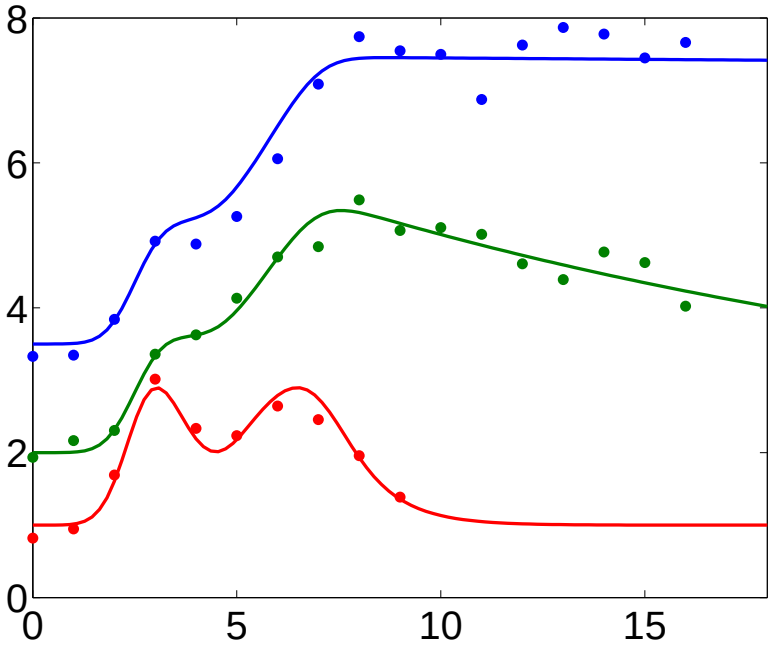
True "gene profiles" and noisy observations.



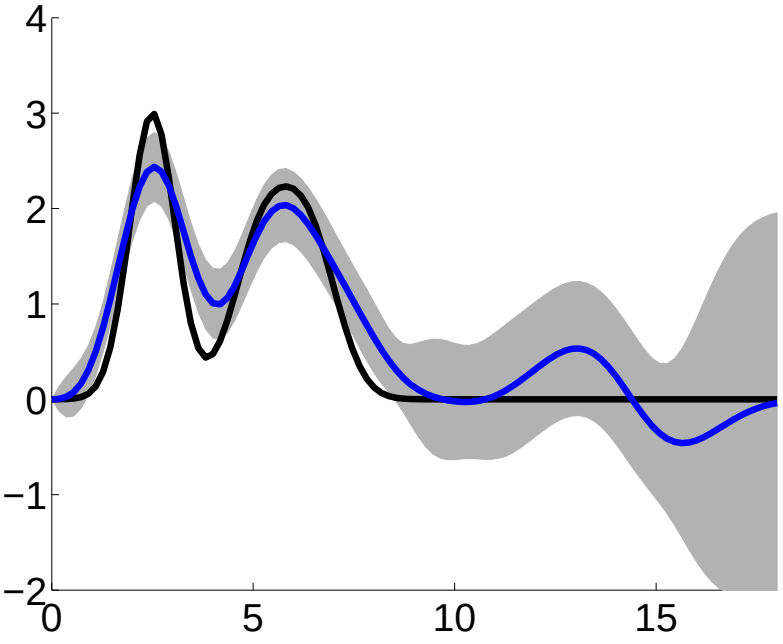
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



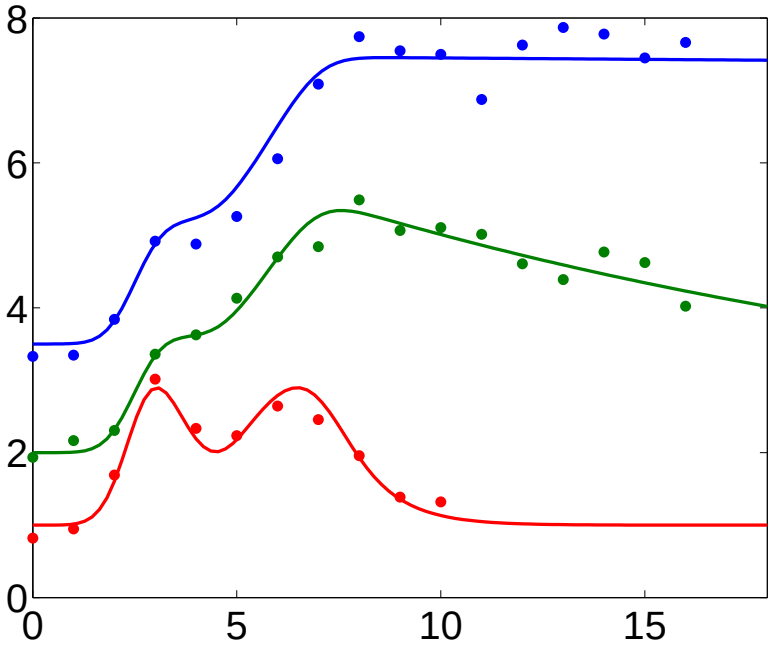
True "gene profiles" and noisy observations.



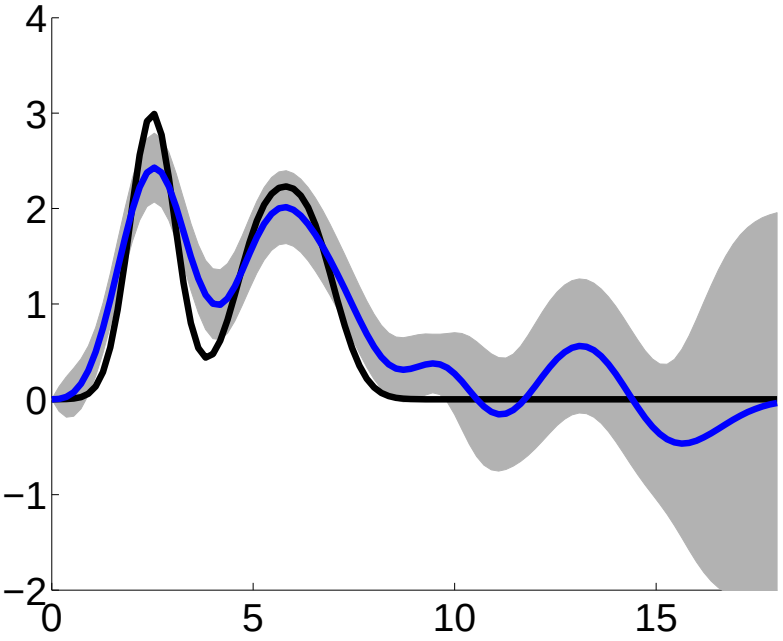
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



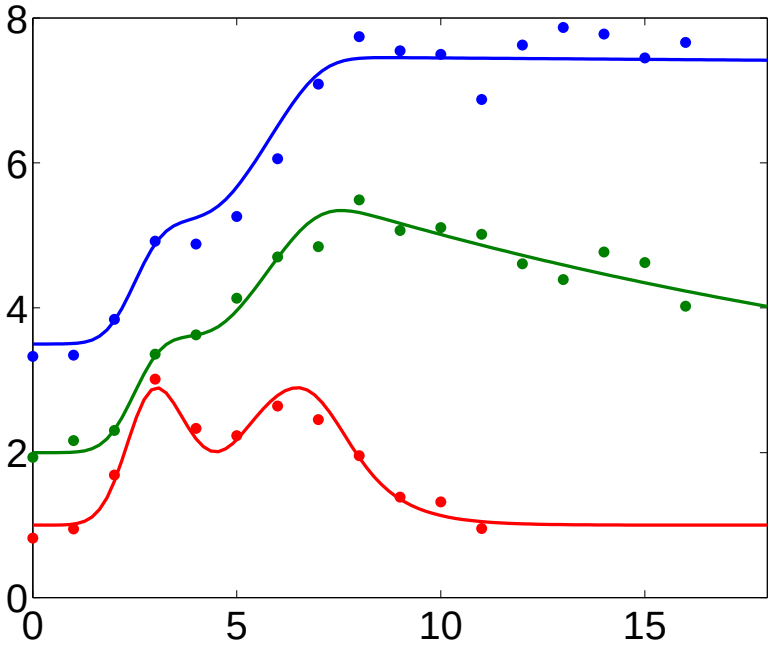
True "gene profiles" and noisy observations.



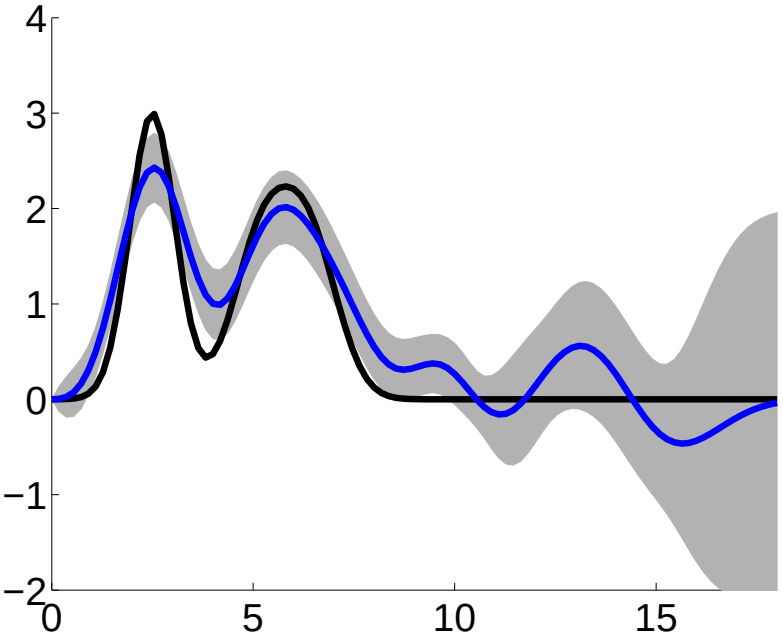
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



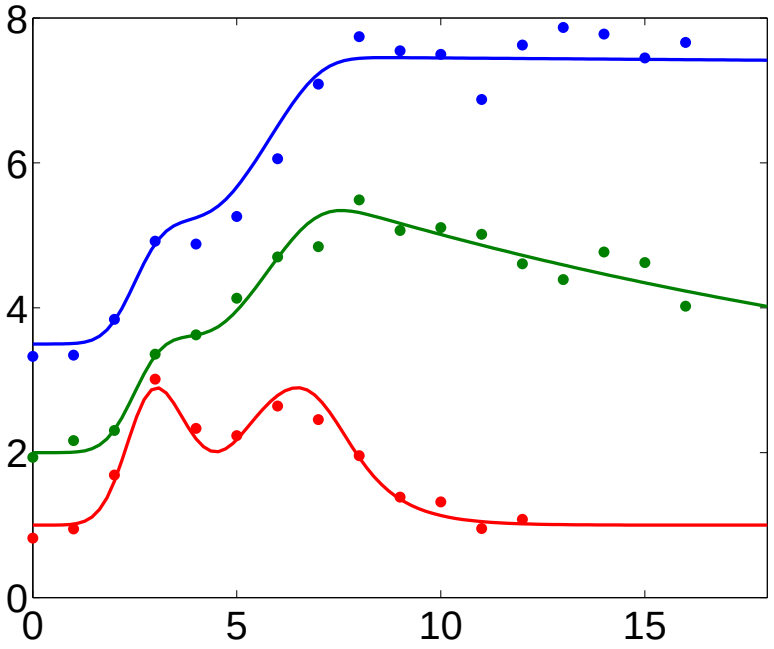
True "gene profiles" and noisy observations.



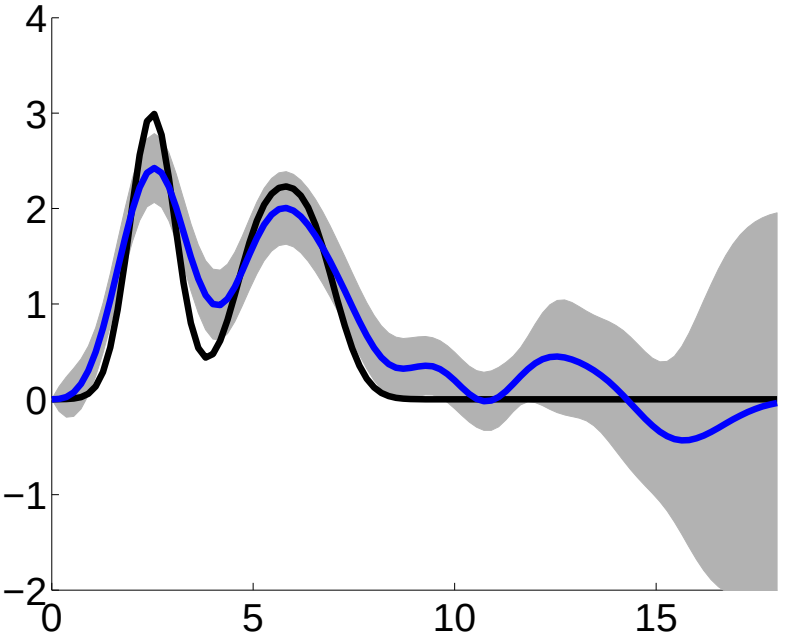
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



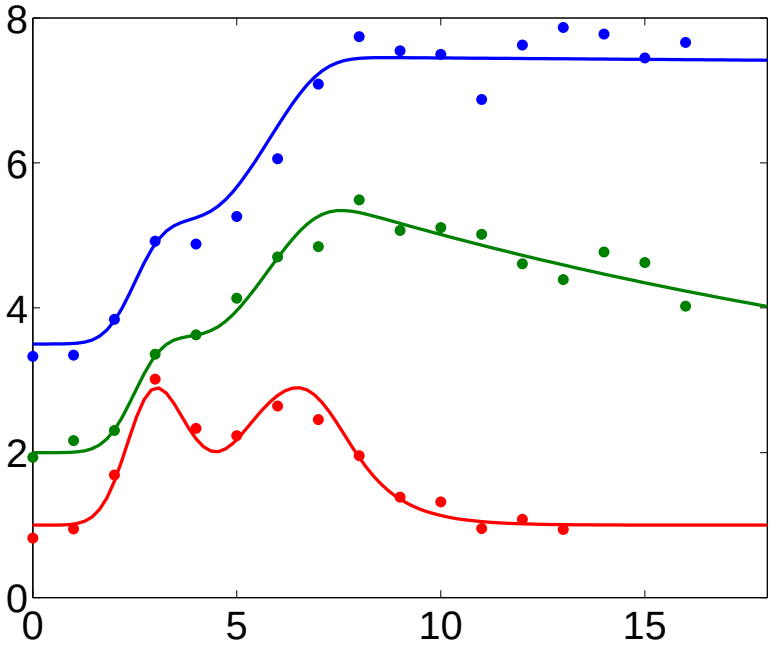
True "gene profiles" and noisy observations.



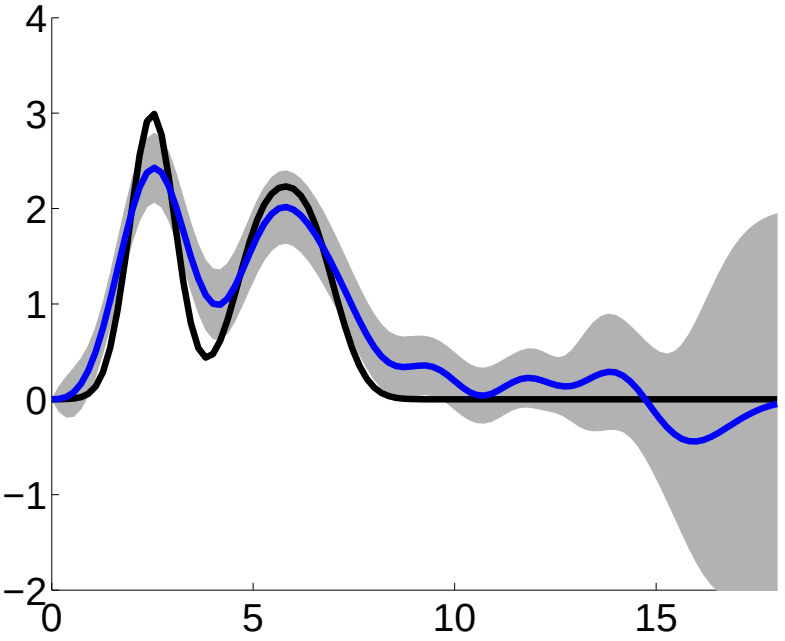
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



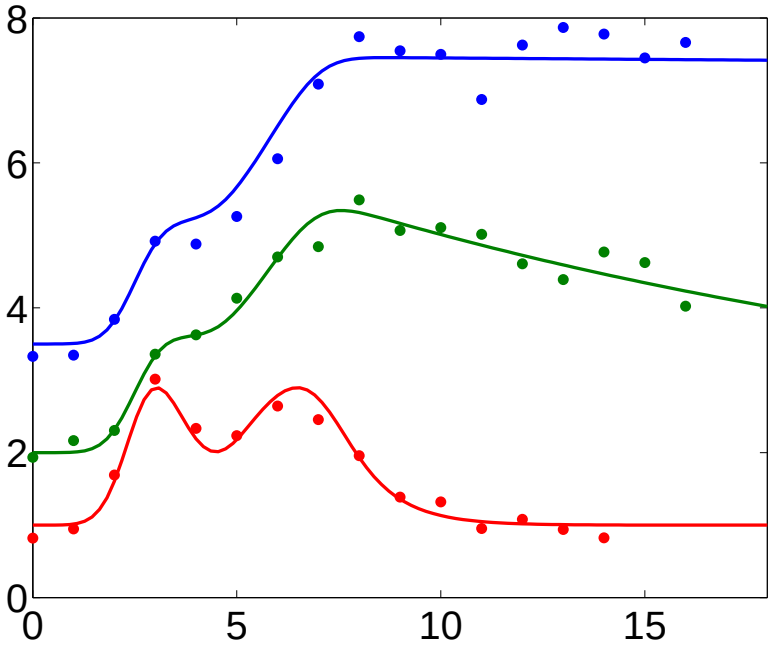
True "gene profiles" and noisy observations.



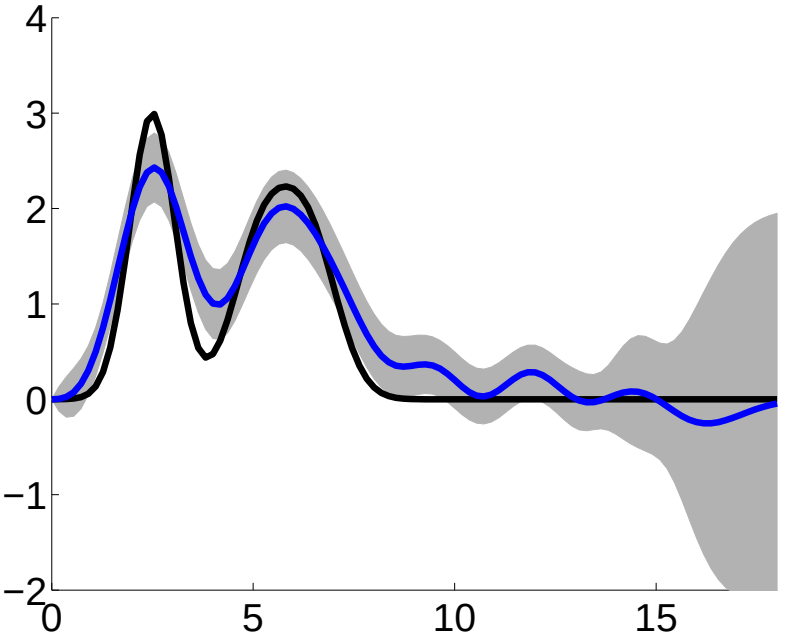
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



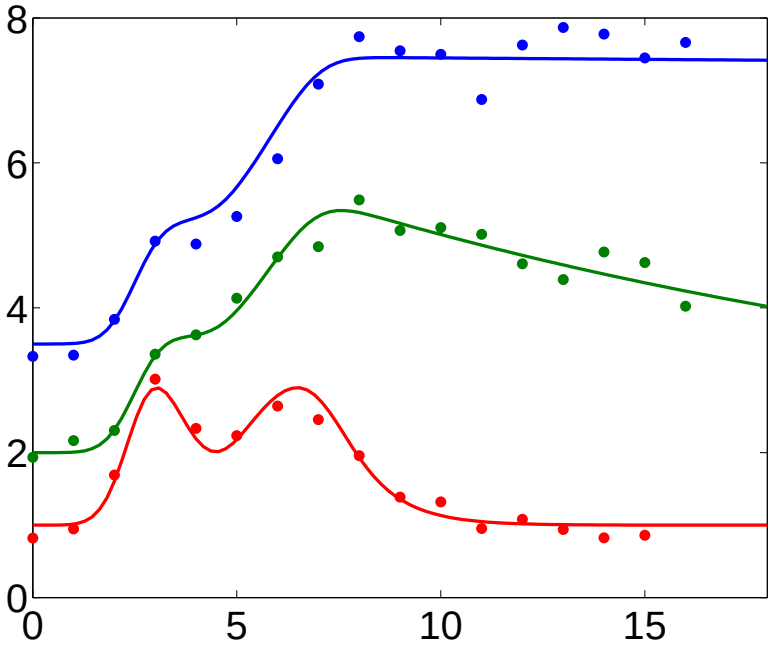
True "gene profiles" and noisy observations.



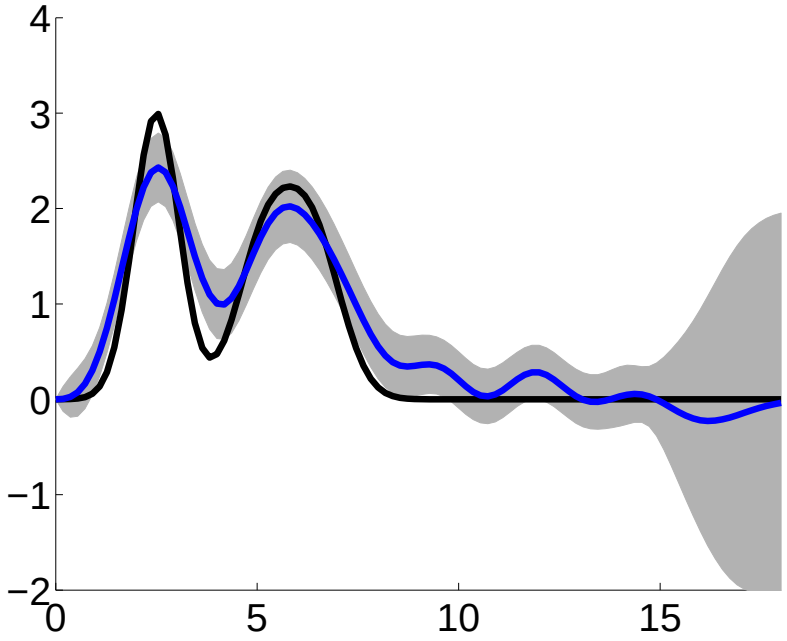
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



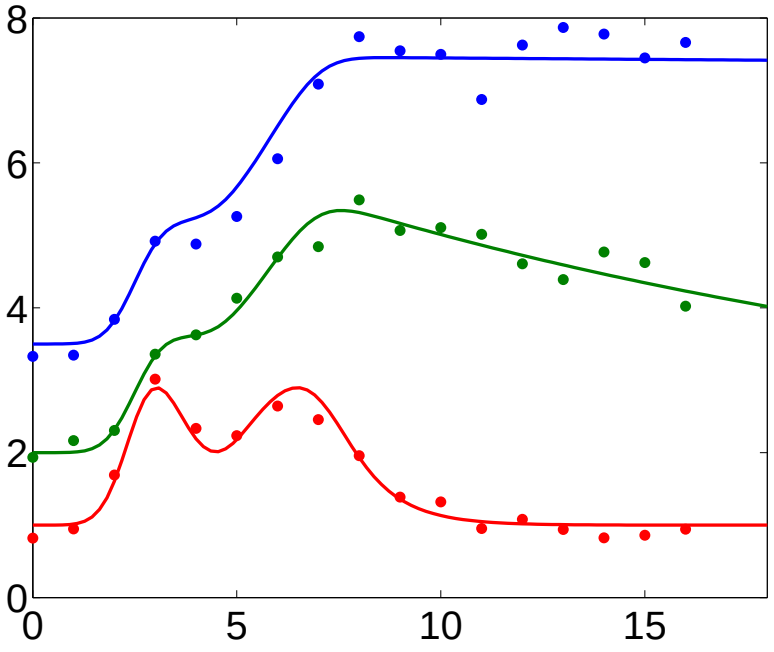
True "gene profiles" and noisy observations.



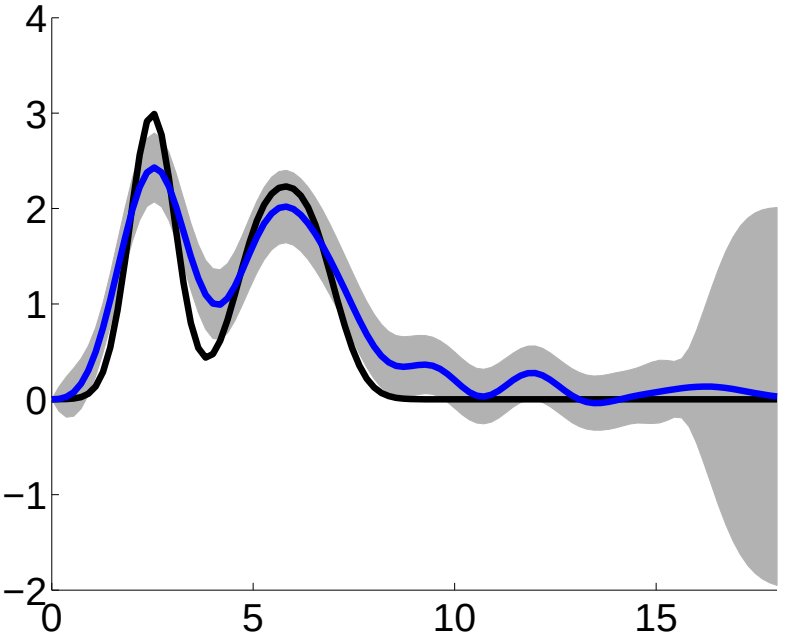
Inferred transcription factor activity.

Artificial Example: Inferring $f(t)$

Inferring TF activity from artificially sampled genes.



True "gene profiles" and noisy observations.



Inferred transcription factor activity.

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

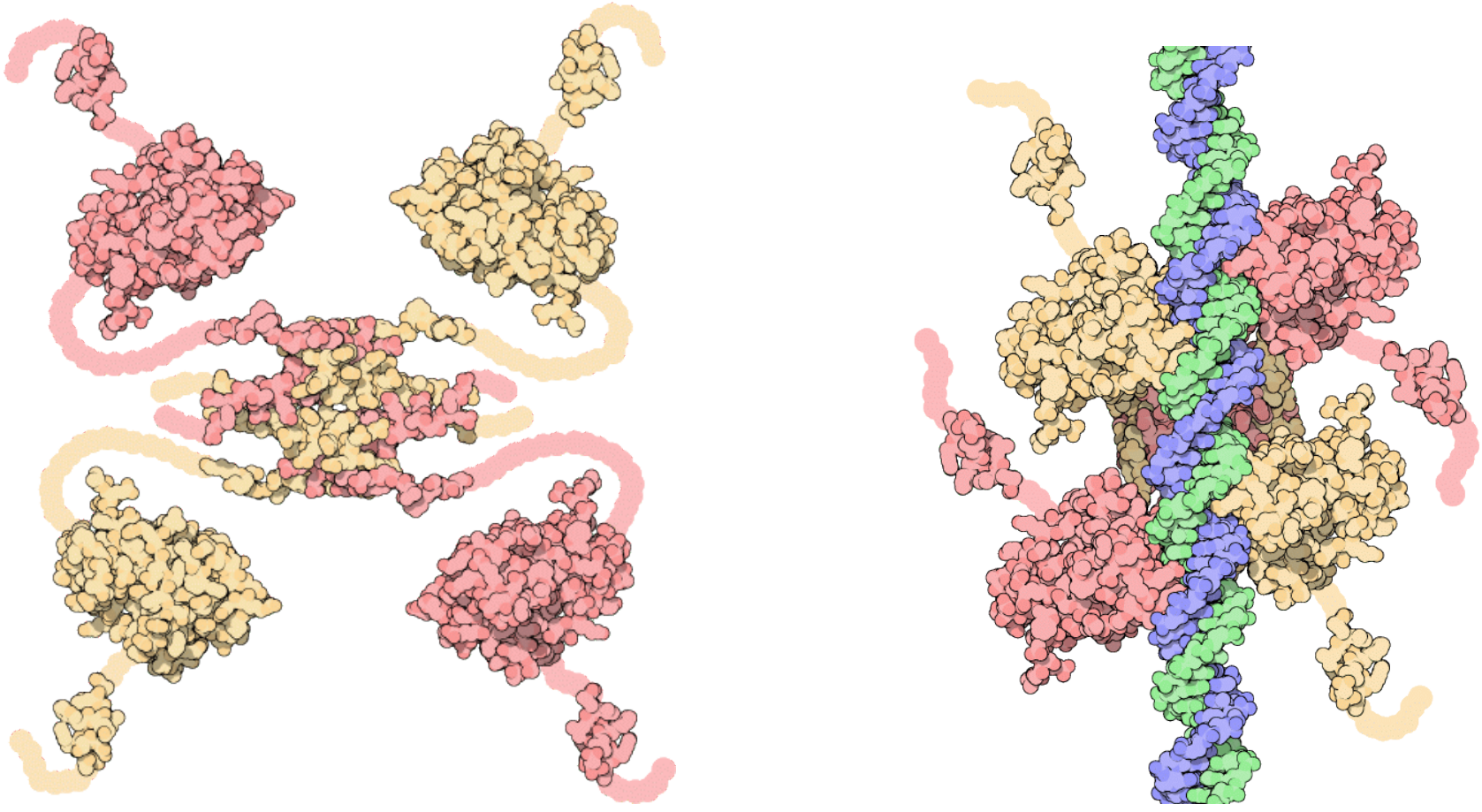


Figure: p53. *Left* unbound, *Right* bound to DNA. Images by David S. Goodsell from `http://www.rcsb.org/` (see the"Molecule of the Month" feature).

p53

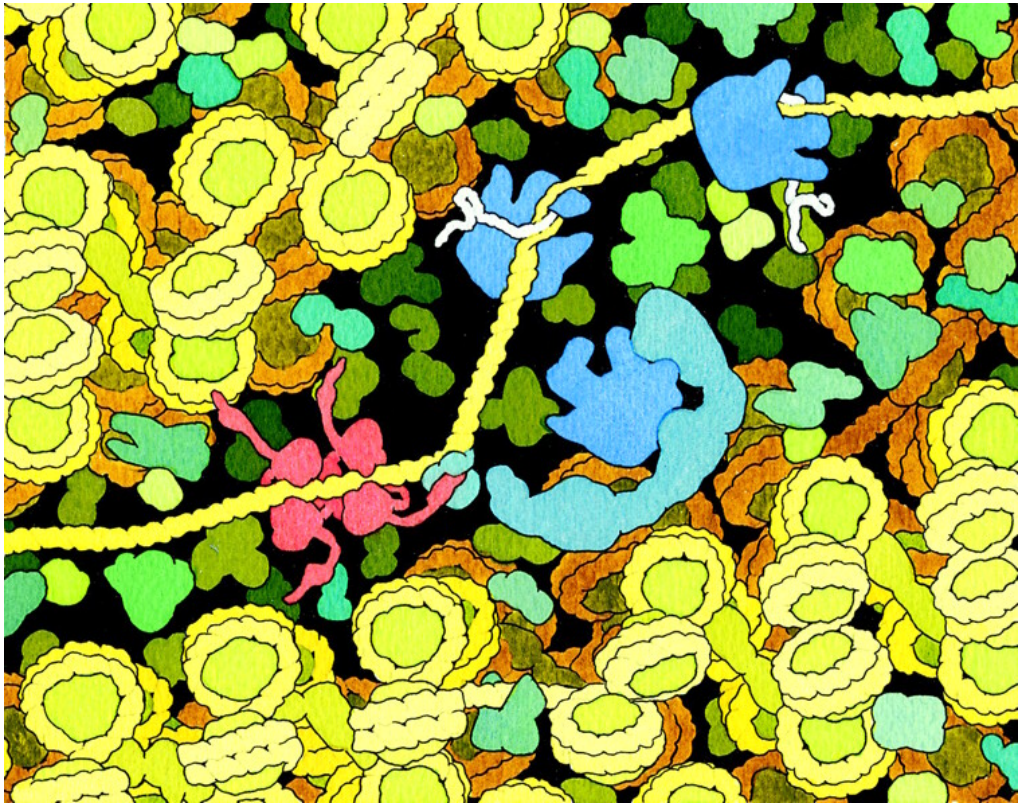


Figure: Repair of DNA damage by p53. Image from Goodsell (1999).

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 governed by SREBP-1a, Sp1, Sp3,...).

hPA26/SESN1 sestrin 1 Cell Cycle arrest.

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

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

Modelling Assumption

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

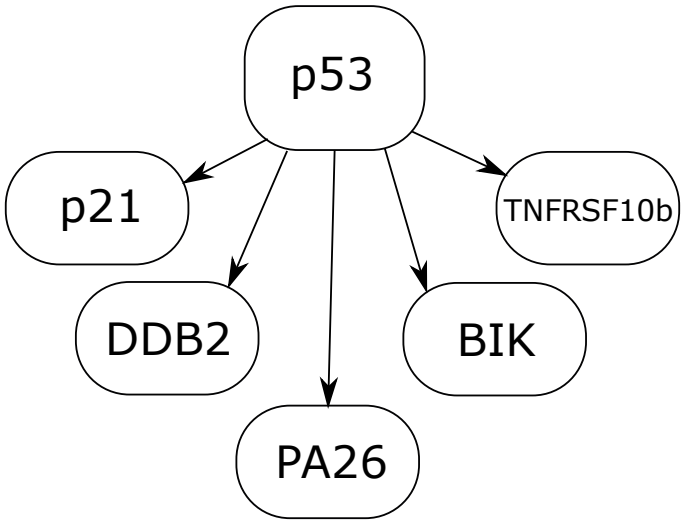


Figure: p53 SIM network motif as modelled by Barenco et al. 2006.

Ordinary Differential Equation Model

- First Order Differential Equation

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

- Proposed by Barenco et al. (2006).
- $x_j(t)$ – concentration of gene j's mRNA
- $f(t)$ – concentration of active transcription factor
- Model parameters: baseline b_j, sensitivity s_j and decay d_j
- Application: identifying co-regulated genes (targets)
- Problem: how do we fit the model when $f(t)$ is not observed?

Ordinary Differential Equation Model

- First Order Differential Equation

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

- Proposed by Barenco et al. (2006).
- $x_j(t)$ – concentration of gene j's mRNA
- $f(t)$ – concentration of active transcription factor
- Model parameters: baseline b_j, sensitivity s_j and decay d_j
- Application: identifying co-regulated genes (targets)
- Problem: how do we fit the model when $f(t)$ is not observed?

Ordinary Differential Equation Model

- First Order Differential Equation

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

- Proposed by Barenco et al. (2006).
- $x_j(t)$ – concentration of gene j's mRNA
- $f(t)$ – concentration of active transcription factor
- Model parameters: baseline b_j, sensitivity s_j and decay d_j
- Application: identifying co-regulated genes (targets)
- Problem: how do we fit the model when $f(t)$ is not observed?

Ordinary Differential Equation Model

- First Order Differential Equation

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

- Proposed by Barenco et al. (2006).
- $x_j(t)$ – concentration of gene j's mRNA
- $f(t)$ – concentration of active transcription factor
- Model parameters: baseline b_j, sensitivity s_j and decay d_j
- Application: identifying co-regulated genes (targets)
- Problem: how do we fit the model when $f(t)$ is not observed?

Ordinary Differential Equation Model

- First Order Differential Equation

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

- Proposed by Barenco et al. (2006).
- $x_j(t)$ – concentration of gene j's mRNA
- $f(t)$ – concentration of active transcription factor
- Model parameters: baseline b_j, sensitivity s_j and decay d_j
- Application: identifying co-regulated genes (targets)
- Problem: how do we fit the model when $f(t)$ is not observed?

Ordinary Differential Equation Model

- First Order Differential Equation

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

- Proposed by Barenco et al. (2006).
- $x_j(t)$ – concentration of gene j's mRNA
- $f(t)$ – concentration of active transcription factor
- Model parameters: baseline b_j, sensitivity s_j and decay d_j
- Application: identifying co-regulated genes (targets)
- Problem: how do we fit the model when $f(t)$ is not observed?

Ordinary Differential Equation Model

- First Order Differential Equation

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

- Proposed by Barenco et al. (2006).
- $x_j(t)$ – concentration of gene j's mRNA
- $f(t)$ – concentration of active transcription factor
- Model parameters: baseline b_j, sensitivity s_j and decay d_j
- Application: identifying co-regulated genes (targets)
- Problem: how do we fit the model when $f(t)$ is not observed?

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

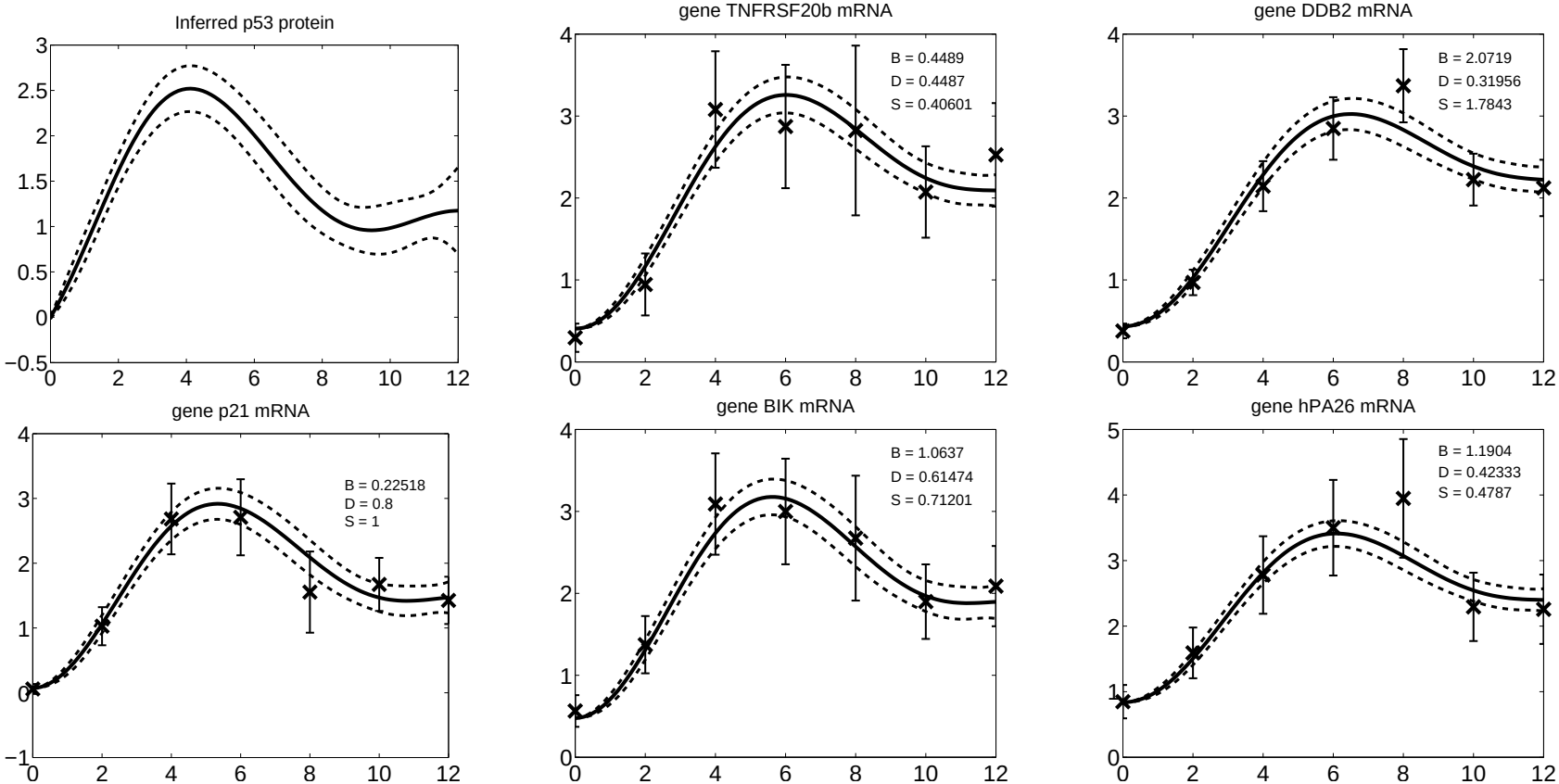
ABSTRACT

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)



Cascaded Differential Equations

PNAS

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

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

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

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

Covariance for Translation/Transcription Model

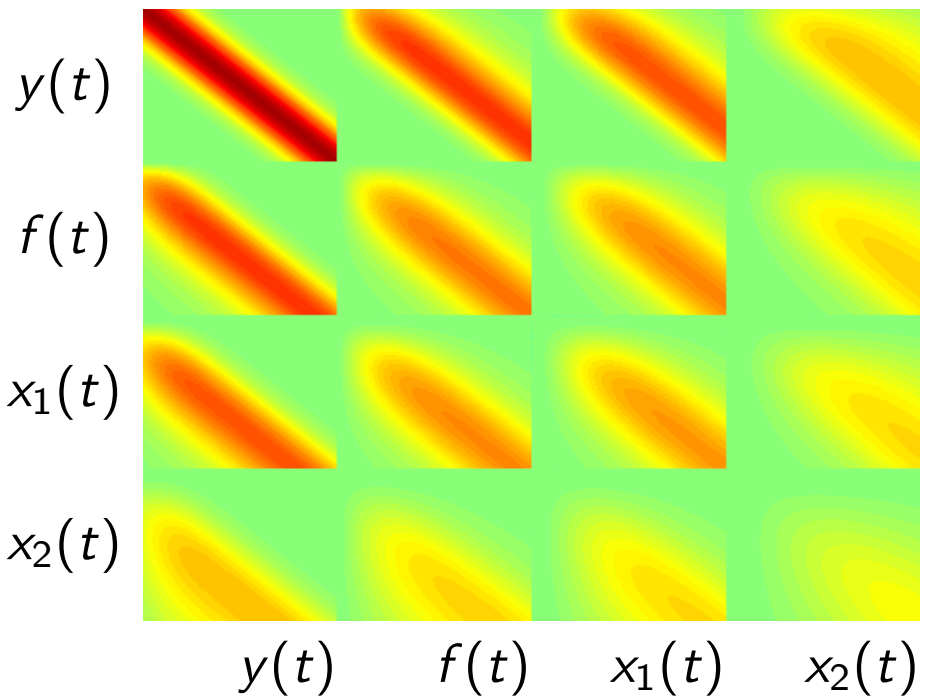
RBF covariance function for $y(t)$

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

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

- 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`

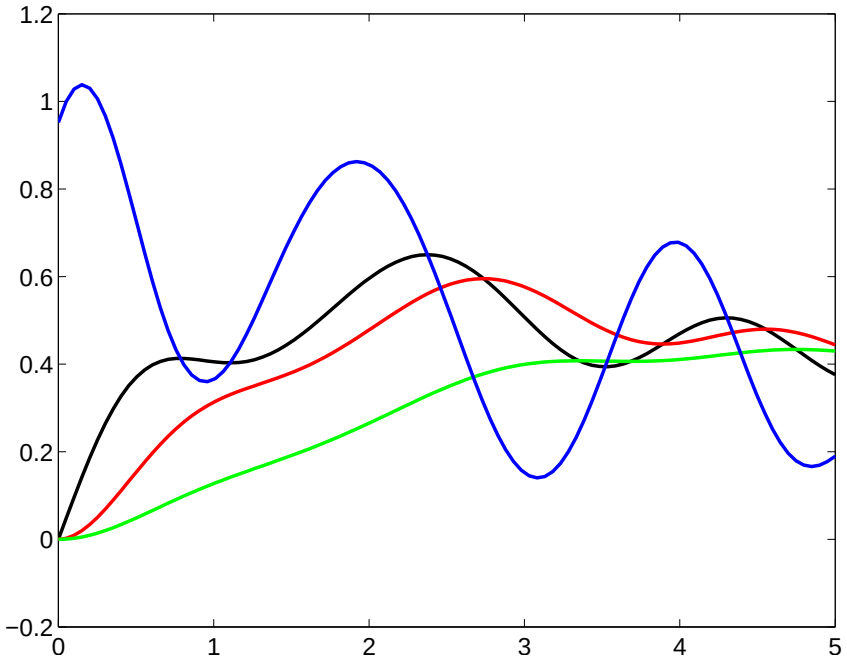


Figure: Joint samples from the ODE covariance, *blue*: $y(t)$ (mRNA of TF), *black*: $f(t)$ (TF concentration), *red*: $x_1(t)$ (high decay target) and *green*: $x_2(t)$ (low decay target)

.

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

- `disimSample`

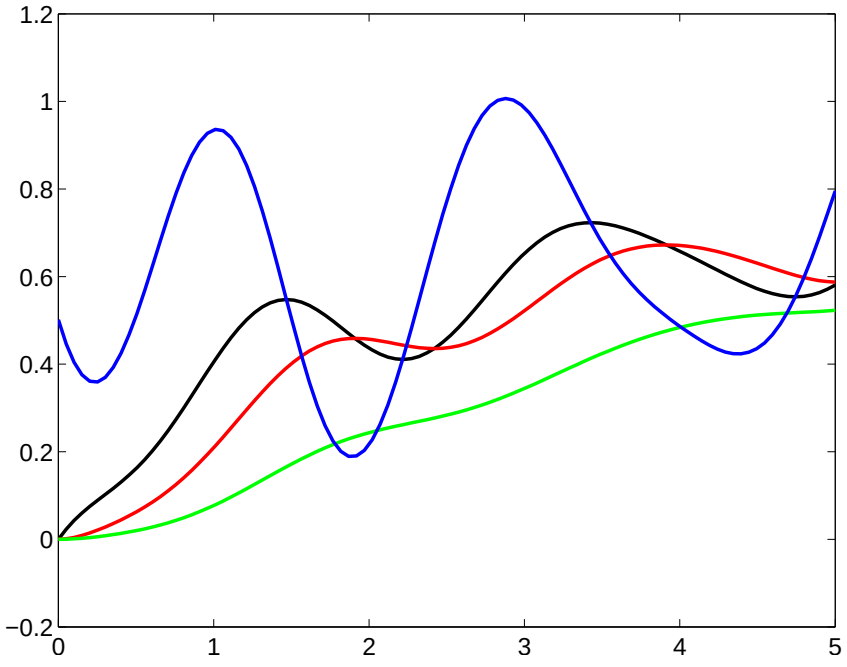


Figure: Joint samples from the ODE covariance, *blue*: $y(t)$ (mRNA of TF), *black*: $f(t)$ (TF concentration), *red*: $x_1(t)$ (high decay target) and *green*: $x_2(t)$ (low decay target)

.

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

- `disimSample`

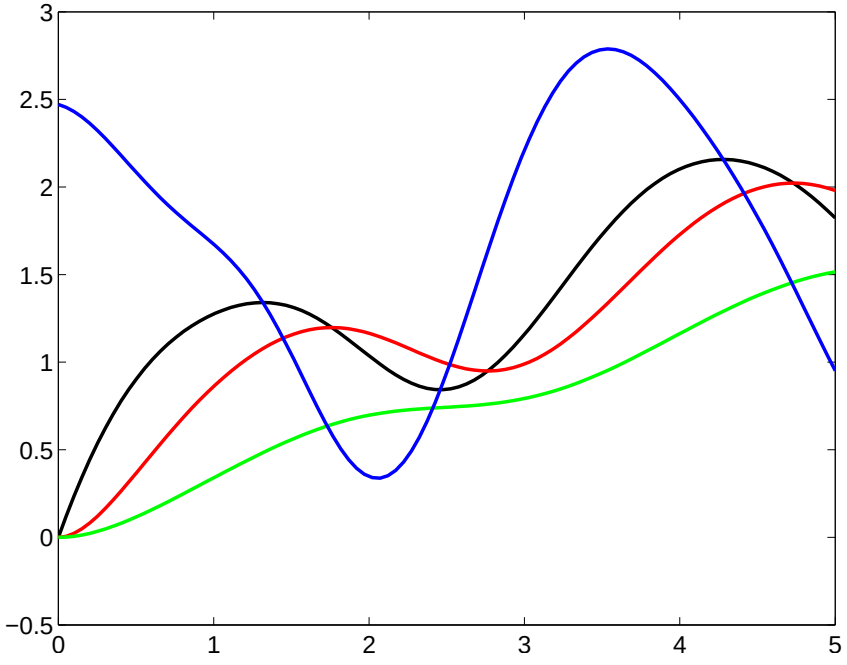


Figure: Joint samples from the ODE covariance, *blue*: $y(t)$ (mRNA of TF), *black*: $f(t)$ (TF concentration), *red*: $x_1(t)$ (high decay target) and *green*: $x_2(t)$ (low decay target)

.

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

- `disimSample`

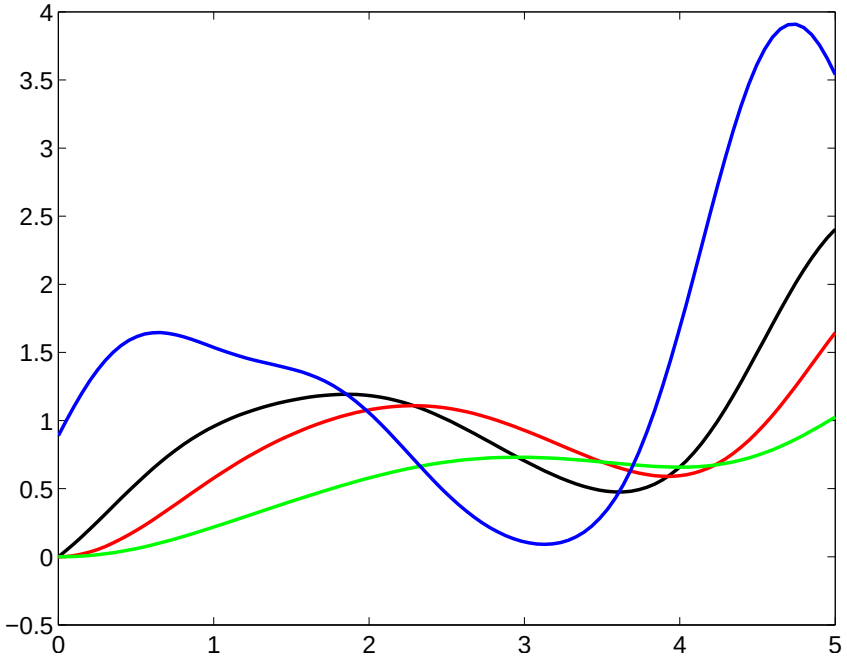


Figure: Joint samples from the ODE covariance, *blue*: $y(t)$ (mRNA of TF), *black*: $f(t)$ (TF concentration), *red*: $x_1(t)$ (high decay target) and *green*: $x_2(t)$ (low decay target)

.

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

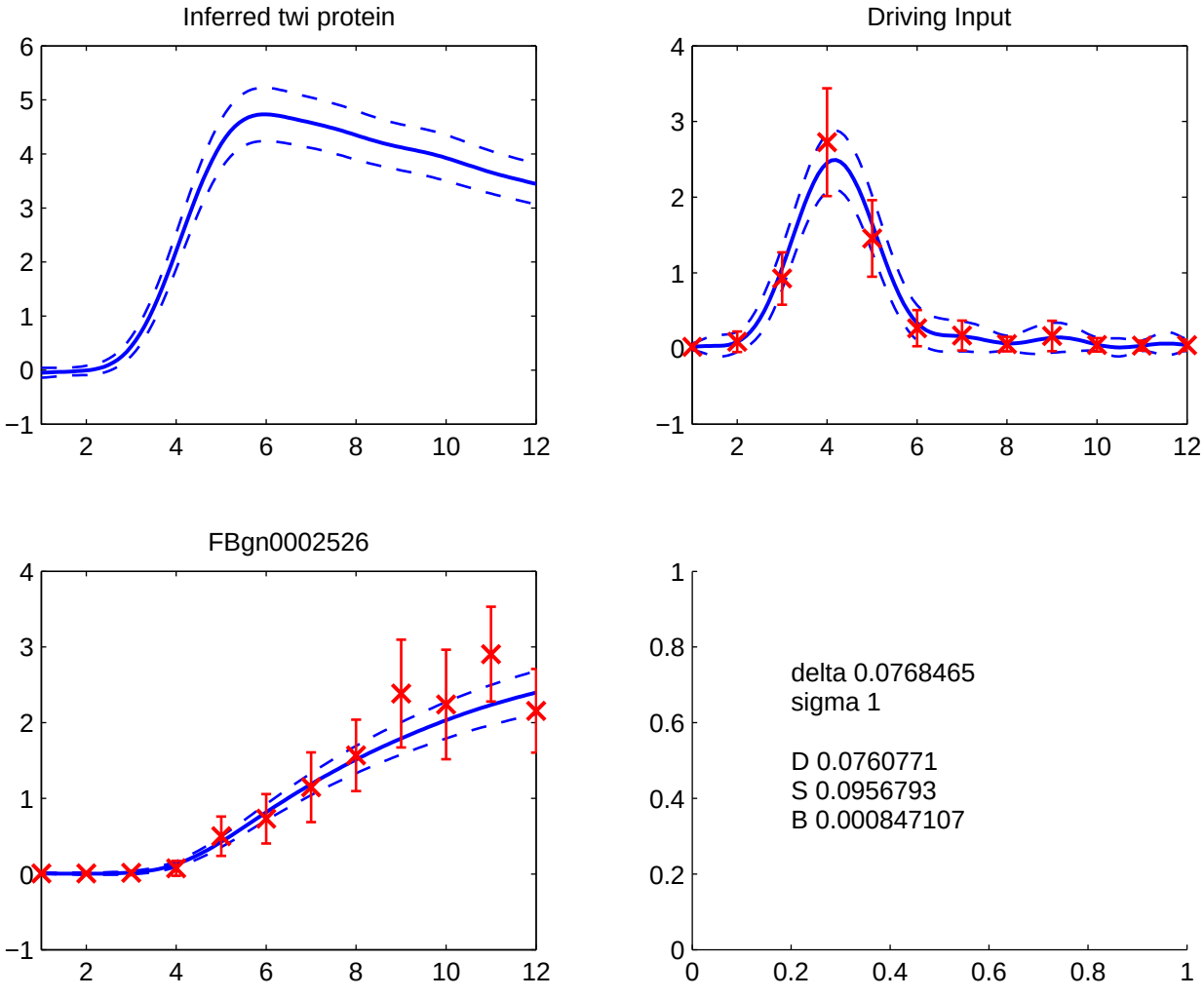


Figure: Model for flybase gene identity FBgn0002526.

Results for Twi using the Cascade model

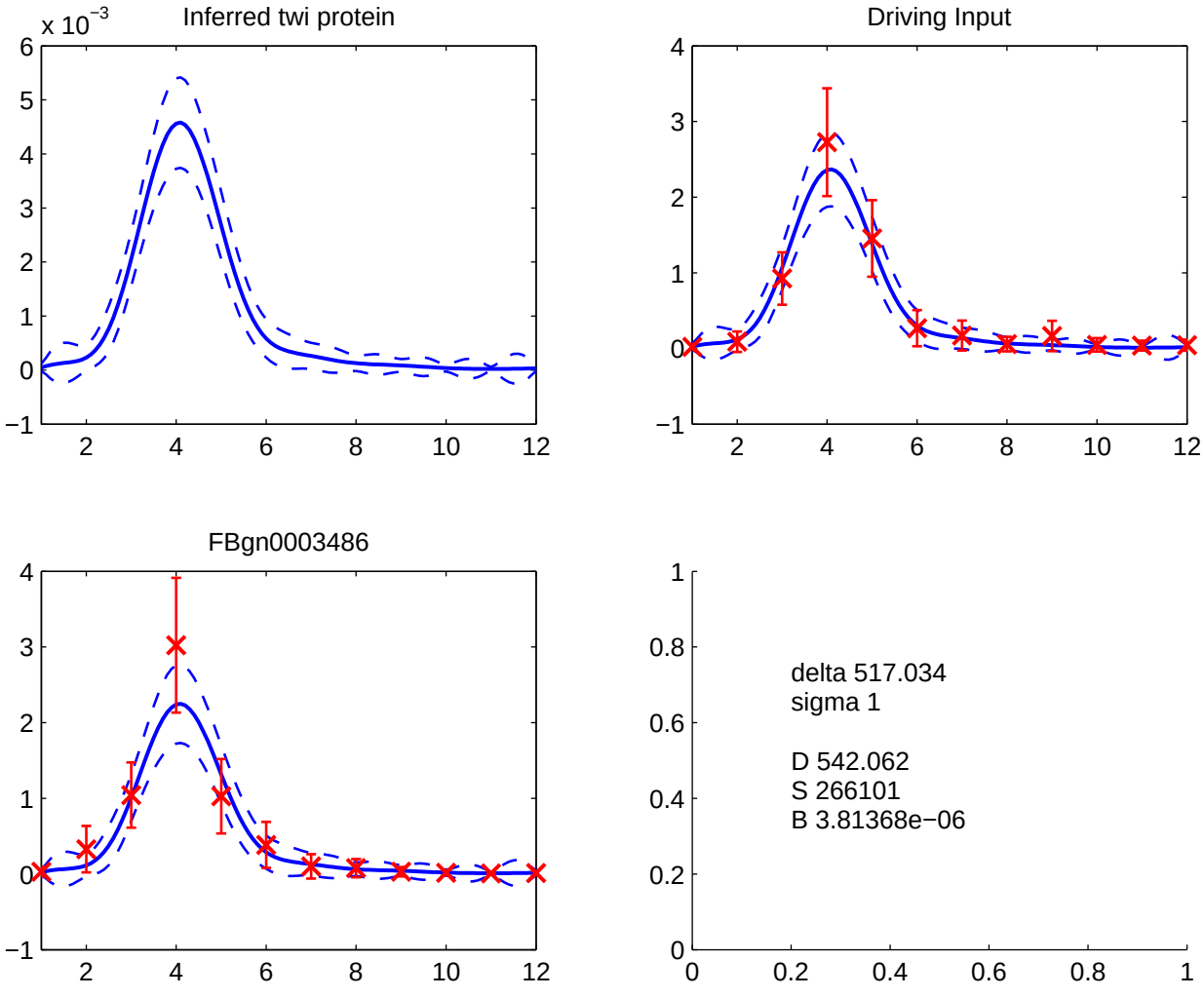


Figure: Model for flybase gene identity FBgn0003486.

Results for Twi using the Cascade model

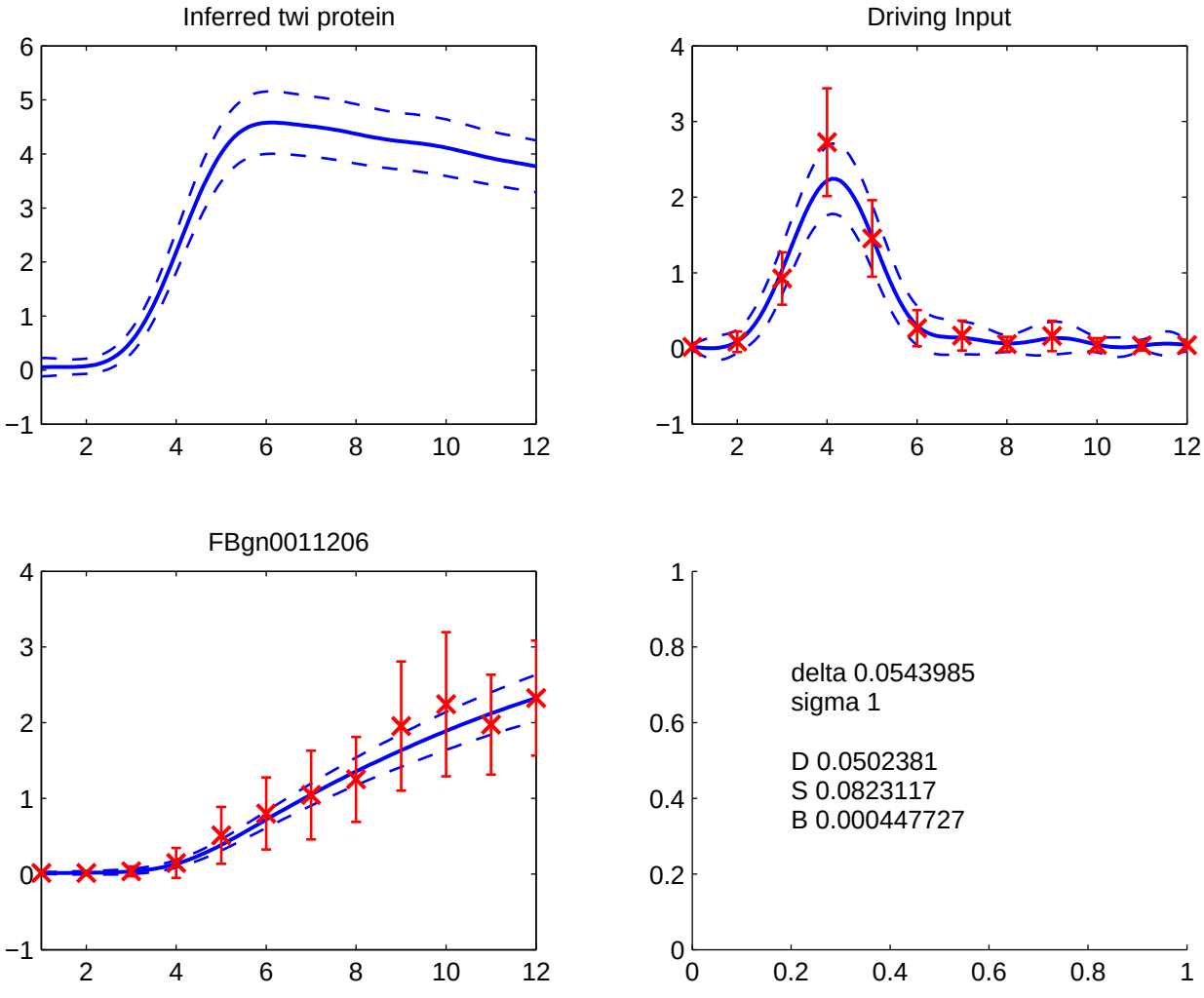


Figure: Model for flybase gene identity FBgn0011206.

Results for Twi using the Cascade model

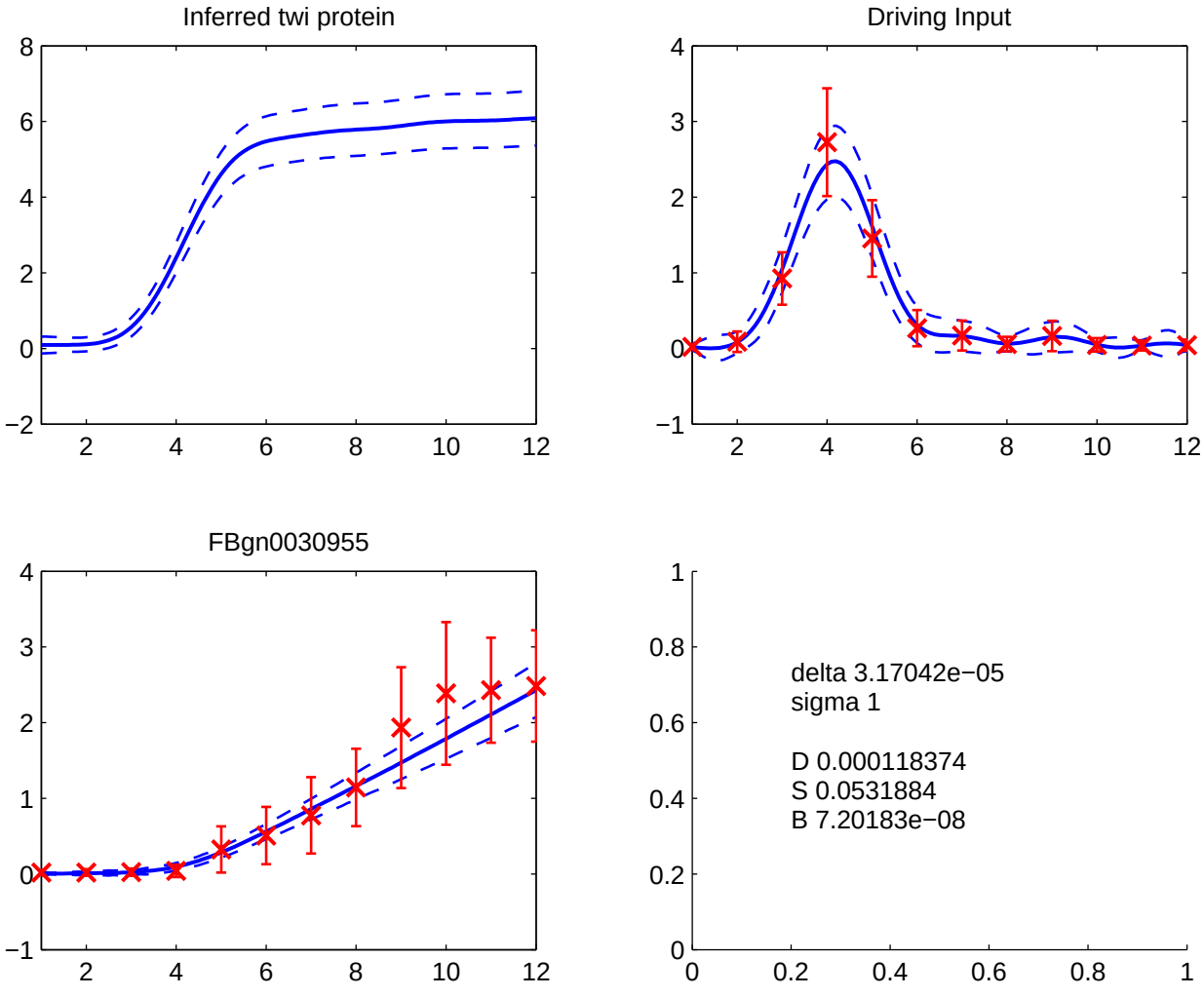


Figure: Model for flybase gene identity FBgn00309055.

Results for Twi using the Cascade model

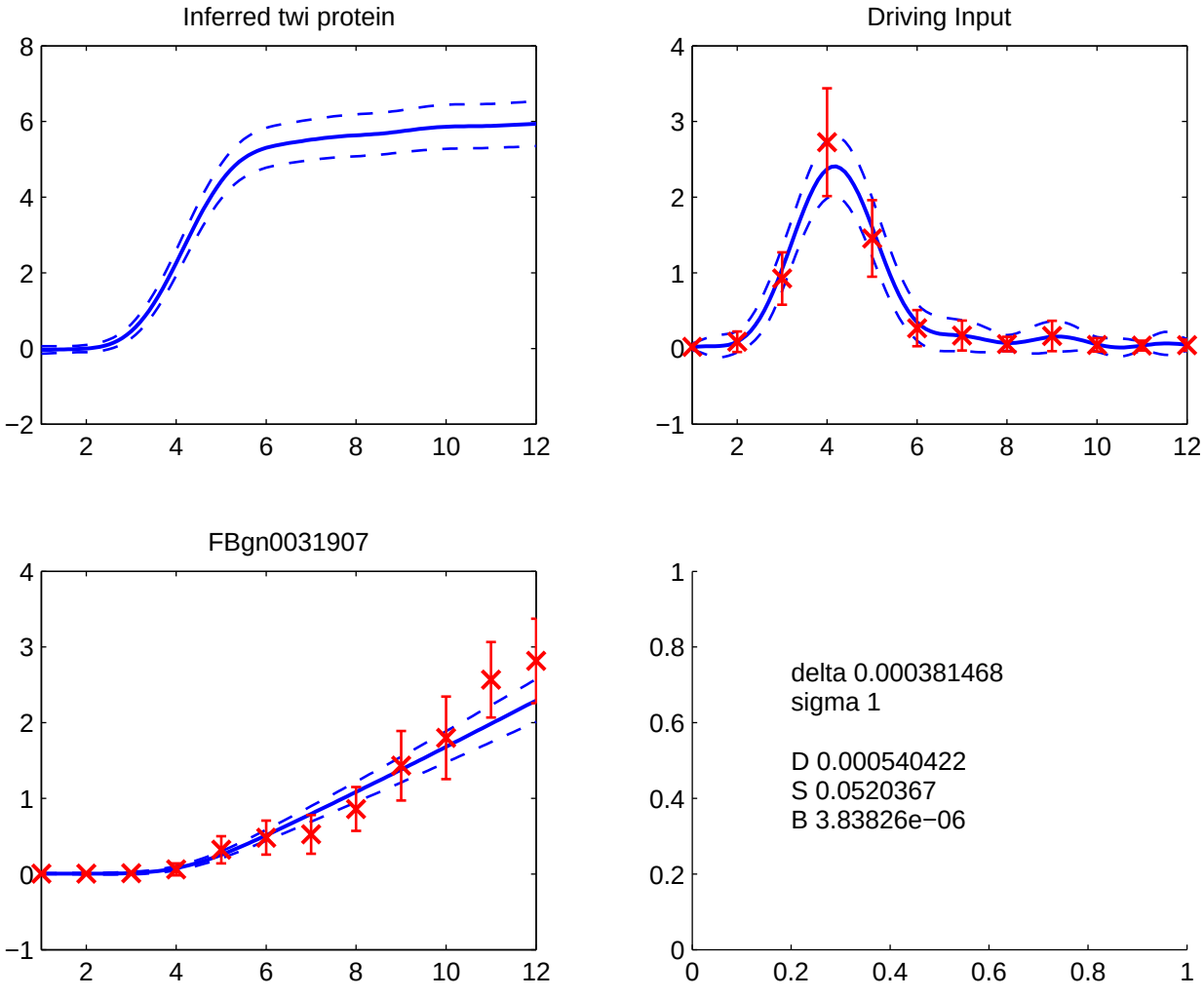


Figure: Model for flybase gene identity FBgn0031907.

Results for Twi using the Cascade model

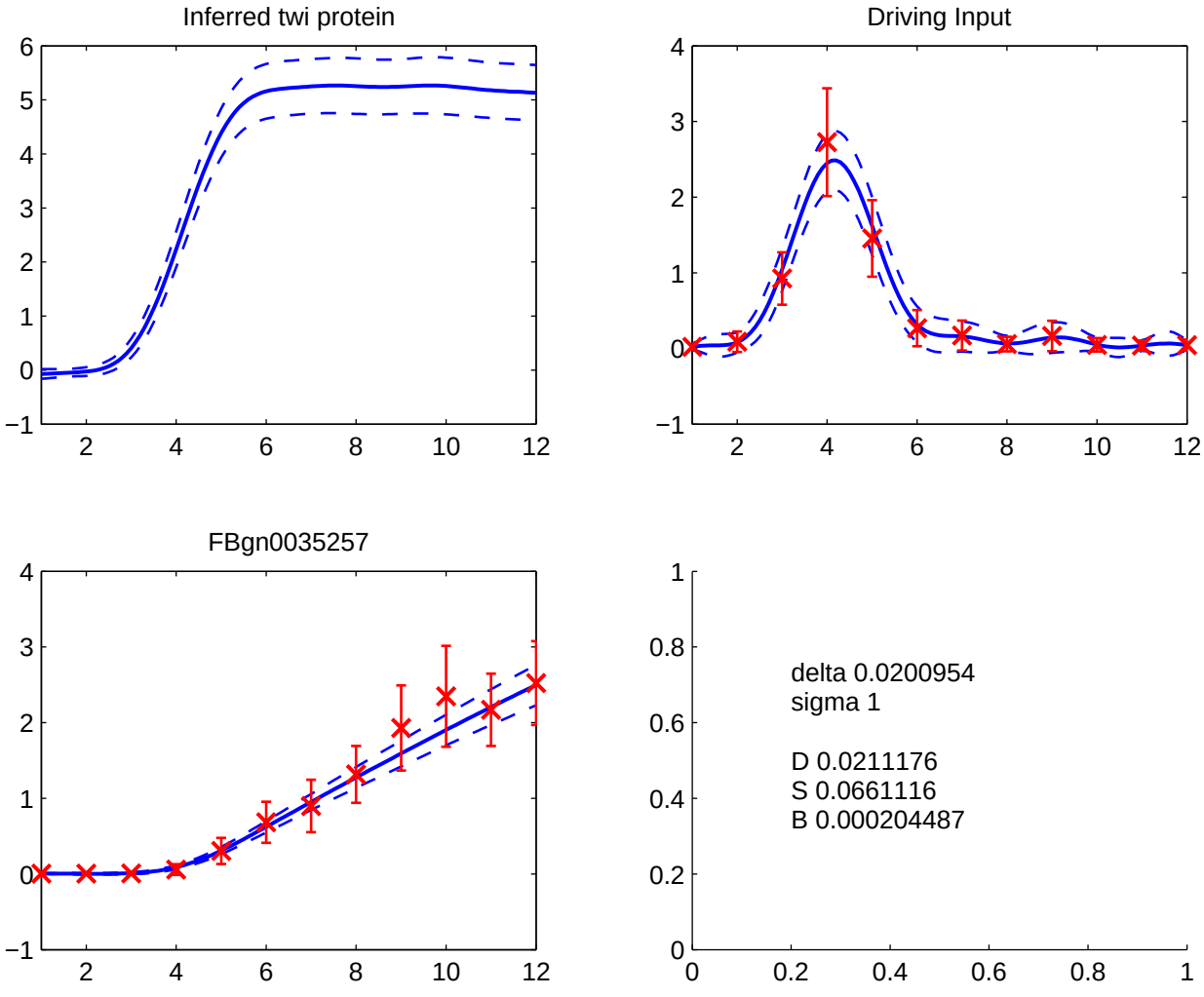


Figure: Model for flybase gene identity FBgn0035257.

Results for Twi using the Cascade model

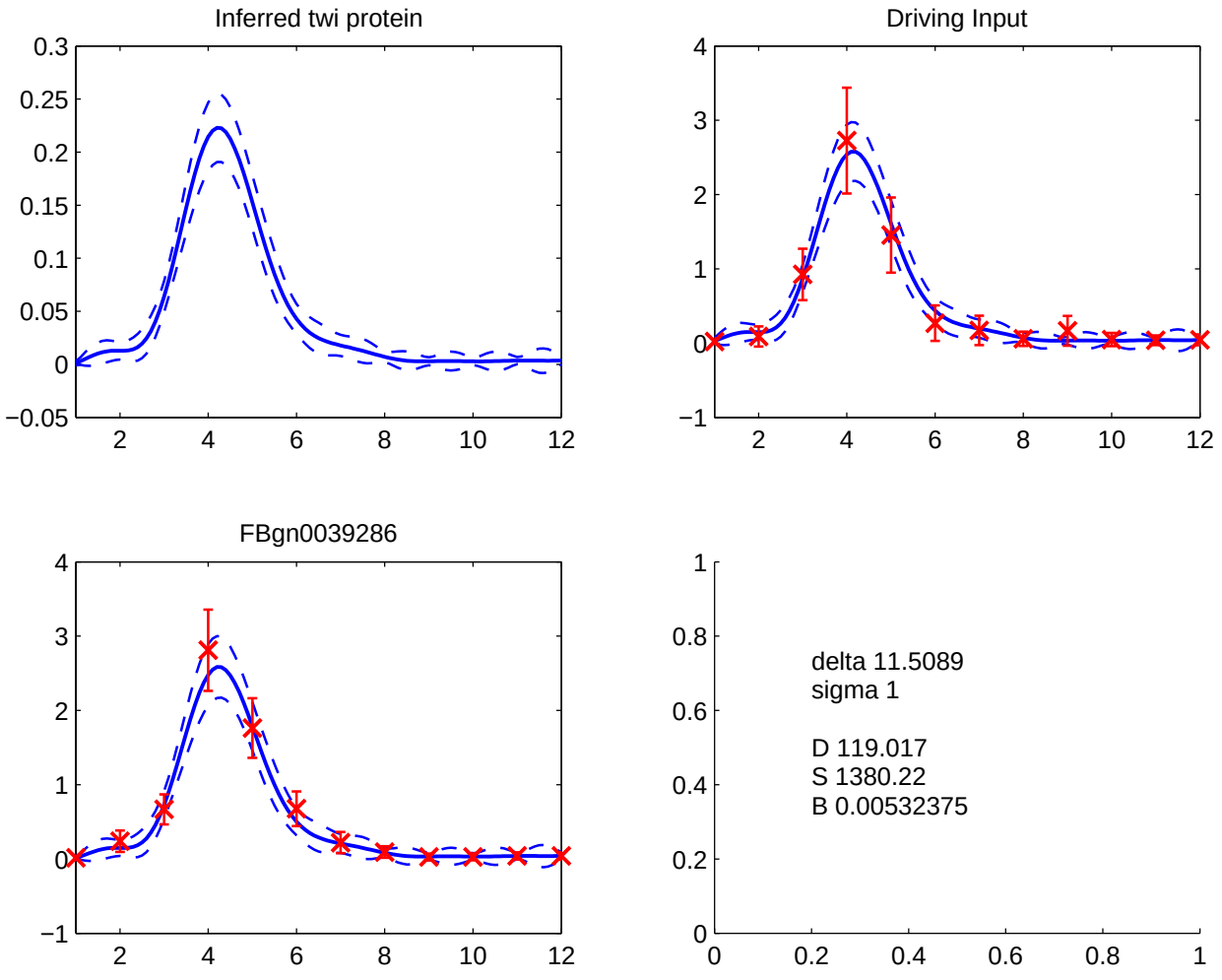
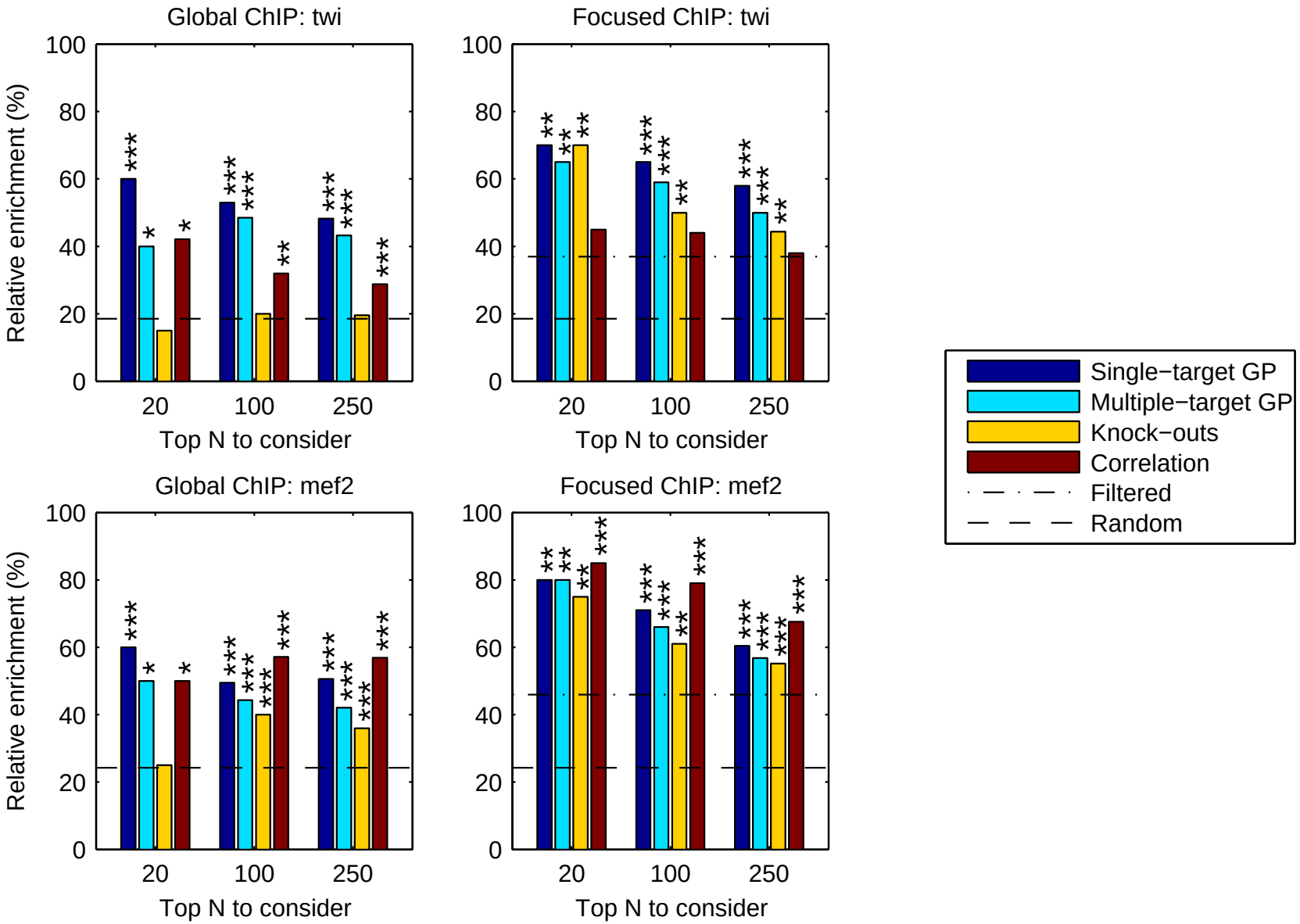


Figure: Model for flybase gene identity FBgn0039286.

Evaluation methods

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

Results



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

Summary

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

Outline

Discussion and Future Work

- Integration of probabilistic inference with mechanistic models.
- Ongoing/other work:
  - Non linear response and non linear differential equations.
  - Scaling up to larger systems Álvarez et al. (2010); Álvarez and Lawrence (2009).
  - Discontinuities through Switched Gaussian Processes Álvarez et al. (2011)
  - Robotics applications.
  - Applications to other types of system, *e.g.* spatial systems.
  - Stochastic differential equations Álvarez et al. (2010).

Acknowledgements

Investigators Neil Lawrence and Magnus Rattray

Researchers Mauricio Álvarez, Pei Gao, Antti Honkela, David Luengo, Guido Sanguinetti, Michalis Titsias, and Jennifer Withers

p53 pathway Martino Barenco and Mike Hubank at UCL Institute of Child Health.

D. Melanogaster Charles Girardot and Eileen Furlong of EMBL in Heidelberg.

Lawrence/Ratray Funding BBSRC award "Improved Processing of microarray data using probabilistic models", EPSRC award "Gaussian Processes for Systems Identification with applications in Systems Biology", University of Manchester, Computer Science Studentship, and **Google Research Award**: "Mechanistically Inspired Convolution Processes for Learning".

Other funding David Luengo's visit to Manchester was financed by the Comunidad de Madrid (project PRO-MULTIDIS-CM, S-0505/TIC/0233), and by the Spanish government (CICYT project TEC2006-13514-C02-01 and researh grant JC2008-00219).

Antti Honkela visits to Manchester funded by PASCAL I & II

References I

M. Álvarez and N. D. Lawrence. Sparse convolved Gaussian processes for multi-output regression. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 21, pages 57–64, Cambridge, MA, 2009. MIT Press. [PDF].

M. Álvarez, D. Luengo, and N. D. Lawrence. Latent force models. In van Dyk and Welling (2009), pages 9–16. [PDF].

M. A. Álvarez, D. Luengo, M. K. Titsias, and N. D. Lawrence. Efficient multioutput Gaussian processes through variational inducing kernels. In Y. W. Teh and D. M. Titterington, editors, *Proceedings of the Thirteenth International Workshop on Artificial Intelligence and Statistics*, volume 9, pages 25–32, Chia Laguna Resort, Sardinia, Italy, 13-16 May 2010. JMLR W&CP 9. [PDF].

M. A. Álvarez, J. Peters, B. Schölkopf, and N. D. Lawrence. Switched latent force models for movement segmentation. In J. Shawe-Taylor, R. Zemel, C. Williams, and J. Lafferty, editors, *Advances in Neural Information Processing Systems*, volume 23, Cambridge, MA, 2011. MIT Press. To appear.

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.

P. Boyle and M. Frean. Dependent Gaussian processes. In L. Saul, Y. Weiss, and L. Bouttou, editors, *Advances in Neural Information Processing Systems*, volume 17, pages 217–224, Cambridge, MA, 2005. MIT Press.

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.

References II

P. Goovaerts. *Geostatistics For Natural Resources Evaluation*. Oxford University Press, 1997. [Google Books] .

D. M. Higdon. Space and space-time modelling using process convolutions. In C. Anderson, V. Barnett, P. Chatwin, and A. El-Shaarawi, editors, *Quantitative methods for current environmental issues*, pages 37–56. Springer-Verlag, 2002.

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].

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.

E. Mjolsness, D. H. Sharp, and J. Reinitz. A connectionist model of development. *Journal of Theoretical Biology*, 152(4):429–453, 1991.

M. A. Osborne, A. Rogers, S. D. Ramchurn, S. J. Roberts, and N. R. Jennings. Towards real-time information processing of sensor network data using computationally efficient multi-output Gaussian processes. In *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN 2008)*, 2008.

A. D. Polyanin. *Handbook of Linear Partial Differential Equations for Engineers and Scientists*. Chapman & Hall/CRC, 1 edition, 2002.

J. Quiñonero Candela and C. E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005.

S. T. Roweis. EM algorithms for PCA and SPCA. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10, pages 626–632, Cambridge, MA, 1998. MIT Press.

E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18, Cambridge, MA, 2006. MIT Press.

References III

Y. W. Teh, M. Seeger, and M. I. Jordan. Semiparametric latent factor models. In R. G. Cowell and Z. Ghahramani, editors, *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pages 333–340, Barbados, 6-8 January 2005. Society for Artificial Intelligence and Statistics.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, B*, 6(3):611–622, 1999. [PDF]. [DOI].

M. K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In van Dyk and Welling (2009), pages 567–574.

D. van Dyk and M. Welling, editors. *Artificial Intelligence and Statistics*, volume 5, Clearwater Beach, FL, 16-18 April 2009. JMLR W&CP 5.

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

Partial Differential Equations and Latent Forces

Mauricio Alvarez

- Can extend the concept to latent functions in PDEs.
- Jura data: concentrations of heavy metal pollutants from the Swiss Jura.
- Consider a latent function that represents how the pollutants were originally laid down (initial condition).
- Assume pollutants diffuse at different rates resulting in the concentrations observed in the data set.

$$\frac{\partial x_q(\mathbf{x}, t)}{\partial t} = \sum_{j=1}^{d} \kappa_q \frac{\partial^2 x_q(\mathbf{x}, t)}{\partial x_j^2},$$

- Latent function $f_r(\mathbf{x})$ represents the concentration of pollutants at time zero (i.e. the system's initial condition).

Solution to the PDE

- The solution to the system (Polyanin, 2002) is then given by

$$x_q(\mathbf{x}, t) = \sum_{r=1}^{R} S_{rq} \int_{\mathbb{R}^d} f_r(\mathbf{x}') G_q(\mathbf{x}, \mathbf{x}', t) \mathrm{d}\mathbf{x}'$$

where $G_q(\mathbf{x}, \mathbf{x}', t)$ is the Green's function given as

$$G_q(\mathbf{x}, \mathbf{x}', t) = \frac{1}{2^d \pi^{d/2} T_q^{d/2}} \exp\left[-\sum_{j=1}^{d} \frac{(x_j - x_j')^2}{4 T_q} \right],$$

with $T_q = \kappa_q t$.

Covariance Function

Mauricio Alvarez

- For latent function given by a GP with the RBF covariance function this is tractable.

$$k_{x_p x_q}(\mathbf{x}, \mathbf{x}', t) = \sum_{r=1}^{R} \frac{S_{rp} S_{rq} |\mathbf{L}_r|^{1/2}}{|\mathbf{L}_{rp} + \mathbf{L}_{rq} + \mathbf{L}_r|^{1/2}} \times \exp\left[-\frac{1}{2}\left(\mathbf{x} - \mathbf{x}'\right)^{\top} \left(\mathbf{L}_{rp} + \mathbf{L}_{rq} + \mathbf{L}_r\right)^{-1} \left(\mathbf{x} - \mathbf{x}'\right)\right],$$

where $\mathbf{L}_{rp}, \mathbf{L}_{rq}$ and $\mathbf{L}_r$ are diagonal isotropic matrices with entries $2\kappa_p t$, $2\kappa_q t$ and $1/\ell_r^2$ respectively. The covariance function between the output and latent functions is given by

$$k_{x_q f_r}(\mathbf{x}, \mathbf{x}', t) = \frac{S_{rq} |\mathbf{L}_r|^{1/2}}{|\mathbf{L}_{rq} + \mathbf{L}_r|^{1/2}} \times \exp\left[-\frac{1}{2}\left(\mathbf{x} - \mathbf{x}'\right)^{\top} \left(\mathbf{L}_{rq} + \mathbf{L}_r\right)^{-1} \left(\mathbf{x} - \mathbf{x}'\right)\right].$$

Prediction of Metal Concentrations

Mauricio Alvarez

- Replicate experiments in (Goovaerts, 1997, pp. 248,249):
  - *Primary variable* (Cd, Cu, Pb, Co) predicted in conjunction with *secondary variables* (Ni and Zn for Cd; Pb, Ni, and Zn for Cu; Cu, Ni, and Zn for Pb; Ni and Zn for Co).[1]
- Condition on the secondary variables to improve prediction for primary variables.
- Compare results for the diffusion kernel with independent GPs and "ordinary co-kriging" (Goovaerts, 1997, pp. 248,249).

[1] Data available at `http://www.ai-geostats.org/`.

Jura Results

Table: Mean absolute error and standard deviation for ten repetitions of the experiment for the Jura dataset. IGPs stands for independent GPs, GPDK stands for GP diffusion kernel, OCK for ordinary co-kriging. For the Gaussian process with diffusion kernel, we learn the diffusion coefficients and the length-scale of the covariance of the latent function.

Metals	IGPs	GPDK	OCK
Cd	0.5823±0.0133	0.4505±0.0126	0.5
Cu	15.9357±0.0907	7.1677±0.2266	7.8
Pb	22.9141±0.6076	10.1097±0.2842	10.7
Co	2.0735±0.1070	1.7546±0.0895	1.5

Outline

Convolutions and Computational Complexity

Mauricio Alvarez

- Solutions to these differential equations is normally as a convolution.

$$x_i(t) = \int f(u)\, k_i(u-t)\, \mathrm{d}u + h_i(t)$$

$$x_i(t) = \int_0^t f(u)\, g_i(u)\, \mathrm{d}u + h_i(t)$$

- Convolution Processes (Higdon, 2002; Boyle and Frean, 2005).
- Convolutions lead to $N \times d$ size covariance matrices $O\left(N^3 d^3\right)$ complexity, $O\left(N^2 d^2\right)$ storage.
- Model is conditionally independent over $\{x_i(t)\}_{i=1}^d$ given $f(t)$.

Independence Assumption

- Can assume conditional independence given given $\{f(t_i)\}_{i=1}^{k}$. (Álvarez and Lawrence, 2009)
  - Result is very similar to PITC approximation (Quiñonero Candela and Rasmussen, 2005).
  - Reduces to $O\left(N^3dk^2\right)$ complexity, $O\left(N^2dk\right)$ storage.
  - Can also do a FITC style approximation (Snelson and Ghahramani, 2006).
  - Reduces to $O\left(Ndk^2\right)$ complexity, $O\left(Ndk\right)$ storage.

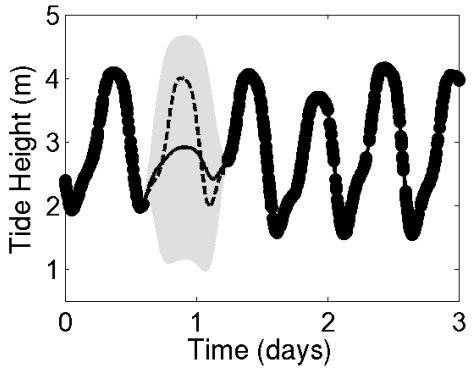
Tide Sensor Network

Mauricio Alvarez

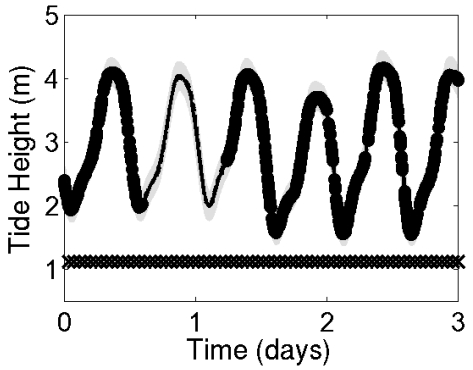
- Network of tide height sensors in the solent — tide heights are correlated.
- Data kindly provided by Alex Rogers (see Osborne et al., 2008).
- $d = 3$ and $N = 1000$ of the 4320 for the training set.
- Simulate sensor failure by knocking out onse sensor for a given time.
- For the other two sensors we used all 1000 training observations.
- Take $k = 100$.

Tide Height Results

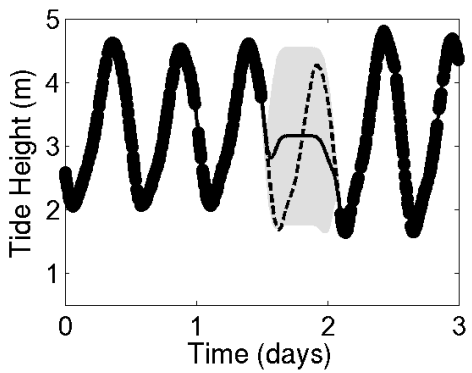
Mauricio Alvarez



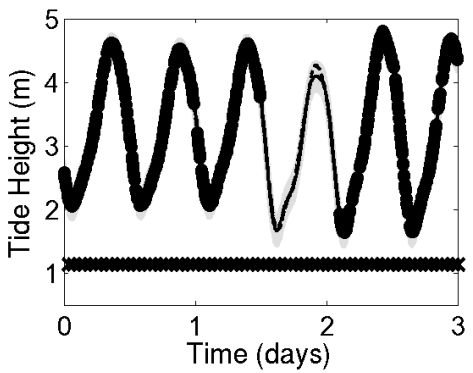
(a) Bramblemet Independent



(b) Bramblemet PITC



(c) Cambermet Independent



(d) Cambermet PITC

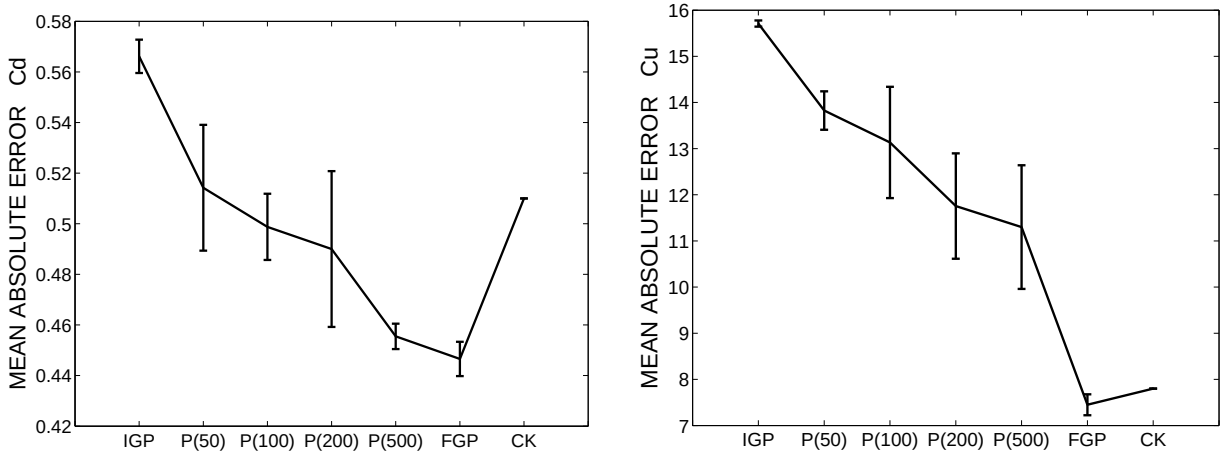
Cokriging Jura

Mauricio Alvarez

- Jura dataset — concentrations of several heavy metals.
- Prediction 259 data, validation 100 data points.
- Predict *primary variables* (cadmium and copper) at prediction locations in conjunction with some *secondary variables* (nickel and zinc for cadmium; lead, nickel and zinc for copper) (Goovaerts, 1997, p. 248,249).

Swiss Jura Results

Mauricio Alvarez



(a) Cadmium (b) Copper

Figure: Mean absolute error. IGP stands for independent GP, P(M) stands for PITC with M inducing values, FGP stands for full GP and CK stands for ordinary co-kriging.

Outline

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,

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

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

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

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

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

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

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

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

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

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

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, $x_i(t)$, that solves the ODE in Eq. 1 can be approximated by

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

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

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

Khanin et al. Results

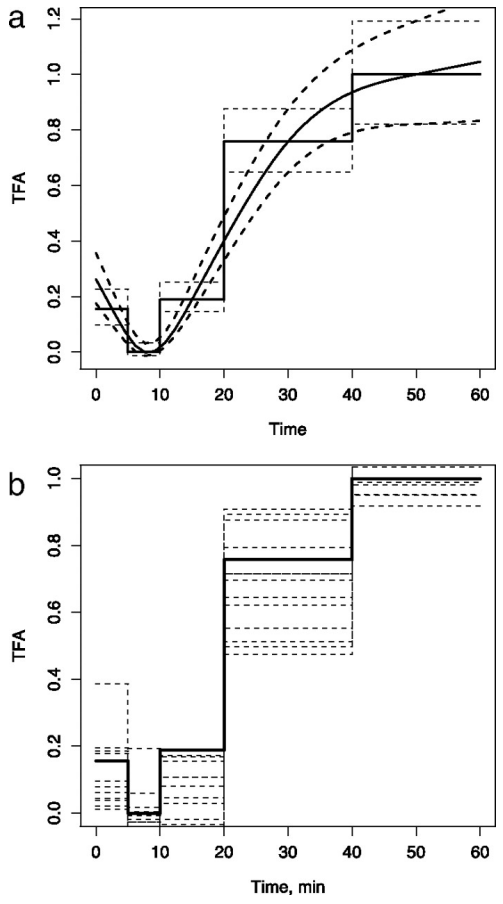


Figure: Fig. 2 from Khanin et al. (2006): Reconstructed activity level of master repressor LexA, following a UV dose of 40 J/m2.

Khanin et al. Results

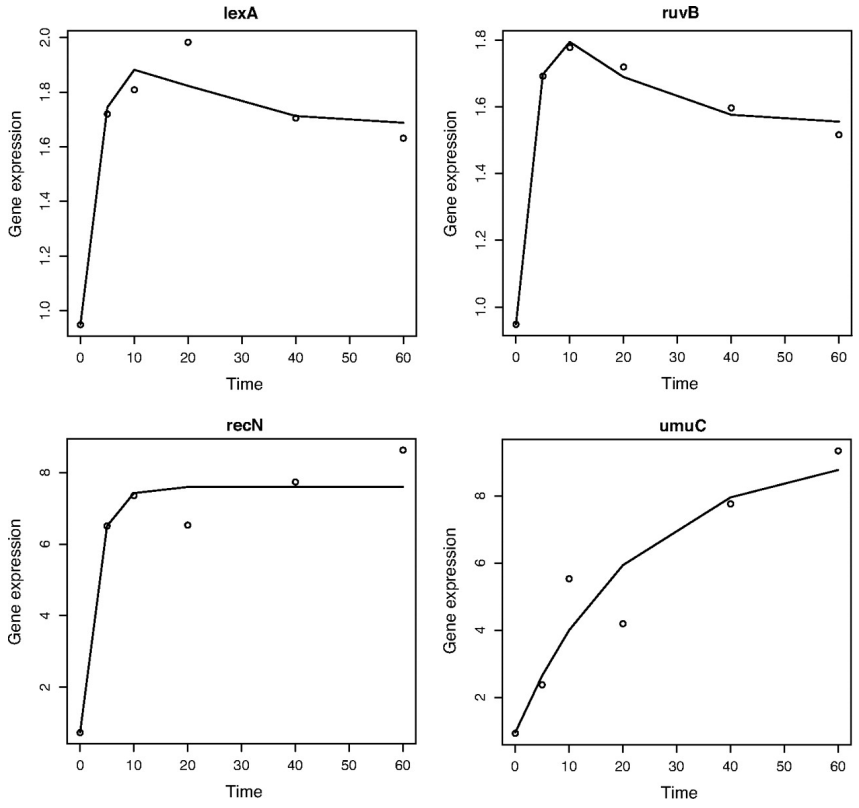


Figure: Fig. 3 from Khanin et al. (2006): Reconstructed profiles for four genes in the LexA SIM.

Repression Model

Pei Gao

- We can use the same model of repression,

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

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

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

Results for the repressor LexA

Pei Gao

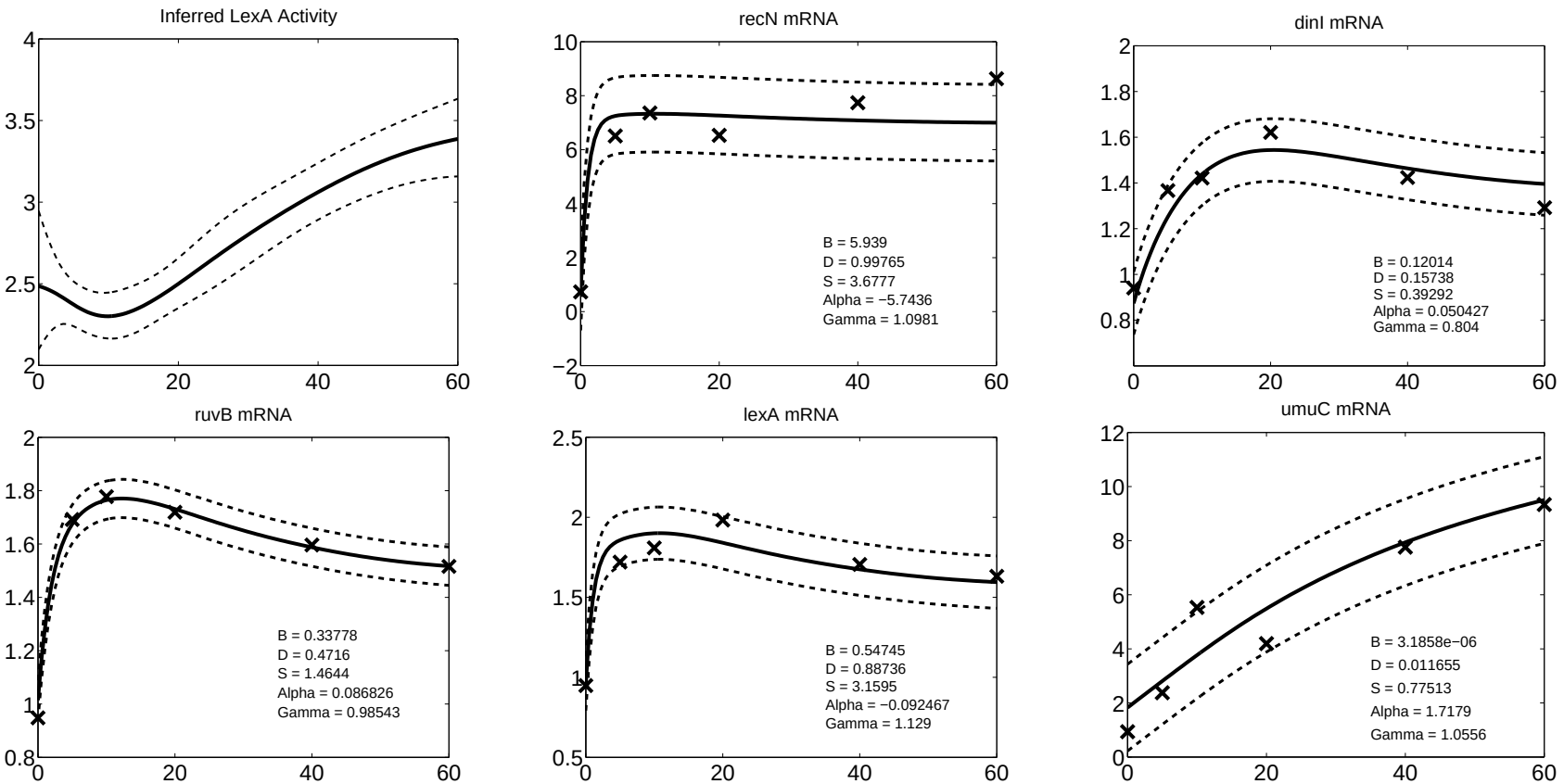


Figure: Our results using an MLP kernel. From Gao et al. (2008).

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?

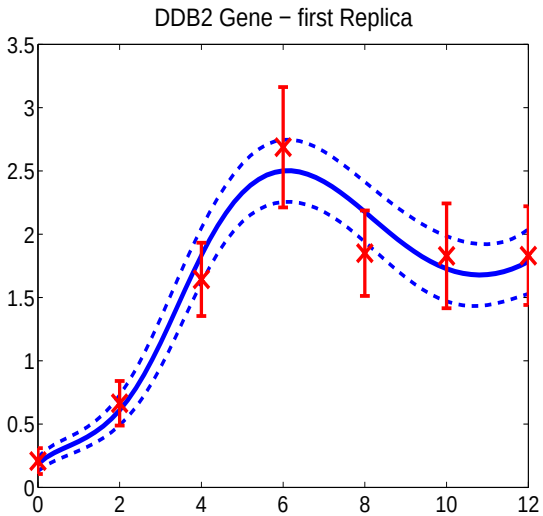
p53 System Again

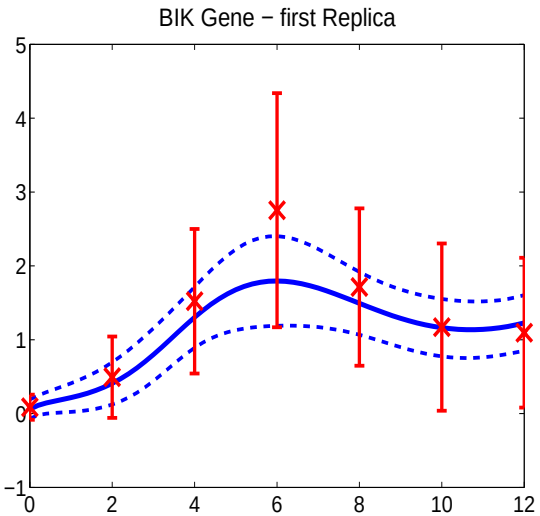
- One transcription factor (p53) that acts as an activator. We consider the Michaelis-Menten kinetic equation

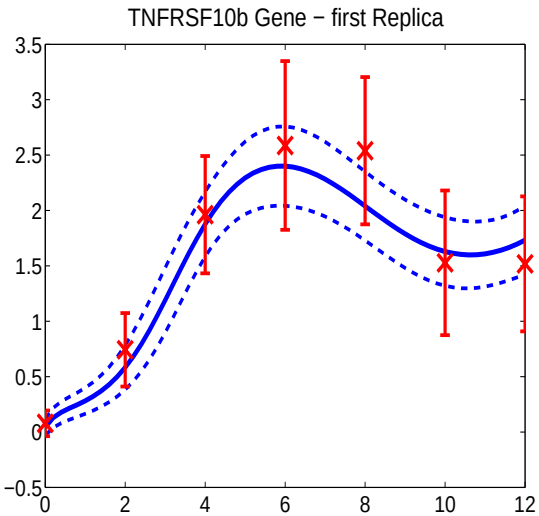
$$\frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = b_j + s_j \frac{\exp(f(t))}{\exp(f(t)) + \gamma_j} - d_j x_j(t)$$

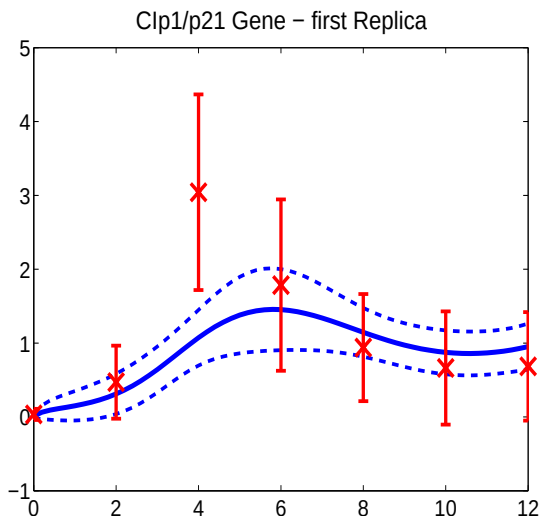
- We have 5 genes
- Gene expressions are available for $T = 7$ times and there are 3 replicas of the time series data
- TF ($\mathbf{f}$) is discretized using 121 points
- MCMC details:
  - 7 control points are used (placed in a equally spaced grid)
  - Running time 4/5 hours for 2 million sampling iterations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

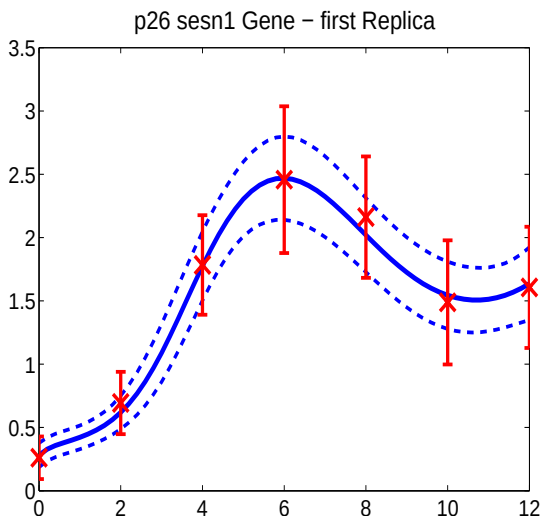
Data used by Barenco et al. (2006): Predicted gene expressions for the 1st replica



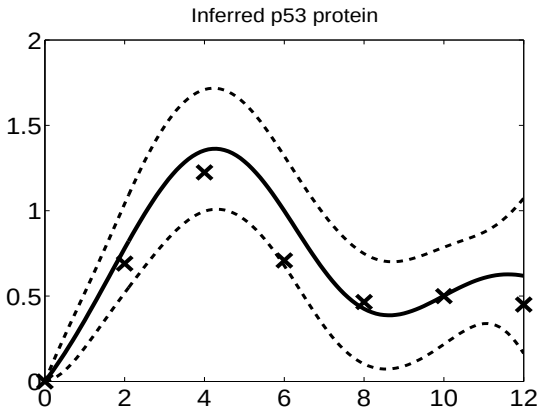


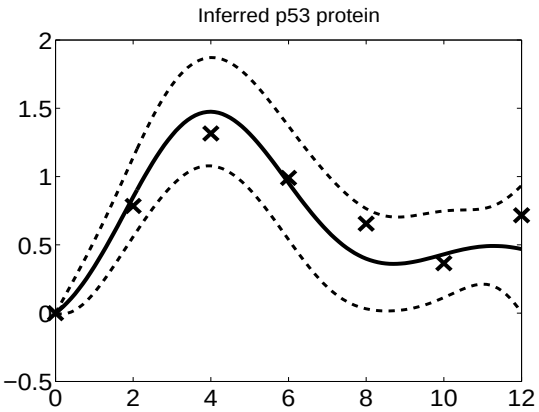


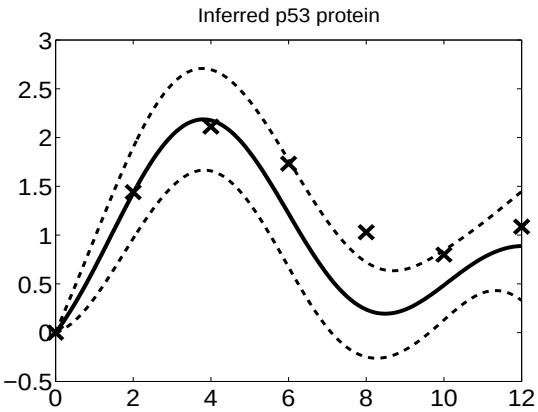




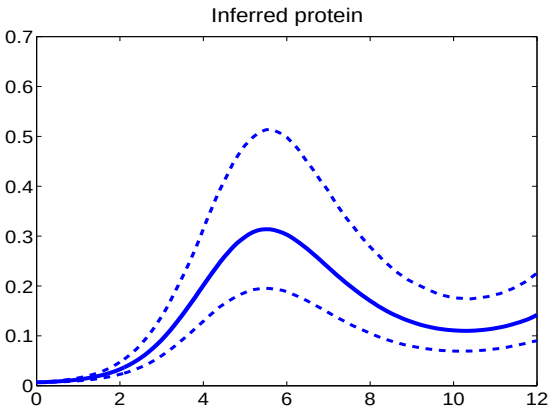
Data used by Barenco et al. (2006): Protein concentrations

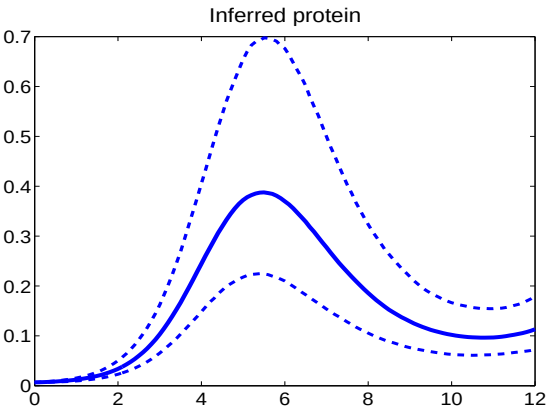


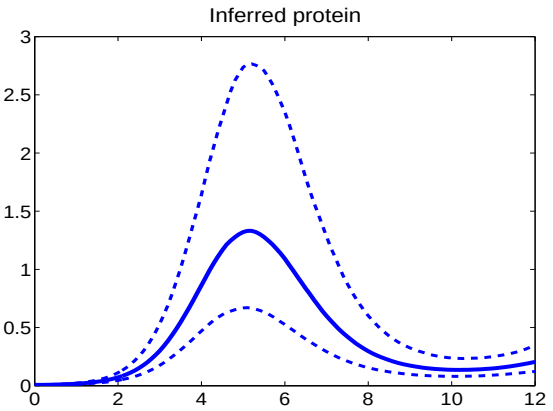




Linear model (Barenco et al. predictions are shown as crosses)

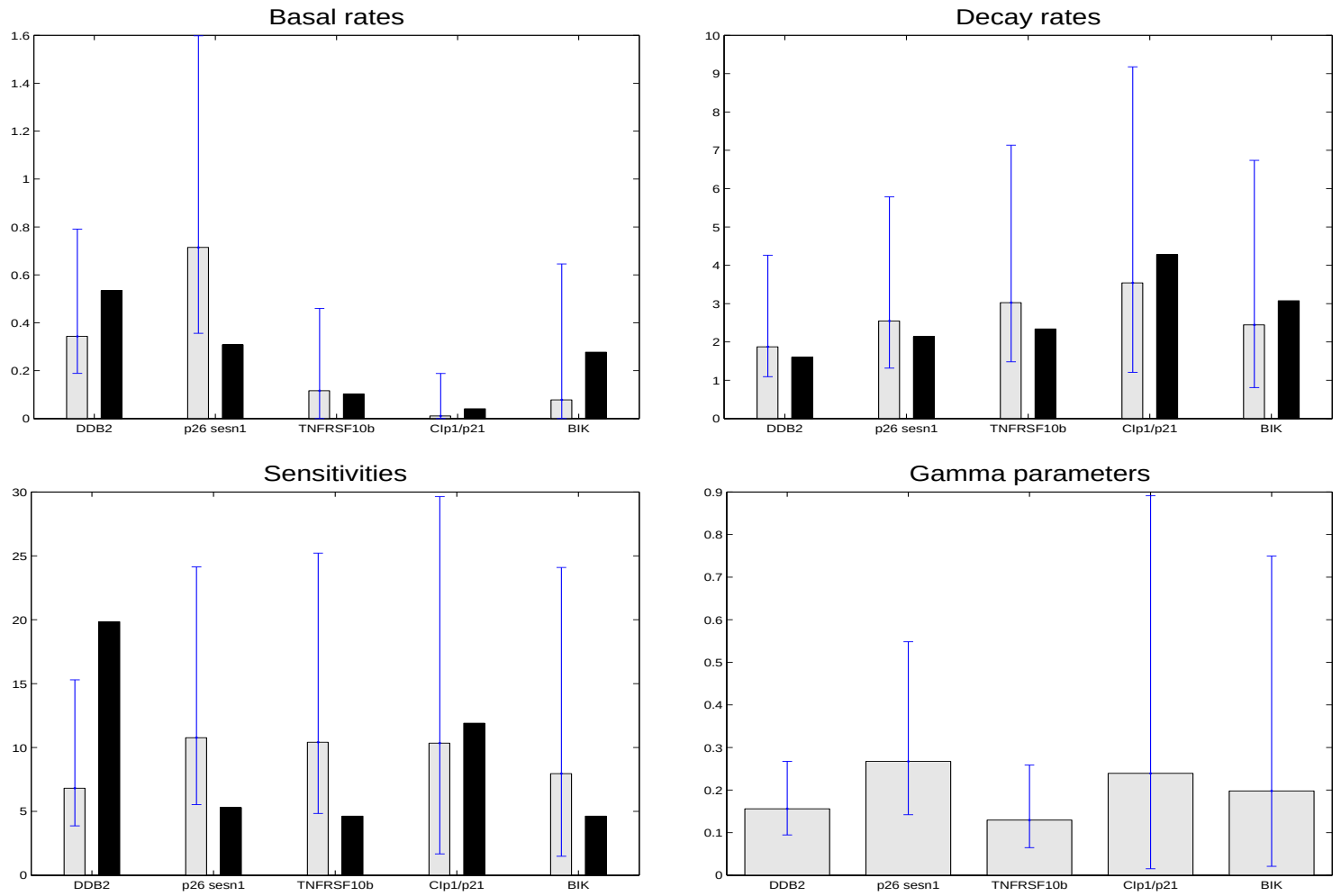






Nonlinear (Michaelis-Menten kinetic equation)

p53 Data Kinetic parameters



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

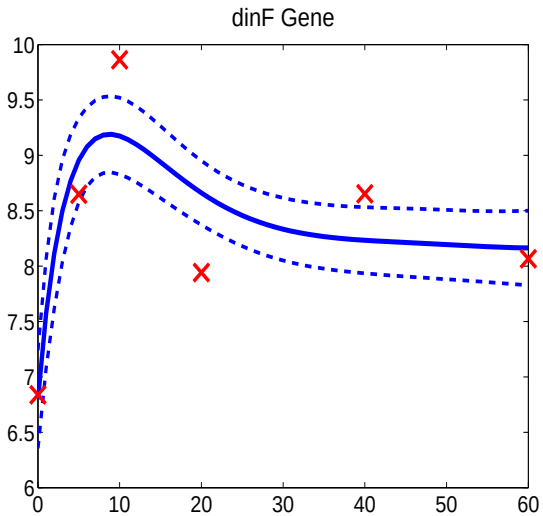
Results on SOS System

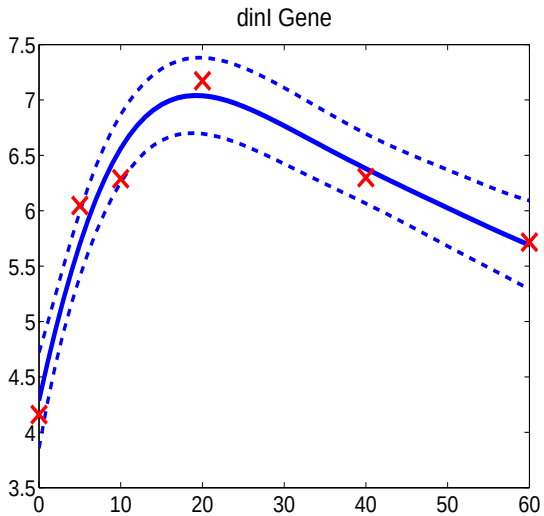
- Again consider the Michaelis-Menten kinetic equation

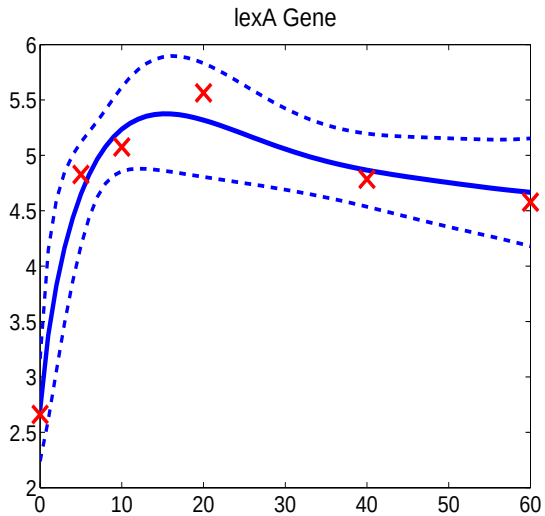
$$\frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = b_j + s_j \frac{1}{\exp(f(t)) + \gamma_j} - d_j x_j(t)$$

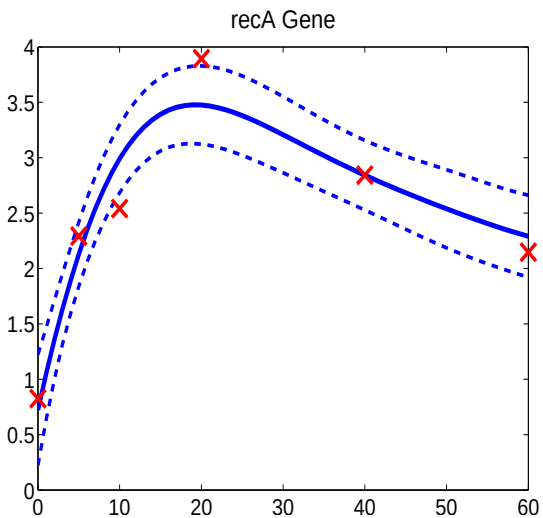
- We have 14 genes (5 kinetic parameters each)
- Gene expressions are available for $T = 6$ time slots
- TF ($\mathbf{f}$) is discretized using 121 points
- MCMC details:
  - 6 control points are used (placed in a equally spaced grid)
  - Running time was 5 hours for 2 million sampling iterations plus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

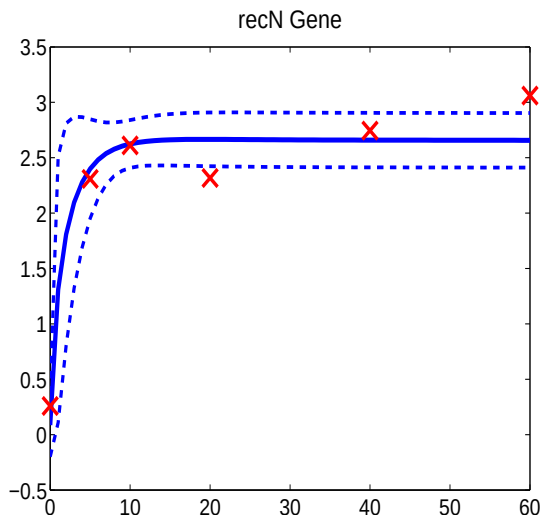
Results in E.coli data: Predicted gene expressions

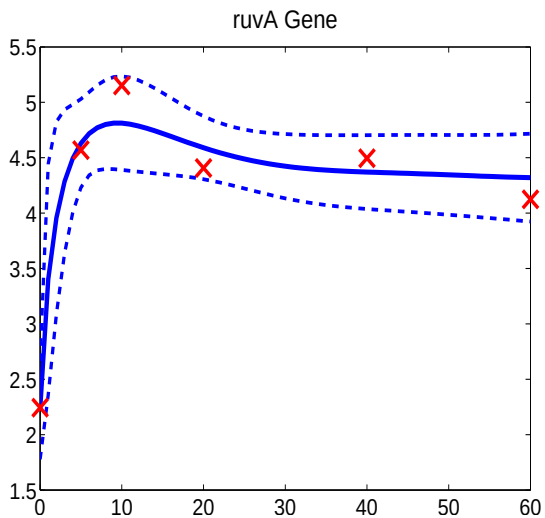




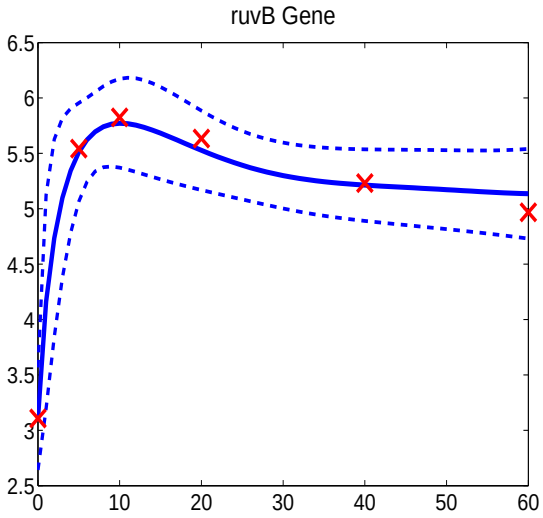


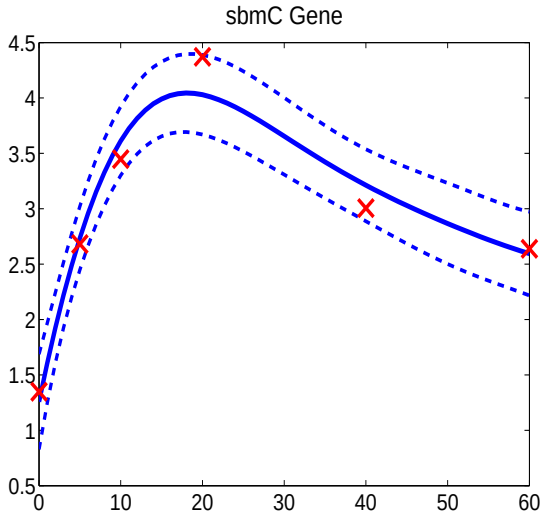


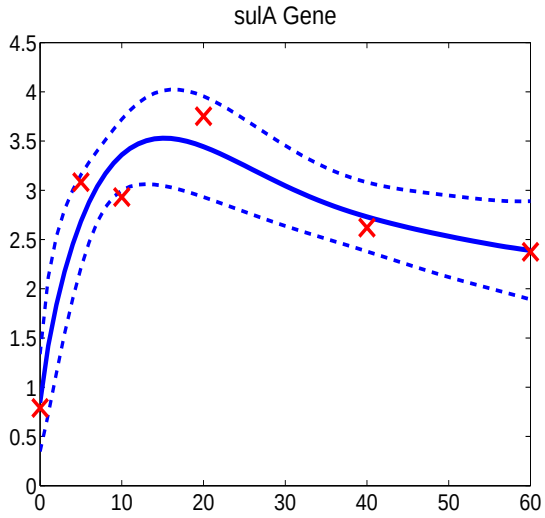


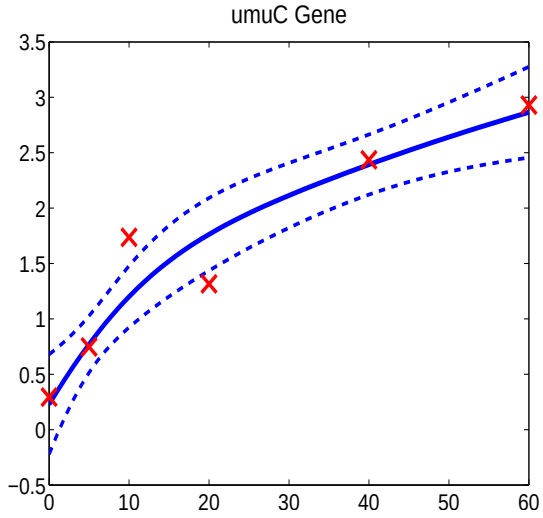


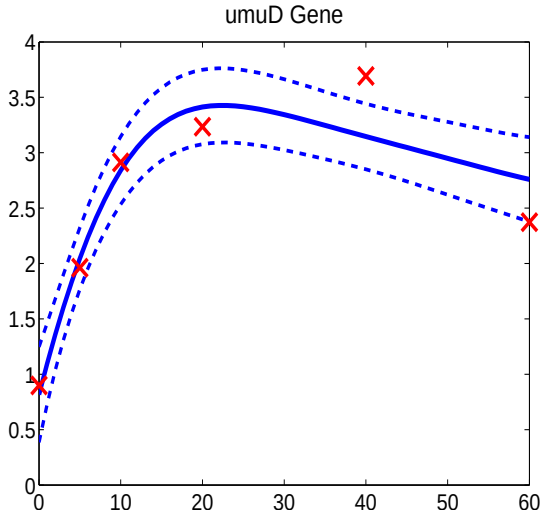
Results in E.coli data: Predicted gene expressions

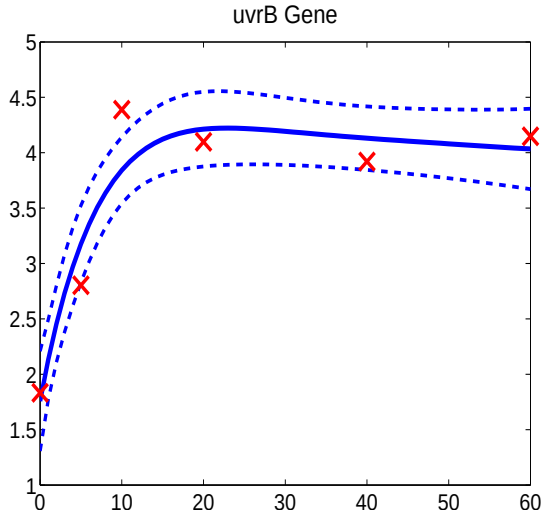




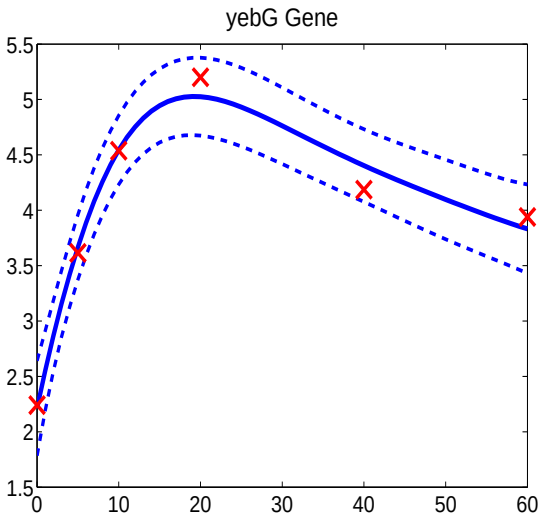


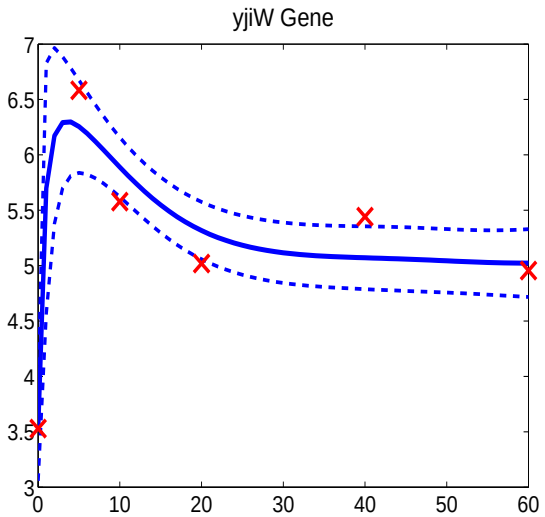




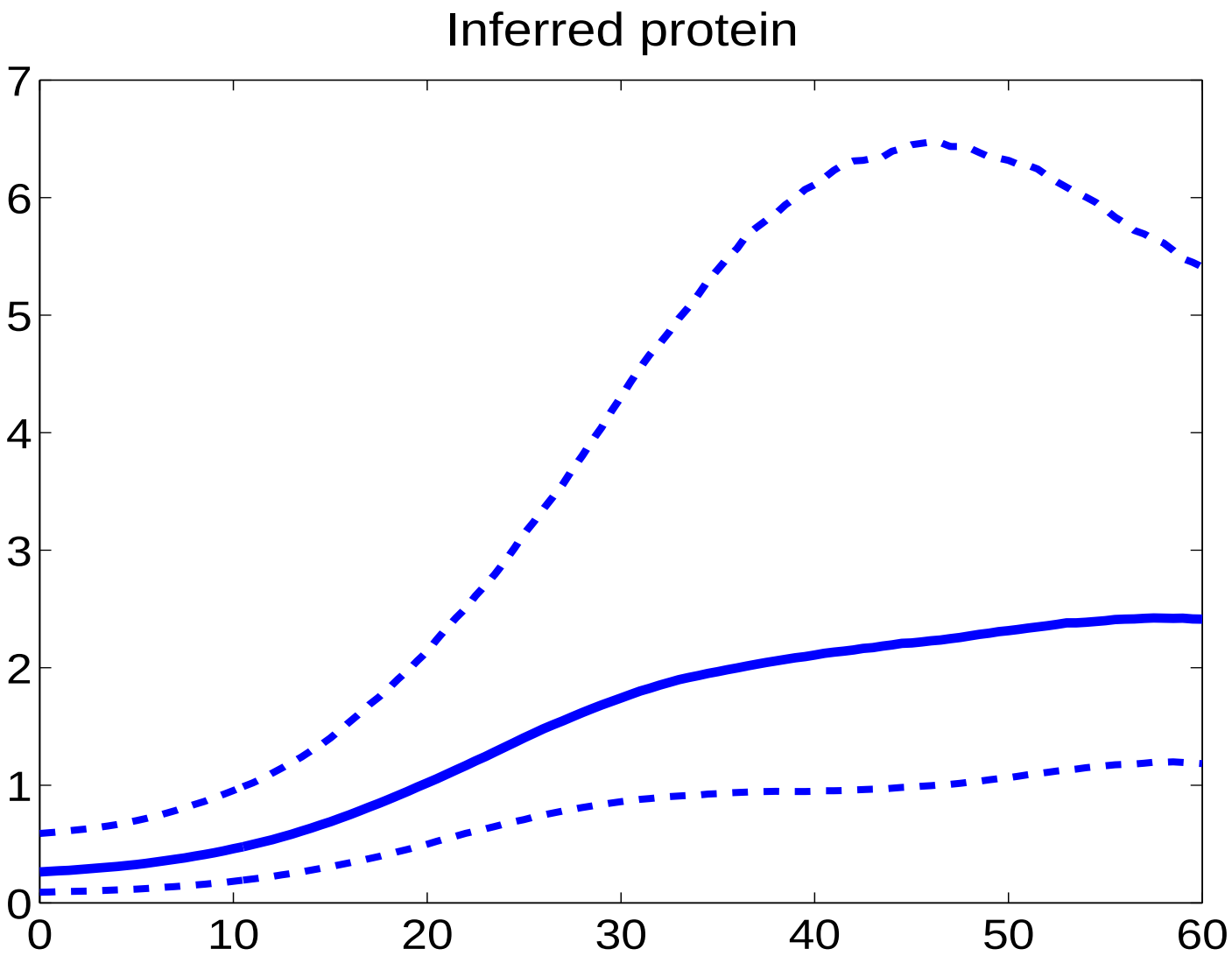


Results in E.coli data: Predicted gene expressions

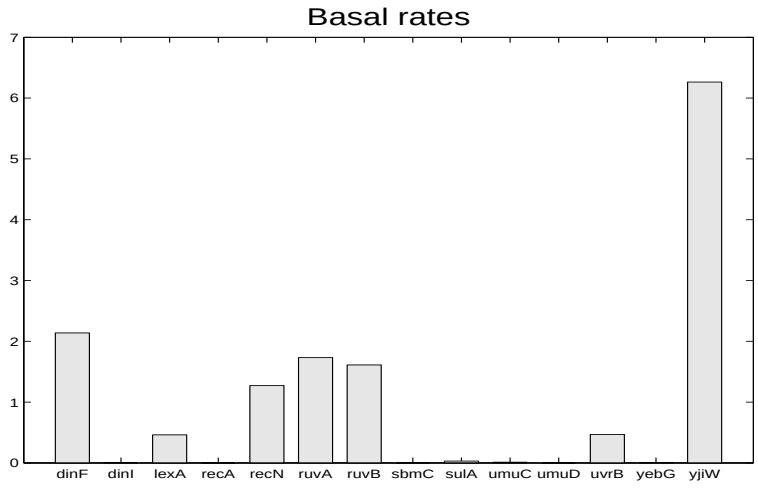


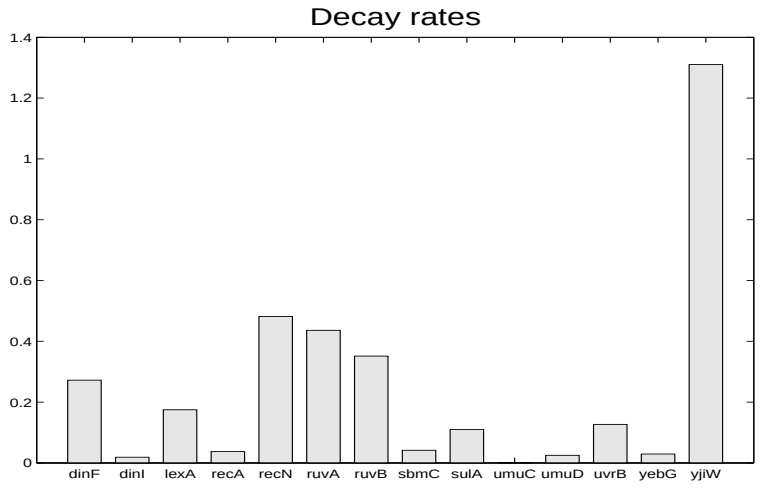


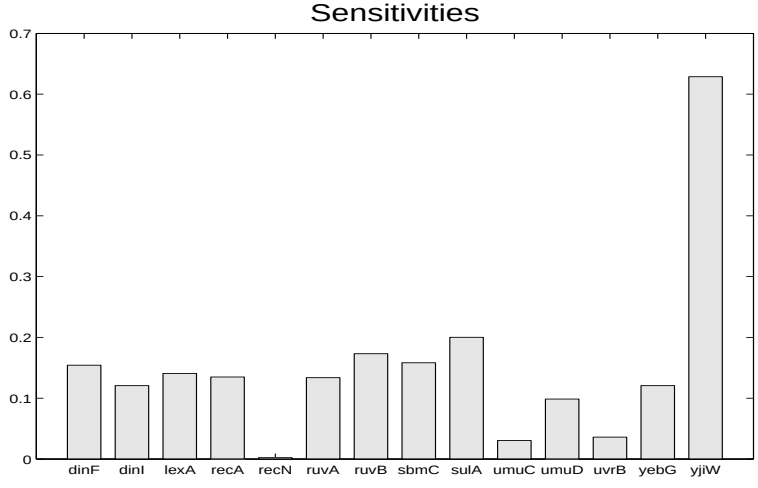
Results in E.coli data: Protein concentration

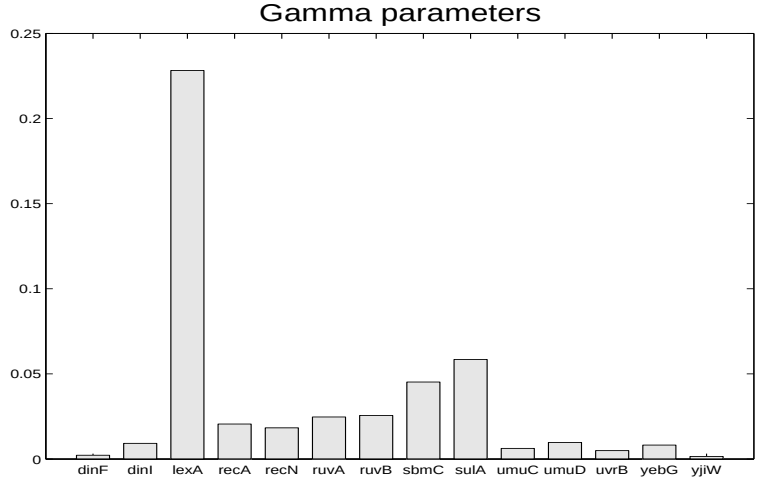


Results in E.coli data: Kinetic parameters

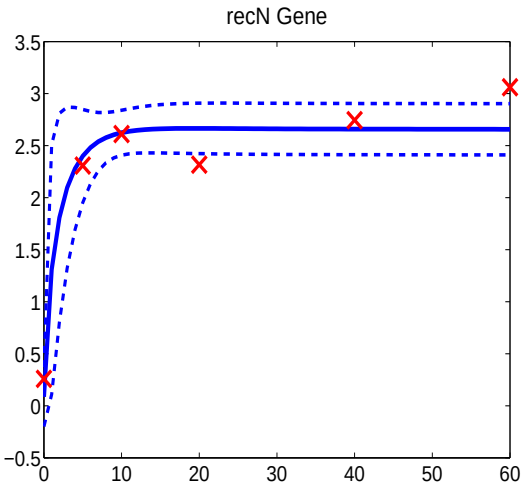


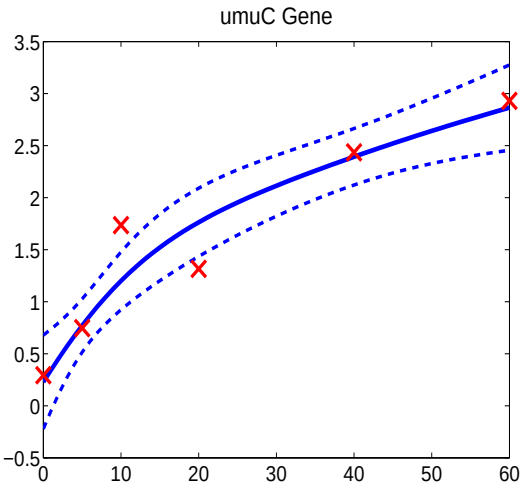


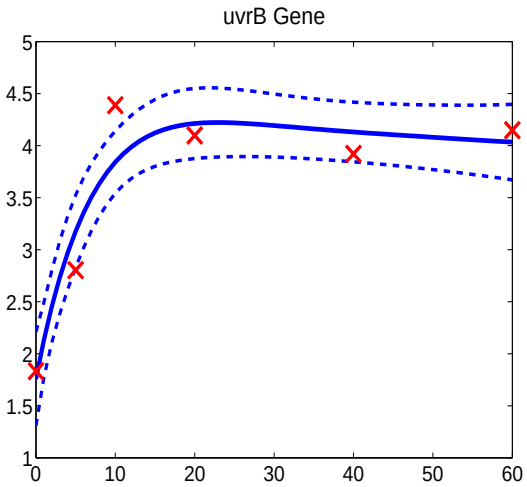


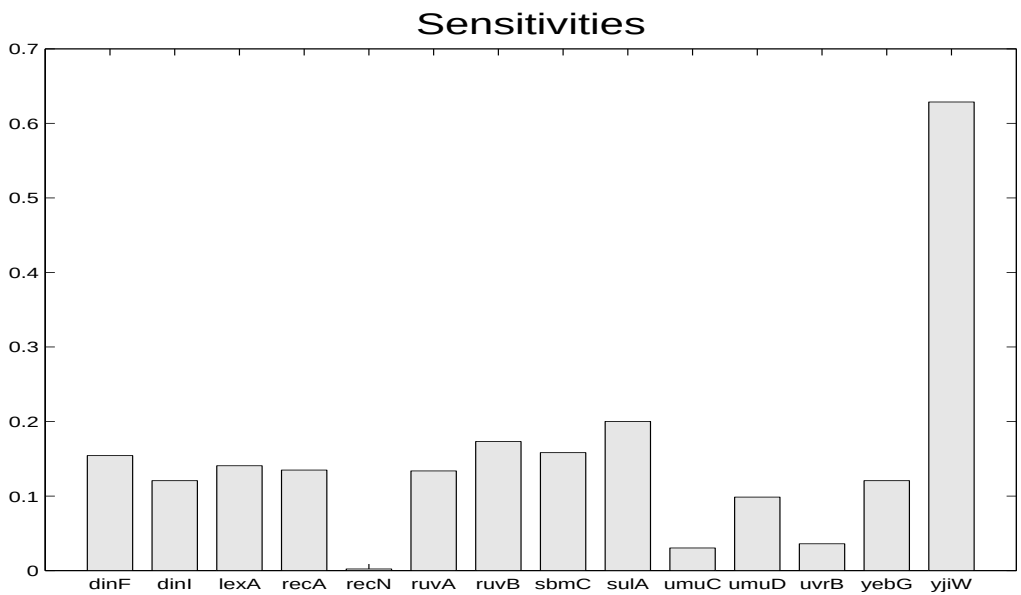


Results in E.coli data: Genes with low sensitivity value

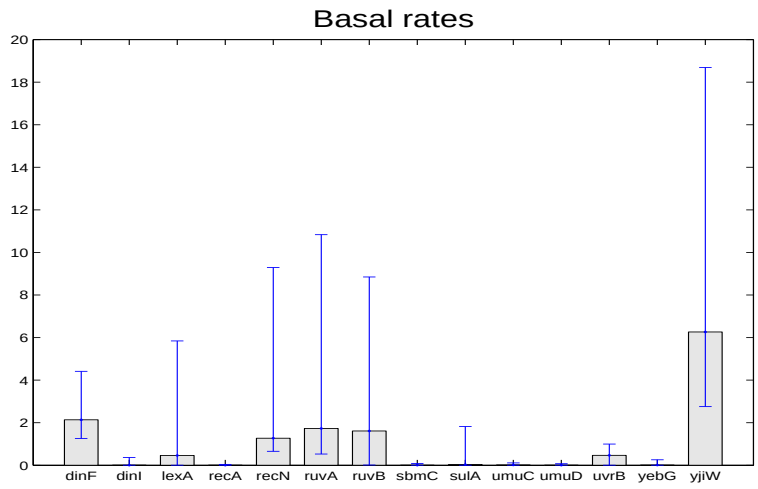


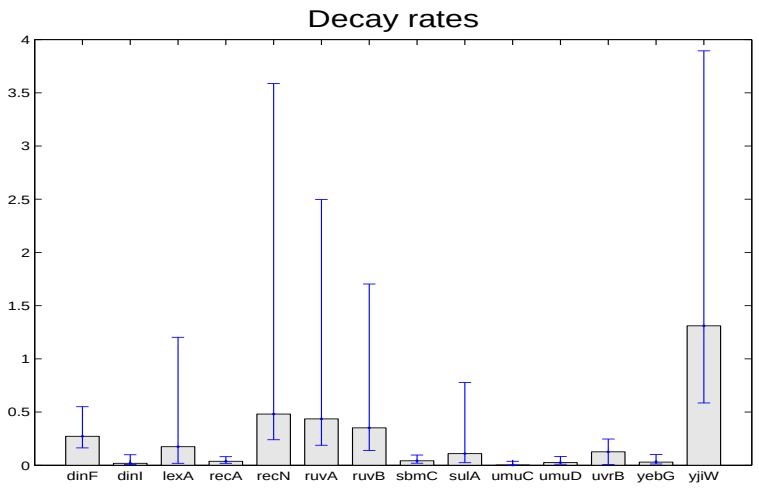


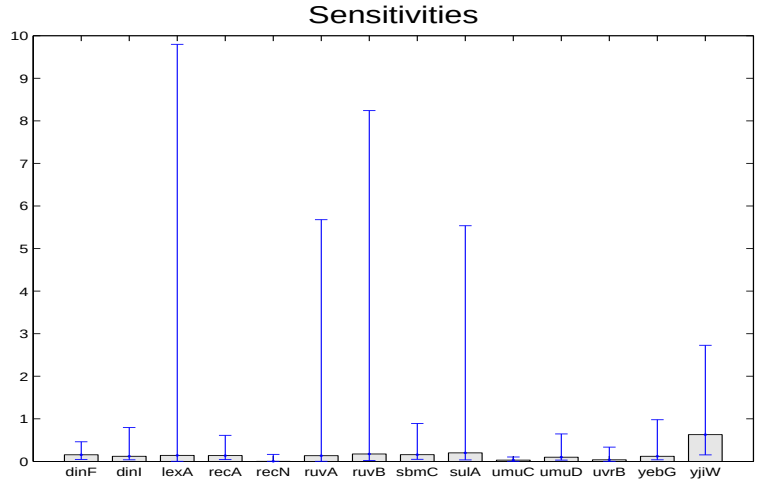


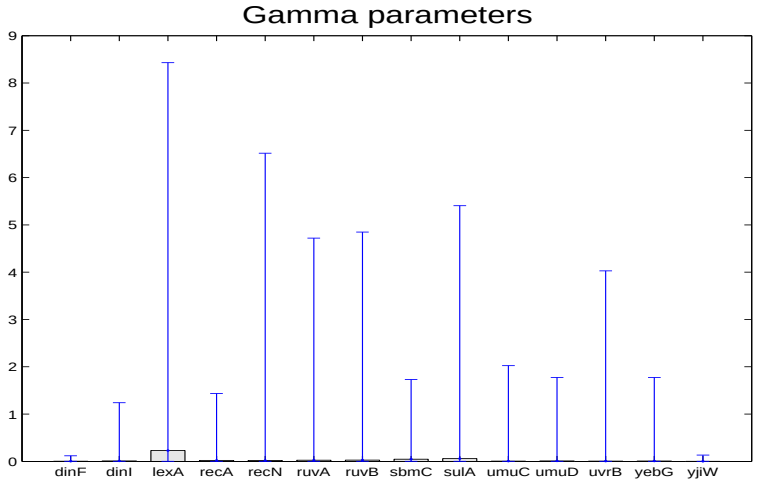


Results in E.coli data: Confidence intervals for the kinetic parameters









Outline

Multiple TFs

- We can generalize the Gaussian process sampling framework to estimate from gene expression data multiple and possibly interacting TFs.
- For linear response, this is tractable, but for nonlinear response (in general) we use sampling.

Learning multiple TFs

- General form of the multiple TF model

$$\frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = B_j + S_j g(f_1(t), \ldots, f_l(t); \mathbf{w}_j) - D_j x_j(t), \quad (2)$$

where the l-dimensional vector $\mathbf{w}_j$ stores the interaction weights between the jth gene and the l TFs. There may be also some bias weight w_{0j} for each gene.

Sigmoid model

- Choose the joint activation function $g(u)$ to be the sigmoid (Mjolsness et al., 1991)

$$h_j = \sum_{i=1}^{I} w_{ji} f_i(t) + w_{j0},$$

$$g(h_j) = \frac{1}{1 + \exp(-h_j)}.$$

- For single TF the above activation function gives rise to Michaelis-Menten when we fix $w_j = 1$.
- For the repressor case we set $w_j = -1$, which however doesn't give rise to the exact Michaelis-Menten repressor equation

Bayesian model

- Likelihood:

$$\prod_{j=1}^{N}\prod_{t=1}^{T} p(x_{jt}|\{\mathbf{f}_i(1 \leq p \leq P_t)\}_{i=1}^{I}, \{A_j, B_j, D_j, S_j\}, \mathbf{w}_j, \sigma_j^2), \tag{3}$$

  where these terms are Gaussians and σ_j^2 is gene-specific variance

- Prior
  - Kinetics $\{A_j, B_j, D_j, S_j\}$ are positive and are represented in the log space: Gaussian priors are used
  - $\{\mathbf{f}\}_{i=1}^{I}$ are the log of the TFs: GP rbf priors with separate timescales
  - $\{\mathbf{w}_j\}$ take real values: Gaussian priors are used
  - Noise variances and GP lengthscales $\{\sigma_j^2, \ell_j^2\}$: Gamma priors

MCMC

Component-wise M-H algorithm. Iteratively sample from conditional posteriors:

1. For $i = 1, \ldots, I$ sample $\mathbf{f}_i$ from the conditional posterior based on the approach of Titsias et. al [2009]
2. For $j = 1, \ldots, N$ sample the kinetic parameters $\{A_j, B_j, D_j, S_j\}$
3. For $j = 1, \ldots, N$ sample the interaction weights $\mathbf{w}_j$
4. For $j = 1, \ldots, N$ sample the gene-specific noise variance σ_j^2.
5. For $i = 1, \ldots, I$ sample the lengthscale ℓ_i^2 of the rbf kernel function.

Side Information

Learning the real TFs that produced the gene expression is not easy because of identifiability problems in parameter space and limited amount of data. Side information obtained from ChiP data can be useful.

- Side information involves the weights W that represent the interactions between genes and TFs. W is $N \times I$ matrix where N the number of genes and I the number of TFs.
- Side information can be expressed as a binary $N \times I$ matrix X. When $x_{ji} = 0$, there is no interaction between the j gene and the i TF, thus $w_{ji} = 0$. When $x_{ji} = 1$, the value w_{ji} can take a positive or negative value which must be inferred by MCMC.
- This scheme can be generalized to probabilistically expressed side information where each x_{ij} is drawn from some probability π_{ji} that expresses our prior belief that the j gene has been regulated by the i TF.

Artificial data

- We consider a toy example with two TFs, that can regulate 20 genes.
- We assume that we have deterministic side information for 8 out of 20 genes. i.e. we know which weights w_{j1} and w_{j2} are zero for these 8 genes, say $j = 1, \ldots, 8$.
- We also assume that the initial conditions in the differential equations are all zero and also that we know the initial (at $t = 0$) activation of the TFs. The number of non-zero elements in the 20×2 matrix W is 25.

Artificial data

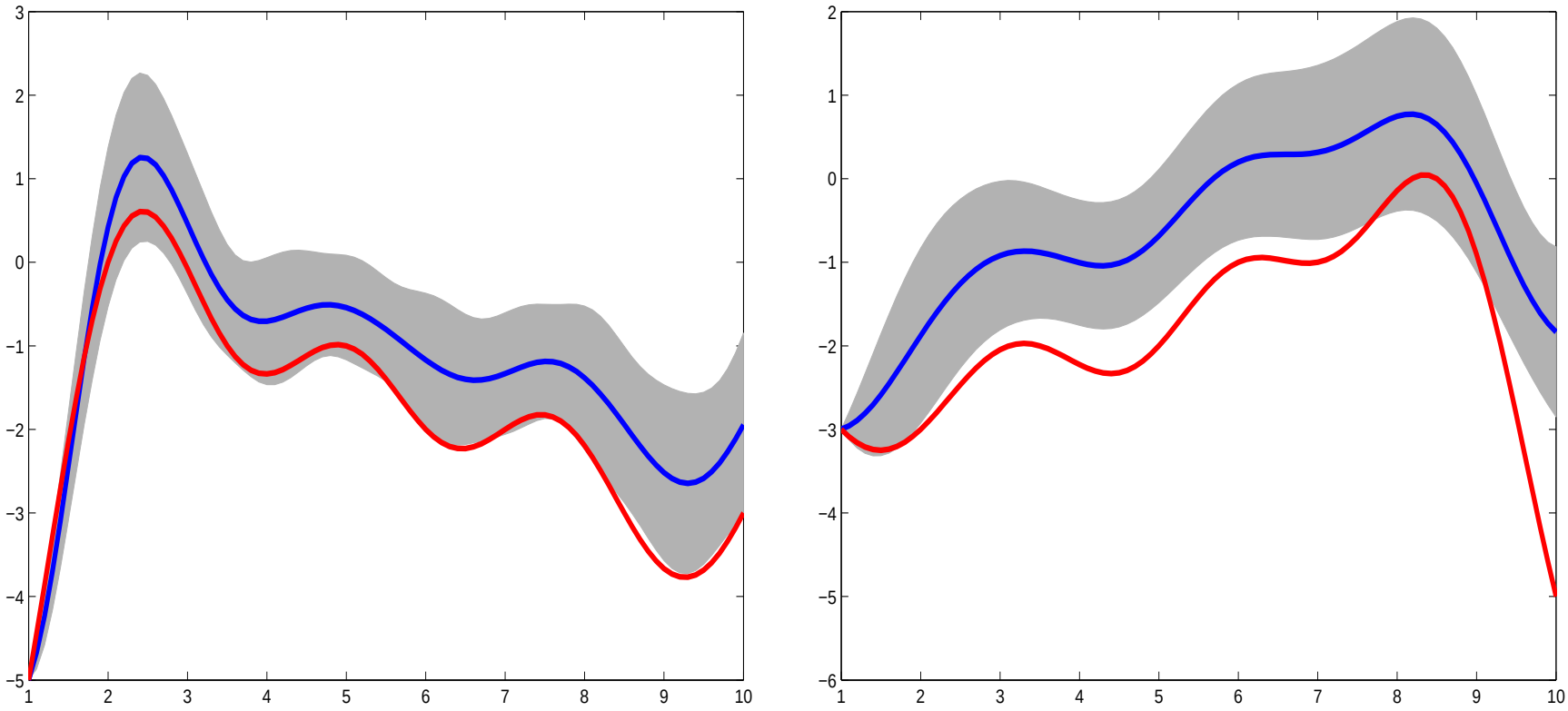


Figure: The inferred profiles of the two TFs (in the log space). With red solid lines are the ground-truth TFs used to generate the toy data. With blue lines shaded error bars are the inferred TF profiles.

Artificial data

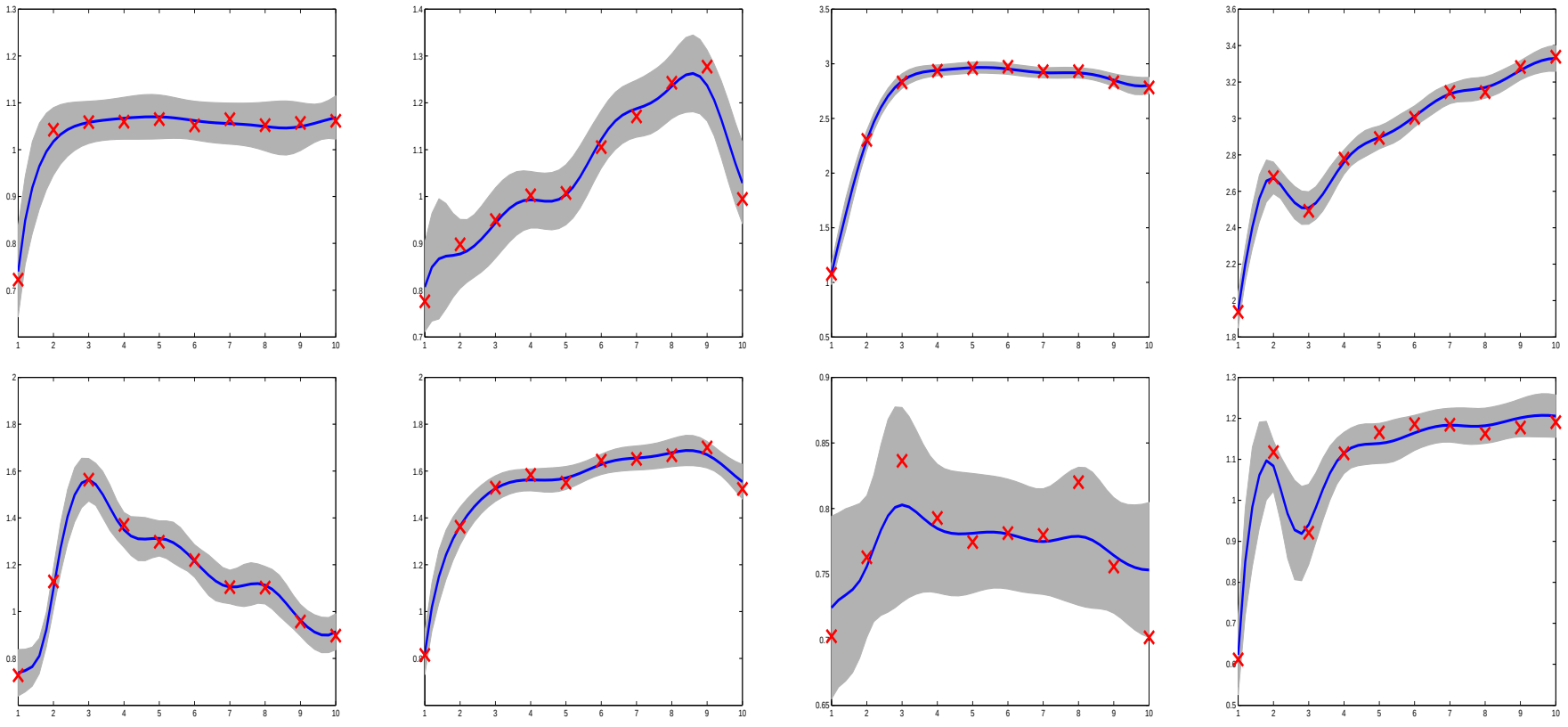


Figure: The predicted gene expressions. Red crosses represent the actual gene expression and the blue line with shaded error bars are the prediction found by MCMC.

Artificial data

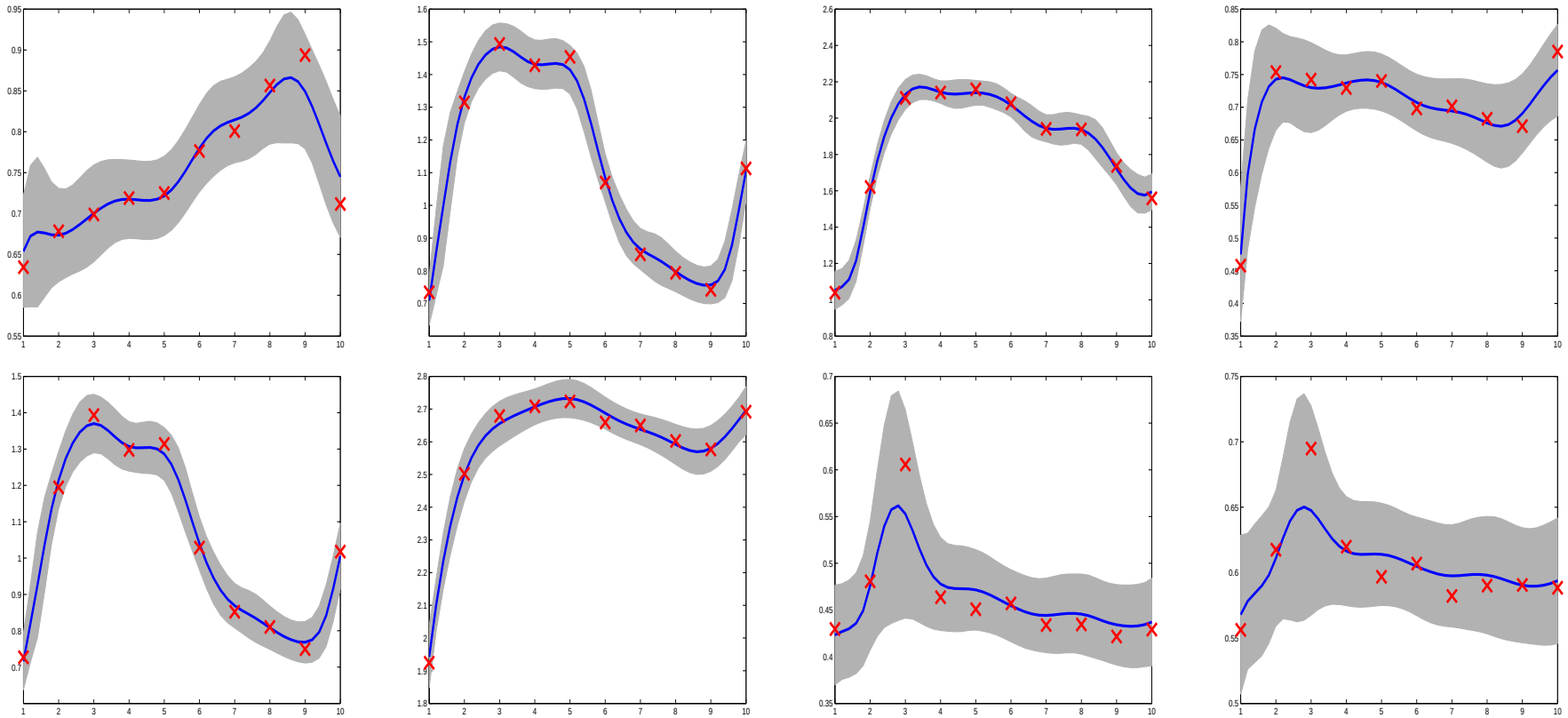


Figure: The predicted gene expressions. Red crosses represent the actual gene expression and the blue line with shaded error bars are the prediction found by MCMC.

Artificial data

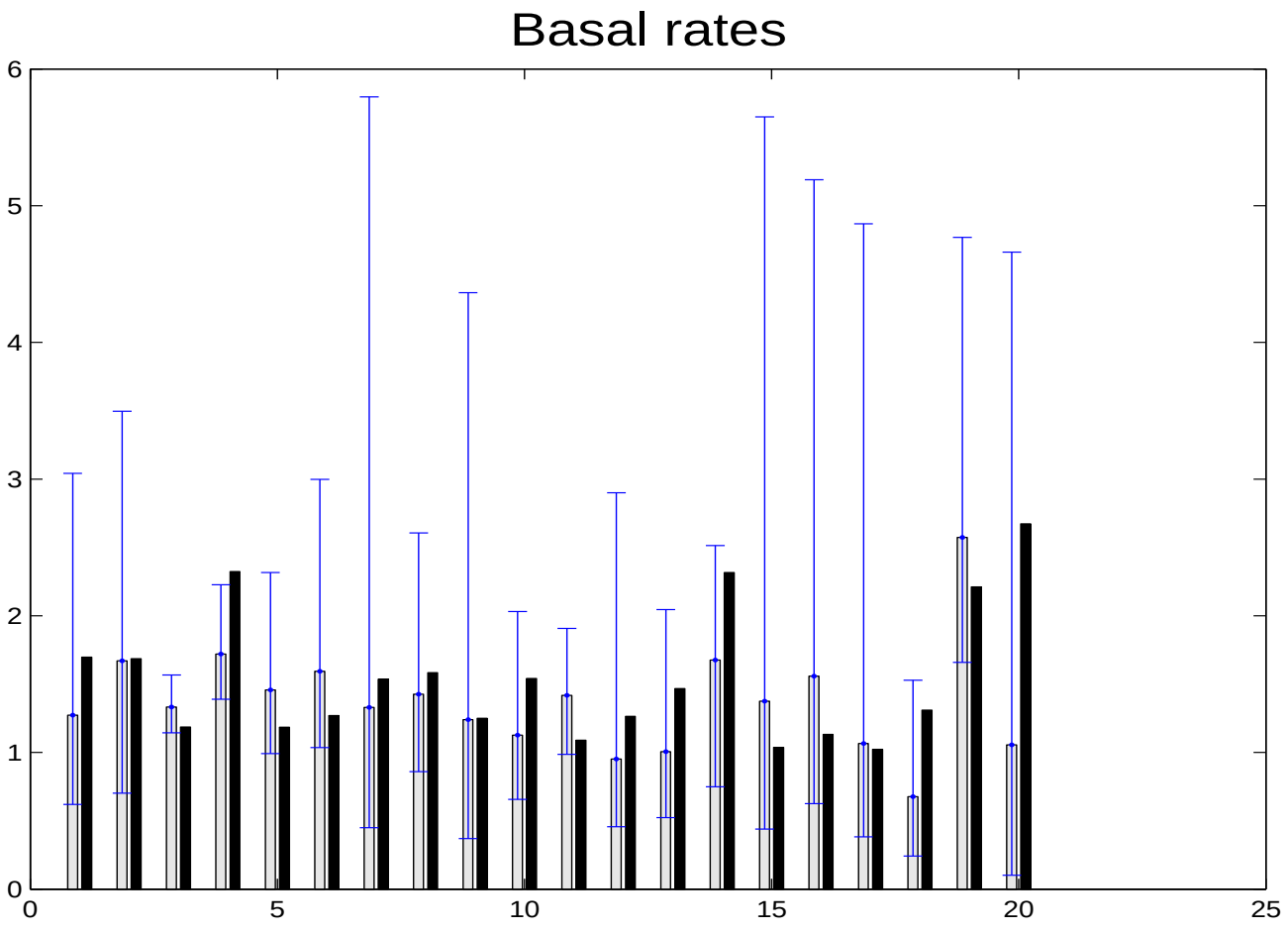


Figure: The inferred basal rates for the 20 genes.

Artificial data

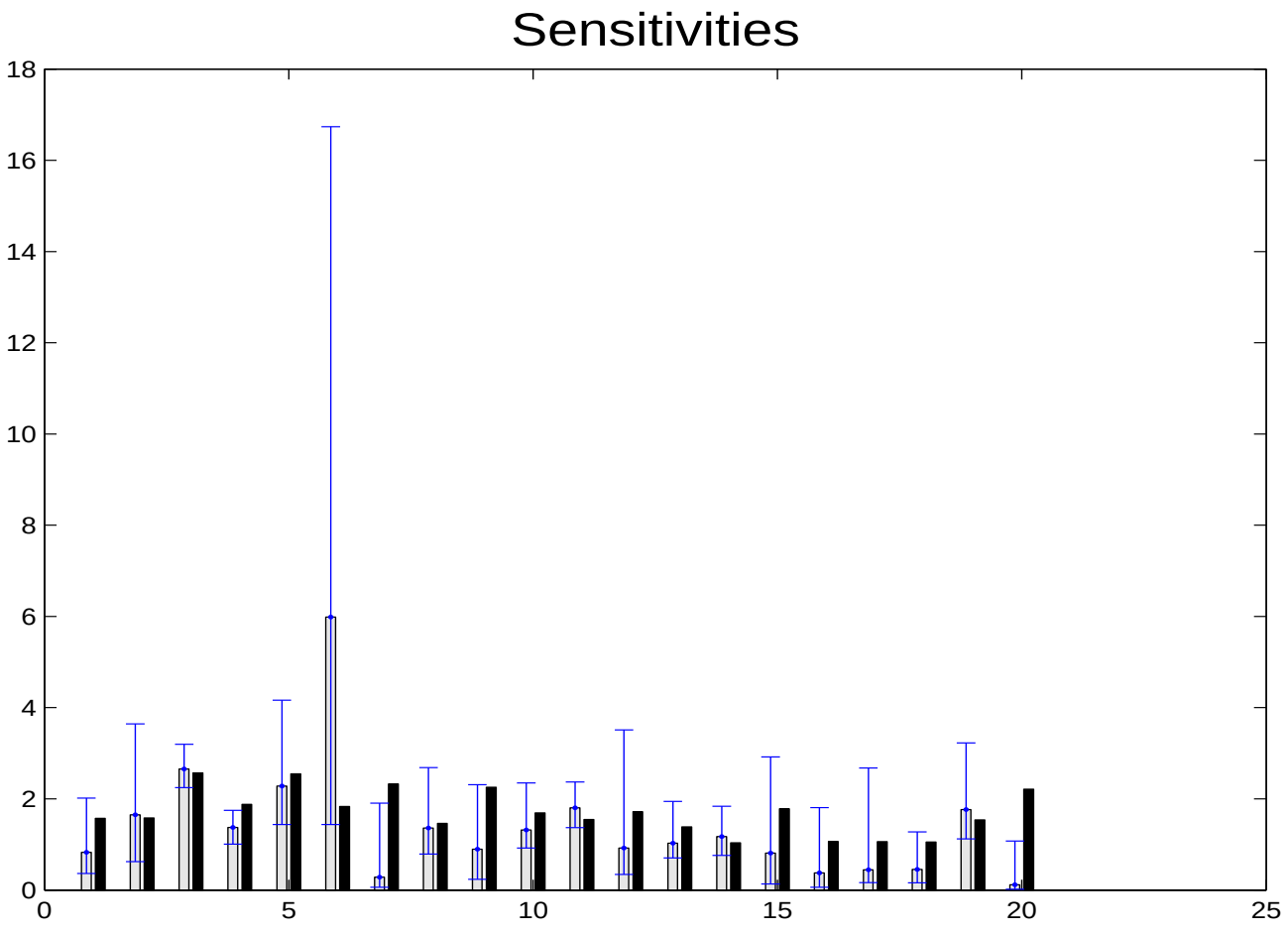


Figure: The inferred sensitivities for the 20 genes.

Artificial data

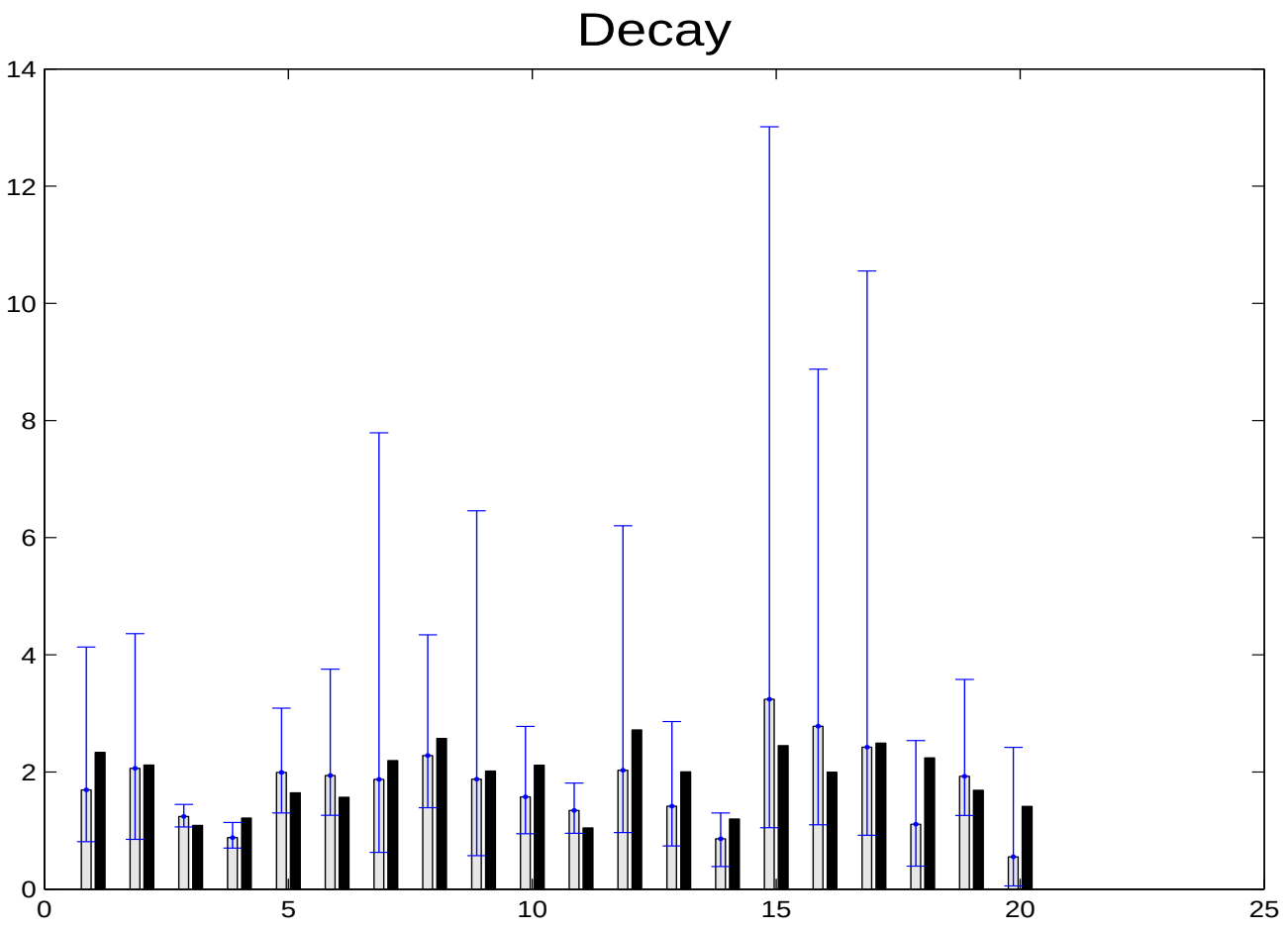


Figure: The inferred decays for the 20 genes.

Artificial data

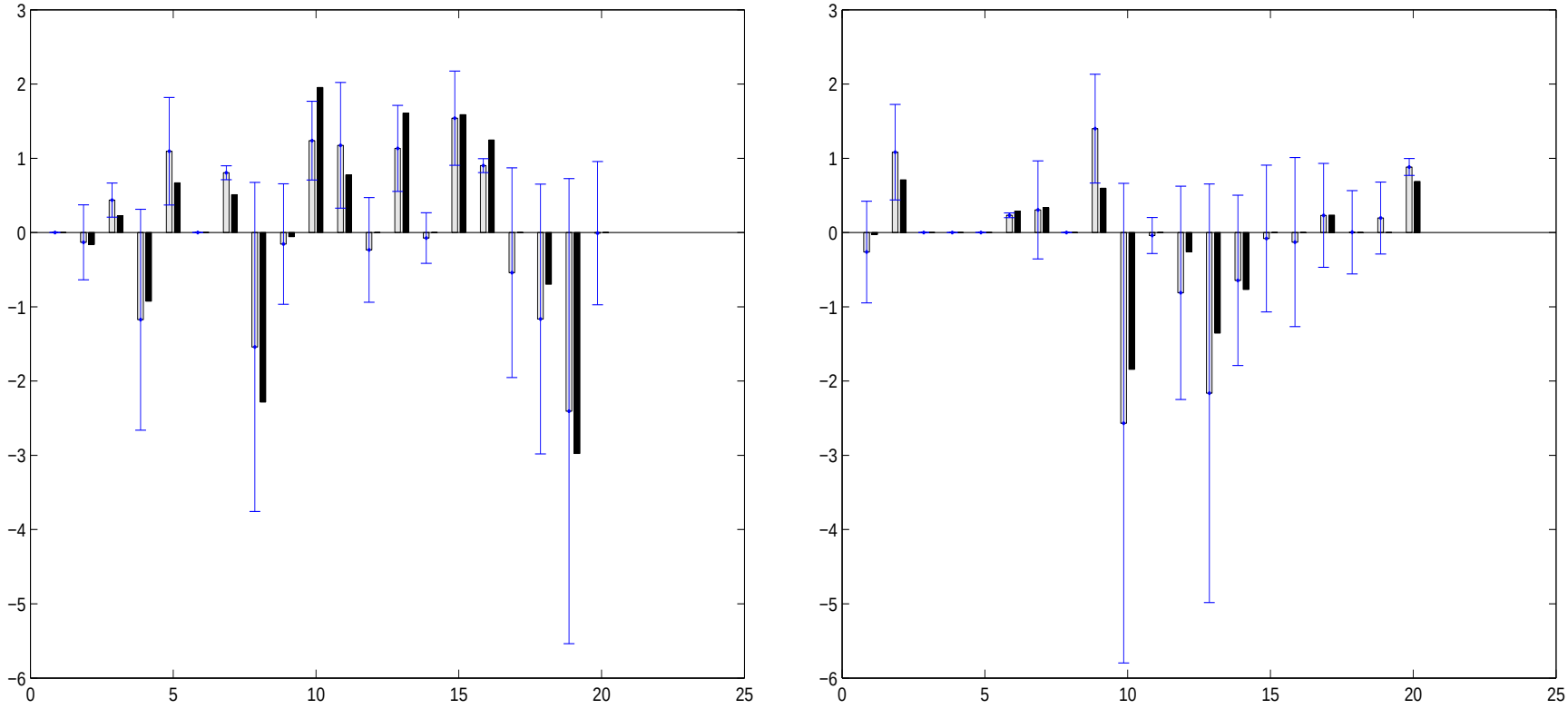


Figure: The inferred interaction weights W. (left) show the interaction weights between the first TF and the 20 genes. (right) show the corresponding weights for the second TF.

Spelman et. al. yeast data

We selected 30 genes regulated by 3 TFs. The 3 TFs are MBP1, FKH2 and STE12. The selection was done based on the ChiP data avalaible so that only the genes that are regulated exclusively by at least one of these 3 TFs were selected.

Yeast data

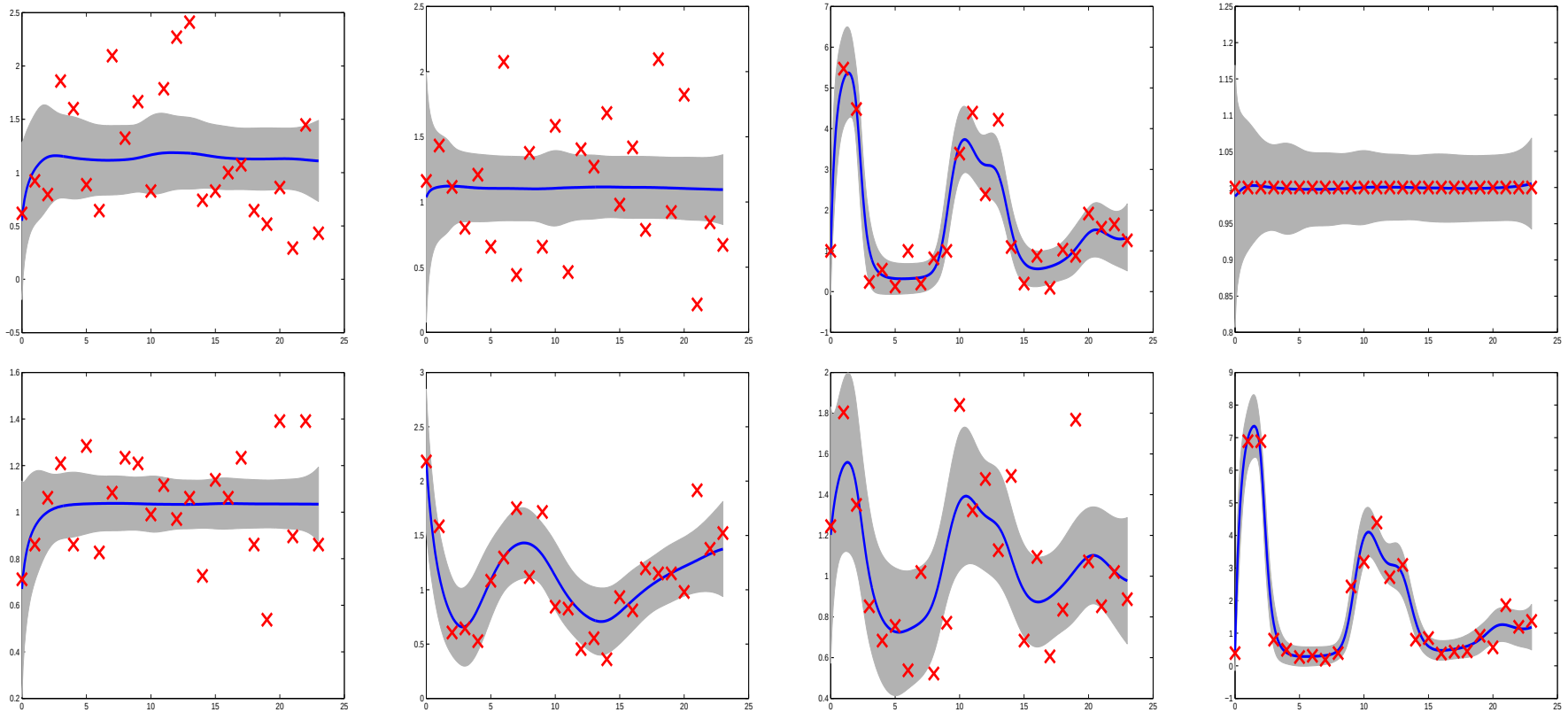


Figure: The predicted gene expressions. Red crosses represent the actual gene expression and the blue line with shaded error bars are the prediction found by MCMC.

Yeast data

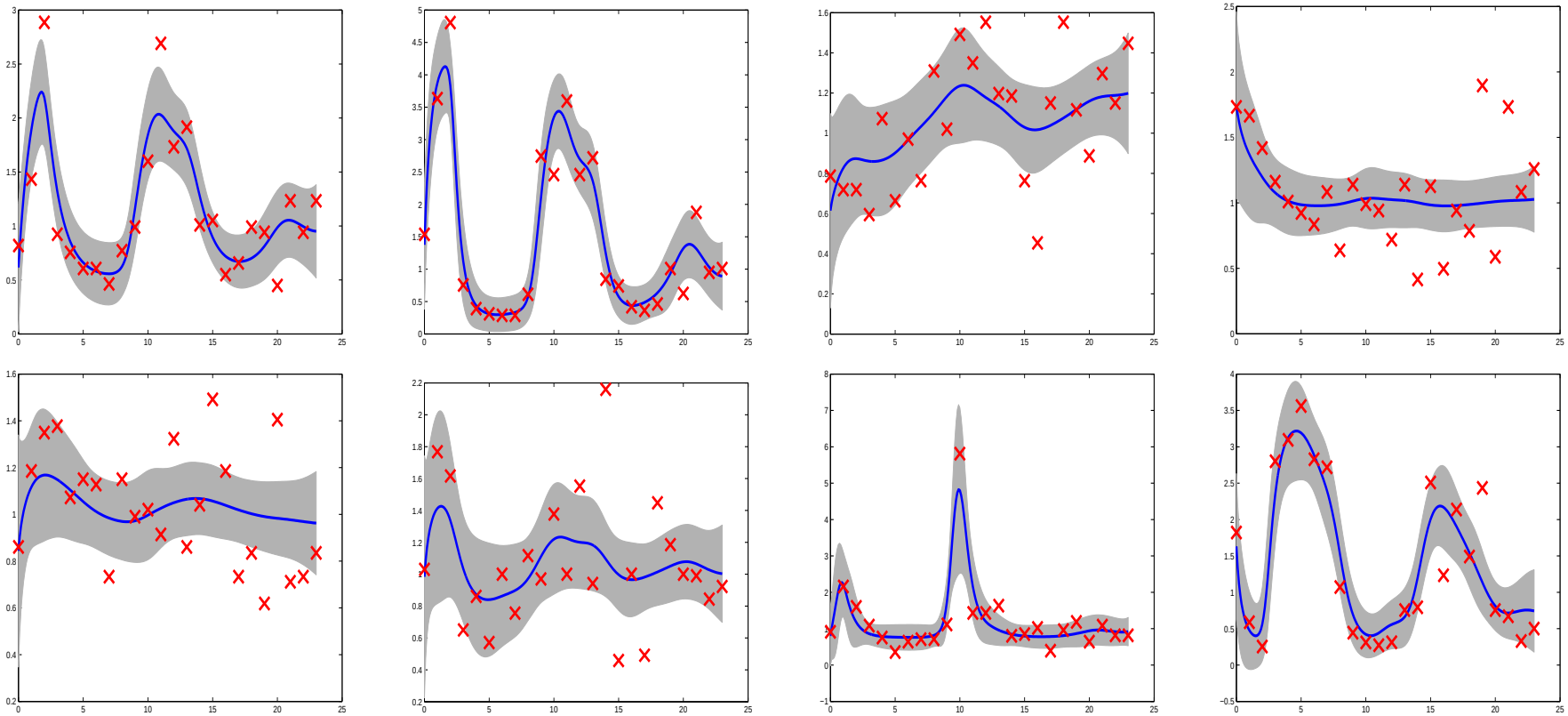


Figure: The predicted gene expressions. Red crosses represent the actual gene expression and the blue line with shaded error bars are the prediction found by MCMC.

Yeast data

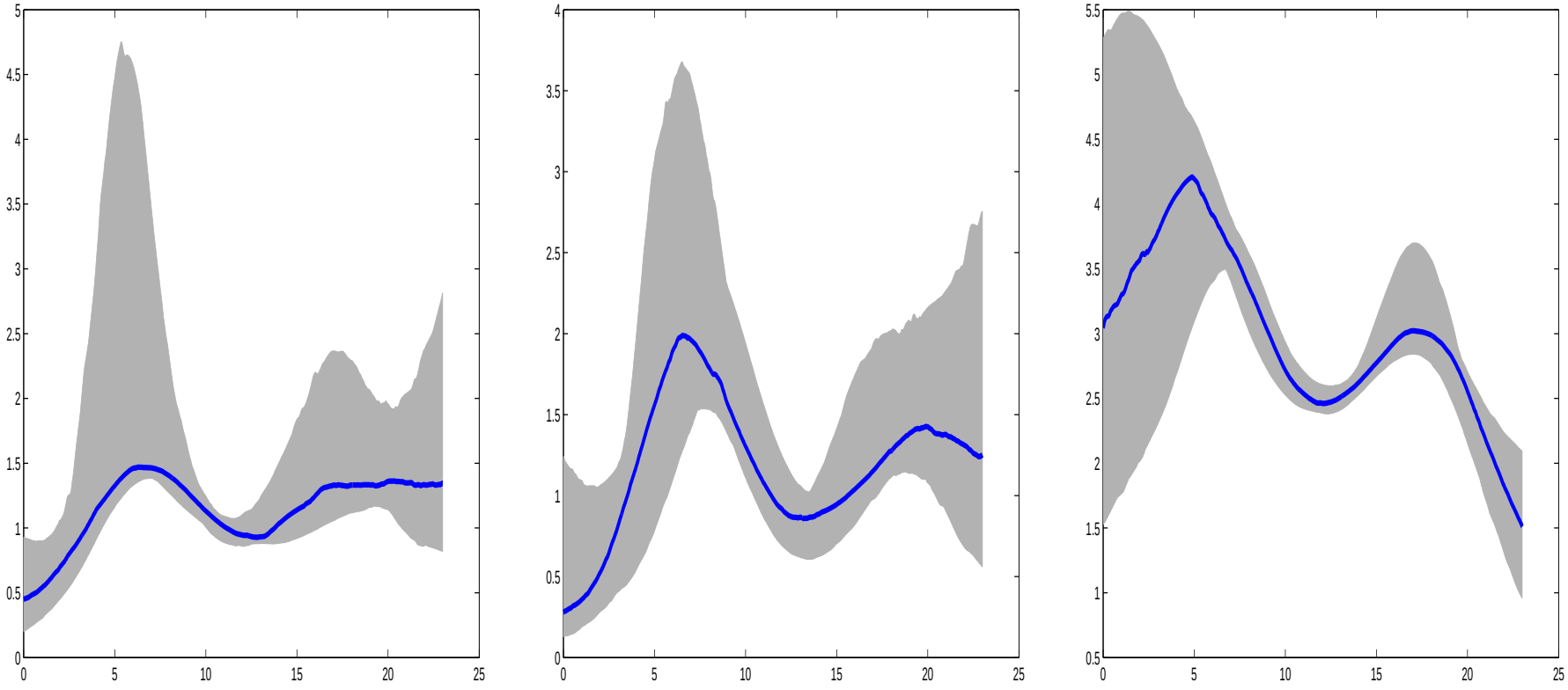


Figure: TF profiles

Sigmoid model

- The sigmoid model is perhaps less biologically plausible. Particularly it assumes that all TFs (activators and repressors) are combined by multiplication

$$\text{sigmoid} = \frac{1}{1 + \prod_{p=1} [\exp(f_p(t))]^{-w_{jp}} \exp(-w_{j0})}$$

  recall that $\exp(f_p(t))$ is the TF.
- This does not look so intuitive.
- Can we define activation functions where the combination is done by addition?
- Saturation and the ability of repressors to turn off the gene expression must be incorporated.
- Next we discuss such a model which can be viewed as a generalization of the Michaelis-Menten model for the single TF case.

Michaelis-Menten multiple TF model

$$\frac{\mathrm{d}x_j(t)}{\mathrm{d}t} = B_j + S_j g(f_1(t), \ldots, f_l(t); \mathbf{w}_j) - D_j x_j(t), \qquad (4)$$

- Let $\mathcal{P} = \{1, \ldots, P\}$ be the set of all TFs
- A_j be the set of TFs that are activators for jth gene and R_j the set of repressors.
- $A_j \cup R_j \subseteq \mathcal{P}$. That is some of the TFs may not regulate the jth gene
- The activation function takes the form

$$g = \frac{\sum_{i \in R_j} w_{ji} + \sum_{i \in A_j} w_{jp} \exp(f_i(t))}{1 + \sum_{i \in R_j} w_{ji} \exp(f_i(t)) + \sum_{i \in A_j} w_{ji} \exp(f_i(t))}$$

where w_{ji} are now non-negative and can be thought as relative sensitivities

Michaelis-Menten multiple TF model

$$g(f_1(t), \ldots, f_I(t); \mathbf{w}_j) = \frac{\sum_{i \in R_j} w_{ji} + \sum_{i \in A_j} w_{ji} \exp(f_i(t))}{1 + \sum_{i \in R_j} w_{ji} \exp(f_i(t)) + \sum_{i \in A_j} w_{ji} \exp(f_i(t))}$$

- Michaelis-Menten equation for a single TF can be obtained as a special case
  - Activation: $A_j = \{1\}$, $R_j = \emptyset$,

$$g(f_1(t); \mathbf{w}_j) = \frac{w_{j1} f_1(t)}{1 + w_{j1} f_1(t)} = \frac{f_1(t)}{\gamma_j + f_1(t)}$$

  - Repression: $A_j = \emptyset$, $R_j = \{1\}$

$$g(f_1(t); \mathbf{w}_j) = \frac{w_{j1}}{1 + w_{j1} f_1(t)} = \frac{1}{\gamma_j + f_1(t)}$$

    where $\gamma_j = \frac{1}{w_{j1}}$

MCMC

- Similar to the sigmoid model. But the set of the activators A_j and the set of repressors R_j are sampled based on Gibbs sampling by taking all possible combinations.

Artificial data

- We consider a set of 30 genes regulated by 3 TFs.
- Side information: We assume we know which TFs regulate each gene, but we do not know whether a TF activates or represses a certain gene
- We wish to estimate the TF profiles, kinetic parameters, etc
  - and to predict which TFs are activators and which are repressors for each gene

Artificial data

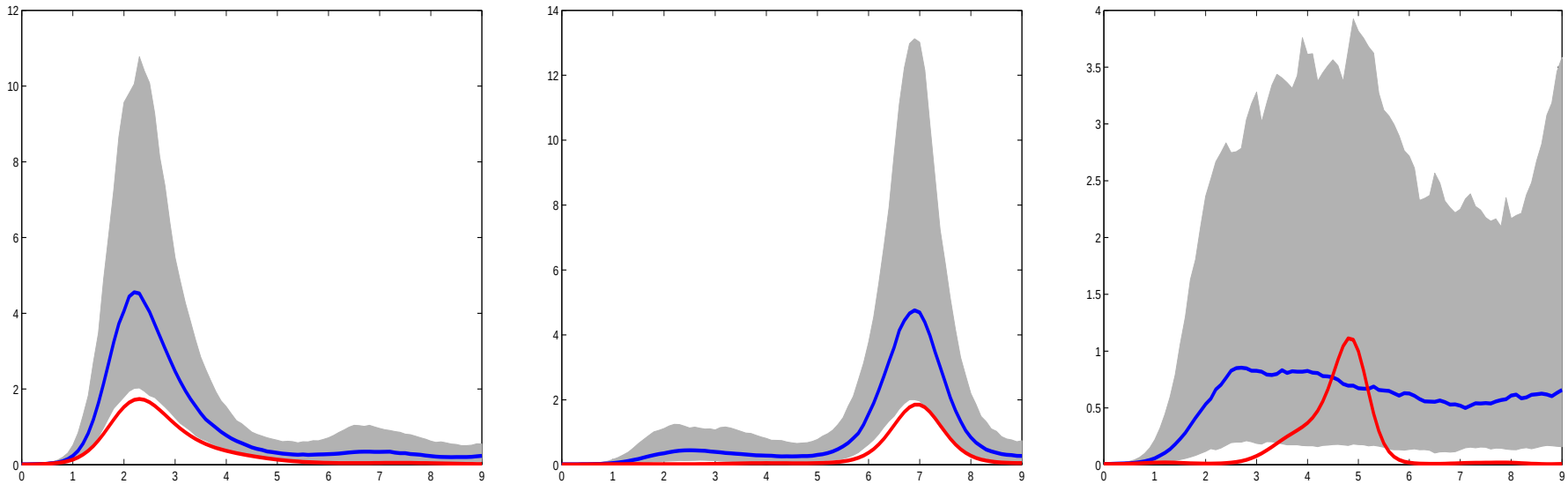


Figure: The inferred profiles of the three TFs. With red solid lines are the ground-truth TFs used to generate the toy data. With blue lines shaded error bars are the inferred TF profiles.

Artificial data

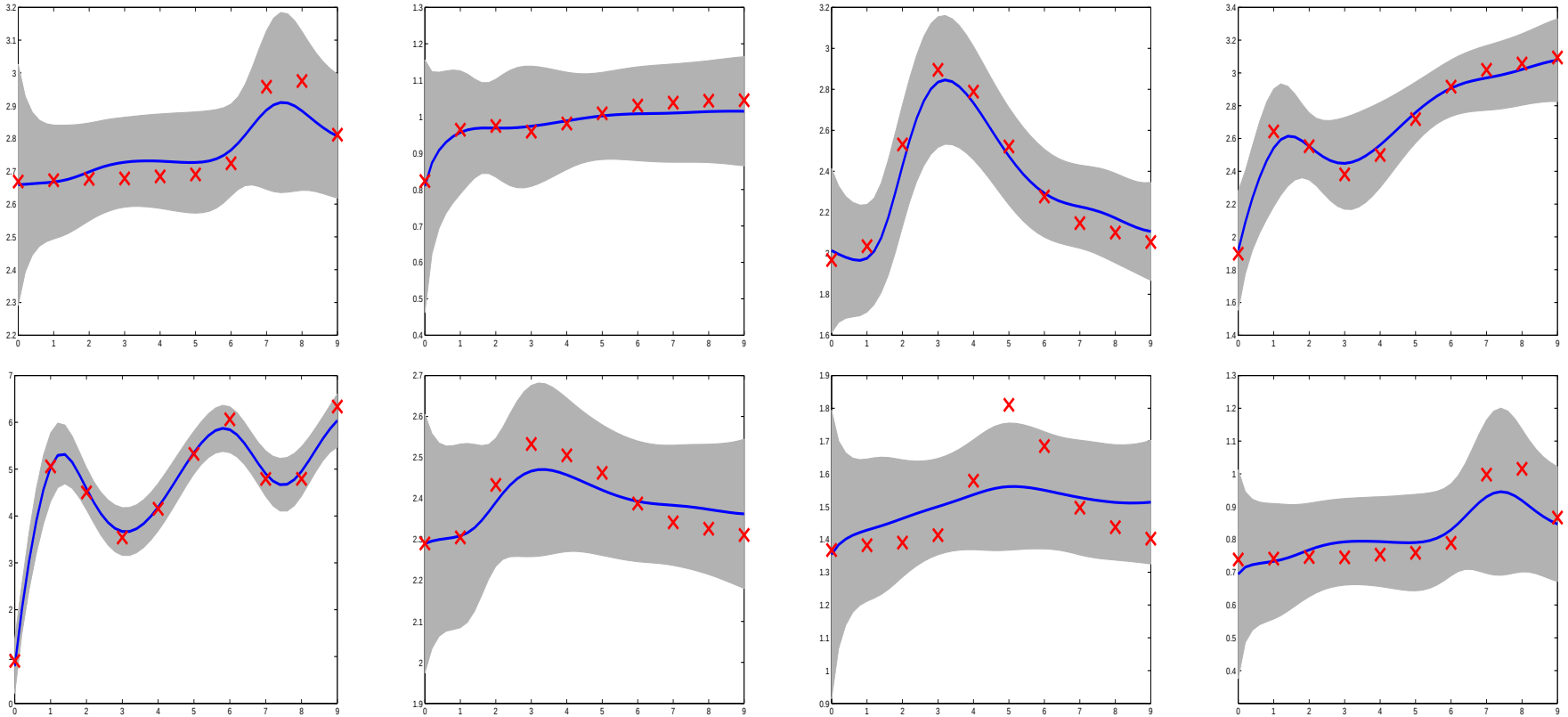


Figure: The predicted gene expressions. Red crosses represent the actual gene expression and the blue line with shaded error bars are the prediction found by MCMC.

Artificial data

Total classfication error regarding which TFs are activators and which are repressors for each gene

$$0.2447 \pm 0.0617$$